The University of Edinburgh's Bengali-Hindi Submissions to the WMT21 News Translation Task

Proyag Pal **Alham Fikri Aji** **Pinzhen Chen** **Sukanta Sen**
School of Informatics, University of Edinburgh, Scotland
{proyag.pal, a.fikri, pinzhen.chen, ssen}@ed.ac.uk

Abstract

We describe the University of Edinburgh's Bengali↔Hindi constrained systems submitted to the WMT21 News Translation task. We submitted ensembles of Transformer models built with large-scale back-translation and fine-tuned on subsets of training data retrieved based on similarity to the target domain. For both translation directions, our submissions are among the best-performing constrained systems according to human evaluation.

1 Introduction

We present the University of Edinburgh's participation in the WMT21 news translation shared task on the Bengali→Hindi (Bn→Hi) and Hindi→Bengali (Hi→Bn) language pairs. We followed the constrained condition, i.e. only using the data provided by the organizers. The training data for these language pairs consisted of noisy crawled data, and was mostly out-of-domain with respect to the validation and test domain. Therefore, most of our efforts concentrated on fine-tuning models to adapt to the target domain. We also explore multiple back-translation methods, and ensembles of models trained and fine-tuned with different methods.

Building our systems consisted of the following steps, each of which is described in more detail in the remaining sections of this paper:

- Cleaning the noisy parallel data (Section 3).
- Training ensembles of Transformer models on the cleaned provided data for back-translation; and using the back-translated data along with the clean parallel data to train new models (Section 4).
- Fine-tuning the models on subsets of training data retrieved that are similar to the target domain, based on different similarity measures (Section 5).
- Ensembling various models and decoding with optimal parameters (Section 6).

We also report some methods that we tried to use but did not work in Section 8.

2 Model Configuration

Our models follow the Transformer-Big architecture (Vaswani et al., 2017): 6 layers of encoders and decoders, 16 heads, an embedding size of 1024, a unit size of 4096, etc. We found that smaller Transformer architectures performed worse.

All models are trained with the same vocabulary of 32k SentencePiece subwords (Kudo and Richardson, 2018) to allow ensembling. We use a shared vocabulary between source and target, as well as tied embeddings (Press and Wolf, 2017). We tried other vocabulary sizes too: 5k, 10k, and 20k, though all of them had similar performance. We also included several special tokens in the vocabulary, of which we finally used only one for tagged back-translation (Caswell et al., 2019).

We train models with 32GB dynamic batch size and an optimizer delay (Bogoychev et al., 2018) of 3 with the Adam optimizer (Kingma and Ba, 2015) under a learning rate of 0.0003, until we see no improvement within 10 consecutive validation steps. All models were trained with the Marian NMT toolkit (Junczys-Dowmunt et al., 2018)[1]

3 Datasets and Cleaning

3.1 Corpora

All our models are trained in the constrained scenario – even more specifically, we only use data provided for the news translation task for these specific language pairs. This consists of 3.3M parallel sentences from the CCAligned corpus (El-Kishky et al., 2020), along with monolingual data in both languages. The details of the corpora used along with their sizes are shown in Table 1.

[1] https://github.com/marian-nmt/marian

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 185–191
November 10–11, 2021. ©2021 Association for Computational Linguistics

Corpus	Lines (M)
Parallel	3.36
+ deduplication and filtering	2.03
Monolingual	
Bn NewsCrawl	10.1
Bn CommonCrawl	49.6
Hi NewsCrawl	46.1
Hi CommonCrawl	202

Table 1: Bn and Hi corpora used in our submissions.

3.2 Cleaning

Since the CCAligned corpus is built from web crawls and is known to be very noisy (Caswell et al., 2021), we focused on cleaning the parallel data before training translation models. Our main approaches are rule-based and heuristic cleaning methods, along with language identification and language model filters. Our final systems used the following cleaning methods for the parallel corpus:

De-duplication Duplicate sentence pairs – around 17.3% of the corpus – were removed.

Splitting multi-language sentences We observed large chunks of the corpus where the sentences on the Bengali side also had their English translations attached in the same line. Some rough punctuation and script-based heuristics were used to remove the English segments from these lines. The roughness of these heuristics also affected a large number of other lines, mostly noisy ones containing non-lexical information, but we observed no degradation of quality due to this inaccuracy. We also found some such sentences on the Hindi side, but they were less frequent and removal showed no improvement in quality, so we did not split Hindi sentences in this way for our final models.

