UdS-DFKI Participation at WMT 2019: Low-Resource (*en–gu*) and Coreference-Aware (*en–de*) Systems

Cristina España-Bonet
Saarland University
DFKI GmbH

Dana Ruiter
Saarland University

Josef van Genabith
Saarland University
DFKI GmbH

druiter@lsv.uni-saarland.de
{cristinae,Josef.Van_Genabith}@dfki.de

Abstract

This paper describes the UdS-DFKI submission to the WMT2019 news translation task for Gujarati–English (low-resourced pair) and German–English (document-level evaluation). Our systems rely on the on-line extraction of parallel sentences from comparable corpora for the first scenario and on the inclusion of coreference-related information in the training data in the second one.

1 Introduction

This document describes the systems and experiments conducted to participate in the news translation tasks of WMT 2019 for Gujarati–English (*gu–en*, low-resourced language pair) and German–English (*de–en*, document-level evaluation). We use different approaches to tackle each setting.

Machine translation (neural, statistical or rule-based), usually operates on a sentence-by-sentence basis. However, when translating a coherent document, surrounding sentences may contain information that needs to be reflected in a local sentence. In our experiments for the **document-level** task in *en2de*, we explore how the information beyond sentence level can be made available to a neural machine translation (NMT) system by modifying —tagging— the data in order to include this knowledge. In a similar way, multilingual NMT systems have already been successfully built by only tagging the source data with the knowledge of the target language (Johnson et al., 2017; Ha et al., 2016). With this approach, we incorporate the knowledge that carries coreferences through a text in every sentence. We expect to improve the translation of ambiguous items such as pronouns in English, so we just tackle a specific number of problems and not translation quality in general.

The approach for the **low-resource** setting is completely different. In this case, we use a neural architecture that allows us to extract parallel data from comparable corpora and filter noise from the available parallel data. The additional data obtained in this way is then used to train SMT models, which we compare to a baseline trained on the available parallel data only to observe the effects of the extraction and filtering.

Below, we describe our coreference-aware system for *en2de* (Section 2) and our low-resourced approach for *en–gu* (Section 3). Finally we summarise our findings in Section 4.

2 Coreference-Aware English-to-German System

2.1 Data Preparation

Our system makes use of the annotation of coreference mentions through documents in the source side of the corpus. Documents are annotated with coreference chains using a neural-network-based mention-ranking model as implemented by the `Stanford CoreNLP` tool (Manning et al., 2014)[1]. The tool detects pronominal, nominal and proper names as mentions in a chain. For every mention, `CoreNLP` extracts its gender (male, female, neutral, unknown), number (singular, plural, unknown), and animacy (animate, inanimate, unknown). This information is not added directly but used to enrich the MT training data by applying a set of heuristics implemented in `DocTrans`[2]:

- We enrich *pronominal mentions* with the head of the chain

[1] This system achieves a precision of 80% and recall of 70% on the CoNLL 2012 English Test Data (Clark and Manning, 2016).

[2] https://github.com/cristinae/DocTrans/

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 183–190
Florence, Italy, August 1-2, 2019.

  - Pronoun "I" is not enriched with any coreference information
  - We clean the head by removing articles and Saxon genitives and we only consider heads with less than 4 tokens in order to avoid enriching a word with a full sentence

- We enrich *nominal mentions* including *proper names* with the gender of the head
- The head itself is enriched with she/he/it/they depending on its gender and animacy

The example below shows how we tag the cleaned version of the head of the chain (*fish skin*) before a pronominal mention (*it*):

baseline:
I never cook with it.
coref:
I never cook with <*b_crf*> *fish skin* <*e_crf*> it.

In order to be able to do this processing, we need documents and that limits the amount of corpora we can use. Even though all the corpora made available for the shared task have document boundaries, ParaCrawl, for instance, has a mean of 1.06 sentences per document which makes it useless within our approach.

2.2 Corpus

Monolingual corpora. We use a subset of the NewsCrawl corpus in English and German (years 2014, 2017 and a part of 2018, named as *ssNewsCrawl* in Table 1) to calculate word embeddings as explained in Section 2.3. We first use *langdetect*[3] to extract only those sentences that are in the desired language and compile the final corpora to have a similar number of subword units (Sennrich et al., 2016a) in both languages and years ($\sim 4.10^9$). The corpus is further cleaned, tokenised, truecased (with Moses scripts[4]) and BPEd (with subword-nmt[5]). The vocabulary of the BPE model depends on the system and is detailed in Section 2.3.