Language ID filtering We used publicly available FastText language identification models (Joulin et al., 2016, 2017)[2] to filter out lines in wrong languages. We get the top 3 predictions for each line, throw out lines where the right language does not appear in the top 3 for one or both sides, sort by the language prediction probabilities, and based on manual inspection, arrive at minimum threshold probabilities of 0.6 for Bengali lines and 0.4 for Hindi lines, above which lines are retained.

Language model filtering We used KenLM (Heafield, 2011) to train separate trigram language models for Bengali and Hindi, on all provided Extended CommonCrawl monolingual data, and used these to score the parallel data. We retain sentences with $\log_{10}$ probabilities greater than -4.

4 Training with Synthetic Data

In each language direction, we trained 4 models with different seeds. We then ensembled these 4 models to back-translate (Sennrich et al., 2016) all the provided monolingual data. We used this translated data in many different ways as described in the remainder of this section.

Tagged back-translation Following Caswell et al. (2019), we prefixed a special <__BT__> token to all back-translated news monolingual data, combined the data with the clean parallel data, and trained new models.

Two-step training We first trained models on all the back-translated data only, then once that converged, continued training on the clean parallel data. Since the amount of monolingual data far exceeds the amount of parallel data, this training regime gave us better results than mixing parallel and back-translated data at the same time. The latter method would also involve finding the right amount of back-translated data to sample/select, since using it all would overwhelm the parallel training data.

Forward translation We also trained models on parallel data along with all the back-translations and all forward translations, i.e. instead of strictly keeping target monolingual data on the target side and synthetic back-translated data on the source side, we used both directions of translated data.

5 Fine-tuning to the Target Domain

5.1 Fine-tuning on retrieved sentences

Unlike many of the other language pairs in the news translation task, the Bengali-Hindi pair does not include any known in-domain training corpora. The training data is aligned from documents obtained through untargeted web crawling (El-Kishky et al., 2020), and thus contains out-of-domain and noisy text. On the other hand, the target domain, reflected in the validation and test sets, consists of Wikipedia content[3].

[2] https://fasttext.cc/docs/en/language-identification.html

[3] Despite it being part of the 'news translation' task

To adapt our models to the target domain, we retrieved sentences from the training corpora which are similar to the **source** side of validation and test sets based on different similarity measures, and then fine-tuned the models on these subsets of data. The remainder of this section describes the different methods to retrieve the relevant subsets of data. The number of sentence pairs retrieved by each of these methods which are then used for fine-tuning is shown in Table 2.

Retrieval	**Source**	**Lines (K)**	
		Bn	**Hi**
1 bigram overlap	dev	448	891
2 bigram overlap	dev	243	597
3 bigram overlap	dev	158	445
1 bigram overlap	dev, test	487	932
2 bigram overlap	dev, test	273	639
3 bigram overlap	dev, test	183	479
LM threshold -2.5	dev	50	175
LM threshold -2.0	dev, test	12	13
TF-IDF	dev, test	5.6	27.9
TF-IDF cluster	dev, test	20	20

Table 2: Number of training sentence pairs retrieved for fine-tuning by different methods.

Based on vocabulary overlap The simplest method is to retrieve any sentence pairs whose source texts have 1, 2, or 3 non-punctuation bigrams which occur on the source side of the validation and test sets. Due to the large mismatch between training corpus and target domain, this method retrieves a surprisingly small proportion of the training corpus, as shown in Table 2.

Based on language model scoring We trained n-gram language models on the validation and test set or validation set data only, scored the parallel data with these language models, then kept sentences scoring above a certain threshold. Even though the small size of the validation data means that the language model is probably not very good, we still see some improvements by fine-tuning on data retrieved this way.

Based on TF-IDF similarity We first adapted the document aligner[4] from ParaCrawl (Bañón et al., 2020) to work at sentence level. This tool uses the translation of a source text (Uszkoreit et al., 2010) to match potential target text using cosine similarity of TF-IDF-weighted word frequency vectors. In this case, we match the source side of our validation and test sets with the parallel text to find potential "matches". This method retrieves too few matches with only the validation set, but we got a few thousand sentence pairs (Table 2) from a combination of validation and test sets.

Following Chen et al. (2020b), we also developed a variant where we first cluster each source sentence with another X sentences in the validation and test sets based on n-gram TF-IDF vector cosine similarity, then treat the cluster as a single query and compare it against each source sentence in the parallel training data. We always picked the top 20K resulting pairs. Through manual inspection, we found that the resulting corpus is very reasonable when we cluster the whole validation and test sets as one query, making the fine-tuning essentially a test domain adaptation process.

5.2 Fine-tuning on the validation set

Since the validation data is the only domain-specific data we had, similar to Chen et al. (2020a), we fine-tuned all our final models on a portion of the validation set (we used 95% of the data instead of 75%) until it stopped improving on the rest of the validation set. This was done as a final additional step after the other kinds of fine-tuning described previously.

6 Ensembles and Decoding Parameters

6.1 Ensembles

As shown in Table 3, our primary submissions consist of ensembles of multiple models trained and fine-tuned in different ways. Due to the component models not being very high-quality, we observed that this type of ensemble produces more robust translations than simple ensembles of models trained identically with different seeds.

6.2 Optimal decoding hyperparameters

Using an initial ensemble of 4 models, we swept a wide range of values of beam size and length normalization hyperparameters to decode the validation set. We find that optimizing these can result in an improvement of up to 0.5 BLEU on the validation set. We obtained the best scores with a beam size of 16, and a length normalization parameter of 1.3 for Bn→Hi and 0.7 for Hi→Bn, and used these values to decode the test set.

[4] https://github.com/bitextor/bitextor/tree/master/document-aligner

	Model	Bn→Hi		Hi→Bn	
		BLEU	ChrF	BLEU	ChrF
(1)	Single model baseline – Parallel data	19.56	0.4638	10.70	0.4378
(2)	Ensemble – Parallel data	20.37	0.4733	11.47	0.4482
(3)	Parallel + back-translated data	18.62	0.4577	9.78	0.4360
(4)	Parallel + backward + forward translations	20.16	0.4697	11.78	0.4503
(5)	Continue training on (3) with parallel data	21.26	0.4784	12.29	0.4587
(6)	Continue training on (4) with parallel data	20.97	0.4767	12.02	0.4470
(7)	Tagged BT (NewsCrawl only) + parallel data	20.61	0.4753	12.13	0.4541
	(5) fine-tuned on:				
(8)	1 bigram overlap, dev	21.55	0.4816	12.26	0.4573
(9)	2 bigram overlap, dev	21.49	0.4806	12.31	0.4587
(10)	3 bigram overlap, dev	21.35	0.4803	12.44	0.4600
(11)	LM threshold -2.5, dev	21.30	0.4794	12.29	0.4590
(12)	1 bigram overlap, dev+test	21.45	0.4814	12.29	0.4599
(13)	2 bigram overlap, dev+test	21.52	0.4812	12.21	0.4568
(14)	3 bigram overlap, dev+test	21.38	0.4794	12.26	0.4594
(15)	LM threshold -2.0, dev+test	21.29	0.4792	12.24	0.4563
(16)	TF-IDF, dev+test	21.32	0.4788	12.32	0.4601
(17)	(6) fine-tuned on TF-IDF cluster, dev+test	20.26	0.4710	12.02	0.4470

Table 3: Validation set BLEU and ChrF scores for our models.

Submitted ensembles	Bn→Hi		Hi→Bn	
	BLEU	ChrF	BLEU	ChrF
(8)+(9)+(10)+(11)	21.75	0.4895	–	
(6)+(7)+(8)+(9)+(10)+(11)+(16)+(17)	–		12.55	0.4536

Table 4: Test set BLEU and ChrF scores for our primary submissions. Model numbers refer to models from Table 3, but note that all models were fine-tuned on the validation set before ensembling.

6.3 Sentence splitting

In the source texts of the test set, we observed many instances of more than one sentence in one line. Since our models are trained on single sentences, we chose to run a sentence splitter on the test source, translate, and rejoin the translated sentences. For this purpose, we used the Moses sentence splitter (Koehn et al., 2007)[5] for Bengali text, and the IndicNLP sentence splitter (Kunchukuttan, 2020) for Hindi.

6.4 Numeral transliteration

Due to the fact that numerals in the Latin script are often used in Bengali and Hindi text, which is reflected by the web crawled training data, our models tend to generate a mix of Latin and Bengali/Hindi numerals, sometimes even in the same sentence. To ensure consistency, we transliterated all Bengali or Hindi numerals in our test outputs to their Latin script counterparts (it is equally feasible to convert Latin numerals to the target language). While this may not help in terms of automatic metrics (we lose 0.3-0.5 BLEU after this step), we believe human evaluators would prefer consistency in this regard.