Parallel corpora. Due to the restrictions explained in Section 2.1, we use the parallel corpora made available for the shared task in different proportions. Our *base* system uses CommonCrawl, Europarl, News Commentary and Rapid Corpus. Our *large* system also uses the ParaCrawl corpus but in a diluted way. The purpose of the dilution is to try to minimise the fact that due to the nature of our system we cannot use single sentences (intrasentence dependencies are already learned by an NMT system) or back-translations (quality is not good enough to extract coreference chains in a source sentence that is an automatic translation).

	# lines	Small	Large
Monolingual			
ssNewsCrawl en	176,220,479	x1	x1
ssNewsCrawl de	220,443,585	x1	x1
Parallel			
CommonCrawl	2,394,878	x1	x4
Europarl	1,775,445	x1	x4
NewsCommentary	328,059	x4	x16
Rapid	1,105,651	x1	x4
ParaCrawlFiltered	12,424,790	x0	x1

Table 1: Number of lines of the monolingual and parallel corpora used in the $en2de$ translation systems for the *base* and *large* configurations. The second and third columns show the amount of oversampling (or dilution) used in both cases.

CommonCrawl, Europarl and News Commentary are cleaned, tokenised, truecased and BPEd with the same tools as the monolingual corpus. For the Rapid corpus, we performed an additional cleaning: since some German sentences were missing umlauts, we removed all the sentences that contained any word clearly missing an umlaut such as *europishen* or *erklrte*. For ParaCrawl, we first removed sentence pairs that were not detected as English and German sentences by *langdetect* and afterwards we removed sentences with emoji, bullets, and specific tokens such as *http*, *pdf*, €, or *hotel*, etc. With this, we reduce the corpus size by more than half of the sentences. The final number of sentences for all the corpora used for training are provided in Table 1. Notice that we do oversampling for the News Commentary corpus as it is supposed to have a similar domain to the test set.

2.3 Neural Machine Translation Systems

Our NMT systems are trained using the transformer architectures implemented in the `Marian` toolkit (Junczys-Dowmunt et al., 2018). We use two architectures *base* and *big* as defined in

[3] https://pypi.org/project/langdetect/
[4] https://github.com/moses-smt/mosesdecoder/tree/master/scripts
[5] https://github.com/rsennrich/subword-nmt

Vaswani et al. (2017):

Transformer base. 6-layer encoder–decoder with 8-head self-attention, a 2048-dim hidden feed-forward, and 512-dim word vectors. Growing learning rate from 0 to 0.0003 till update 16,000 (warmup). Decaying learning rate afterwards. Adam optimisation with β_1=0.9, β_2=0.98 and ϵ=1e-09. Tied target embeddings.

Transformer big. As *Transformer base* but with word embeddings with 1024-dim, 4096-dim hidden feed-forward layers, learning rate of 0.0002 with the same warmup and decay. β_2=0.998.

Using these architectures as basis, we train several models on 4 TITAN X GPUs using an adaptive batch size that differ on:

- Corpus size. Small vs. Large as defined in Table 1
- Vocabulary. Joint *en–de* BPE with 40K subword units (*join*) vs. separated vocabularies with 50K subword units each (all the other models).
- Initial word embeddings. Source and target initialisation with monolingual embeddings estimated with word2vec[6] (Mikolov et al., 2013) (*Emb*) vs. source and target initialisation with bilingual embeddings mapped using vecmap[7] (Artetxe et al., 2017) (*EmbMap*) vs. no initialisation (all the other models).
- Annotation. No annotation (*Baseline*) vs. tags with coreference information (all the other models).
- Ensembling. Combinations of the previous models at decoding time.

The terms in parenthesis refer to the models in Table 2. Model names are structured as `architectureVocabulary-Annotation-Embeddings-Corpus`.

2.4 Results

Table 2 shows the BLEU scores of the different models and ensembles on newstest-2017 (validation) and news-test2018 (test). The first block presents the results of a baseline system without any document-level information; the second block shows the models explored to determine the best configuration; and the third block summarises the ensembling combinations explored in order to chose our primary submission.