7 Results

Table 3 shows BLEU[6] and ChrF[7] scored using sacreBLEU (Post, 2018) on the validation sets. We see that fine-tuning on the retrieved subsets of data consistently results in quality gains. We tried many different ensembles and, upon visual inspection, found that models fine-tuned on data retrieved on the basis of similarity to validation and test sets were not necessarily better than those from validation sets only.

[5] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/ems/support/split-sentences.perl

[6] signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1

[7] signature: chrF2+numchars.6+space.false+version.1.5.1

Ave.	Ave. z	System
82.1	0.202	GTCOM
79.1	0.163	Online-B
77.5	0.080	TRANSSION
78.0	0.076	MS-EgDC
78.0	**0.054**	**UEdin**
76.1	-0.015	Online-Y
75.7	-0.080	HuaweiTSC
75.7	-0.107	Online-A
70.8	-0.373	Online-G

(a) bn→hi

Ave.	Ave. z	System
95.0	0.245	HuaweiTSC
94.8	0.236	Online-A
94.5	0.233	GTCOM
94.6	**0.214**	**UEdin**
92.3	0.080	Online-Y
92.0	0.045	TRANSSION
91.3	0.029	Online-B
90.9	-0.008	MS-EgDC
73.5	-1.100	Online-G

(b) hi→bn

□ constrained ▨ unconstrained

Table 5: Human evaluation results. Our submissions are in bold. Systems within a cluster are considered tied.

Table 4 reports the automatic scores of our final submitted systems on the test sets. As shown in Table 5, according to human evaluation conducted by the task organizers, our systems rank at the top (tied) among all the constrained submissions for both translation directions.

8 Unsuccessful Attempts

In this section, we document some methods that we tried to use, but which did not work at all or did not result in better systems.

Dual conditional cross-entropy filtering Our initial cleaning effort was to use dual conditional cross-entropy (Junczys-Dowmunt, 2018) to self-filter the parallel data, which yielded no useful results. We also randomly split the data into two halves, trained translation models on each half, to score and filter the other half of the data – this method did not work either. We conclude that these methods are not suitable in this scenario where we do not have any clean data, however small, to train the initial cleaning model.

Copied monolingual data We attempted to synthesize training data by copying (Currey et al., 2017) and transliterating[8] monolingual data in the target language to source. In this way, we obtained pseudo parallel data that could potentially improve the decoder side of a translation model without harming the encoder much.

Transfer learning We also explored utilizing dataset from another language in the form of model pre-training. Following Aji et al. (2020), we initialize our Bengali↔Hindi model weights, excluding the embeddings, from our English↔German submission to WMT21 (Chen et al., 2021).

These methods above did not increase BLEU, except that transliterated monolingual data brought a tiny improvement. Model pre-training achieved the convergence faster, but did not achieve better final BLEU. Consequently, we did not carry out any further experiments with these methods.

Acknowledgements

This work used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk) funded by the University of Edinburgh and EPSRC (EP/P020267/1).

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk).

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Gov-

[8] https://github.com/indic-transliteration/indic_transliteration_py

ernment is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

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.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, D. Esch, Nasanbayar Ulzii-Orshikh, A. Tapo, Nishant Subramani, A. Sokolov, Claytone Sikasote, Monang Setyawan, S. Sarin, Sokhar Samb, B. Sagot, Clara Rivera, Annette Rios Gonzales, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, Toan Q. Nguyen, Mathias Muller, A. Muller, S. Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, M. Jenny, Orhan Firat, Bonaventure F. P. Dossou, S. Dlamini, N. D. Silva, Sakine cCabuk Balli, Stella Rose Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, P. Baljekar, Israel Abebe Azime, A. Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. Quality at a glance: An audit of web-crawled multilingual datasets. *ArXiv*, abs/2103.12028.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020a. Facebook AI's WMT20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation*, pages 113–125, Online. Association for Computational Linguistics.

Pinzhen Chen, Nikolay Bogoychev, and Ulrich Germann. 2020b. Character mapping and ad-hoc adaptation: Edinburgh's IWSLT 2020 open domain translation system. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 122–129, Online. Association for Computational Linguistics.

Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. The University of Edinburgh's English-German and English-Hausa submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, M. Douze, H. Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *ArXiv*, abs/1612.03651.

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

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

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, Brussels, Belgium. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1101–1109, Beijing, China. Coling 2010 Organizing Committee.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.