Model	news17	news18
Baseline		
M01:trBig-Baseline-Small	25.82	37.62
M02:trBig-Baseline-Large	27.07	40.38
Coreference-Aware		
M03:trBase-Join-Small	20.00	29.08
M04:trBase-Small	24.74	36.56
M05:trBase-Large	26.35	38.74
M06:trBase-Emb-Large	16.15	22.20
M07:trBase-EmbMap-Large	26.72	39.12
M08:trBig-Small	25.85	37.55
M09:trBig-Large	26.38	38.53
M10:trBig-EmbMap-Large	26.33	39.12
M11:trBig-2-Large	27.42	40.07
M12:trBig-2-EmbMap-Large	27.28	40.28
Ensembling		
M05-M07-M10	27.18	40.92
M07-M09	27.29	40.10
M05-M07-M09	27.24	40.56
M05-M07-M09-M10	27.31	40.98
M05-M07-M10-M11	27.58	41.58
M07-M10-M11-M12	**27.62**	**42.82**

Table 2: BLEU scores of the models trained for the *en2de* translation task. The boldfaced ensembled model was submitted as the primary submission; the best performing model with boldfaced BLEU scores was not ready at submission time.

The first thing to notice is that in terms of BLEU systems with and without **coreference annotations** are not significantly different (M01 vs. M08; M02 vs. M09/M11). Since we are modifying only specific aspects of the translation —few words in a document—, we do not obtain large improvements according to automatic evaluation measures, but we expect differences in translation quality according to human evaluators.

The **vocabulary** turned out to be critical. A system with a joint vocabulary of 40K subword units (M03) is 5-6 BLEU points below its counterpart with 50k units and independent vocabularies (M04).

Embeddings are not that decisive. An initialisation of the system using bilingual embeddings slightly improves the results (M07 vs. M05; M10 vs. M09; M12 vs.M11). Using monolingual embeddings implies a very slow training. M06 in

[6] https://github.com/tmikolov/word2vec
[7] https://github.com/artetxem/vecmap

Table 2 is 10 BLEU points below its counterpart with bilingual embeddings (M07), but the training was far from converging even when running for more days.

As expected, increasing the **size of the corpus** and the number of parameters of the **architecture** is beneficial for the final translation quality. The former has the only disadvantage of needing more time and computing power. The latter even if achieving around 2 BLEU points of improvement (M04 vs. M05; M08 vs. M09) does not allow us to use document level information during training for part of the data.

An **ensemble** of different high performing models showed better results than the combination of the last check-points of the best model. Different combinations are reported in Table 2, all of them using a beam search of size 10 which also performed better than the default value of 6. The best ensemble comes from the combination of the four best performing individual models, but unfortunately the two best performing models were not ready at submission time. M11 and M12 are the same as M09 and M10 before convergence and were the ones used in the ensembled translation as our primary submission.

3 English–Gujarati Systems

3.1 Corpus

Monolingual corpora. The monolingual corpora were used mainly as additional data for training word-embeddings in *en* and *gu*. For English we use the same *NewsCrawl* selection as for *en–de* (ssNewsCrawl). For Gujarati we use the 2018 version of *NewsCrawl* and *CommonCrawl*.

To further increase the available data size for training Gujarati embeddings as well as to add similar content to the English word embeddings, we crawled additional Gujarati news pages and, if existent, their English counterparts. This yielded an increase of about 2 M monolingual Gujarati sentences. While crawling for the news articles, articles written during the period from which the test corpus *newstest2019* was created[8] were not included in the creation of these data sets. The number of sentences and tokens extracted from each news outlet is shown in Table 3.

Wikipedia (WP) is a popular source for comparable documents. In order to later extract parallel sentences from it, the WP dumps[9] for English and Gujarati are downloaded. Only the subset of articles that are linked across both languages using Wikipedia's *langlinks* are extracted. That is, an article is only taken into account if there is a linked article in the other language. For these purposes, we use `WikiTailor` (Barrón-Cedeño et al., 2015)[10] to obtain the intersection of articles of both languages. We additionally use the *en–gu* WP *reference* which was made available for WMT 2019. The monolingual WP in Gujarati is added to the monolingual data for training the embeddings.

Parallel corpora. We use the concatenation of several parallel corpora available for the *en–gu* news translation task to train the base model. Firstly, the *bible* corpus[11] as well as two corpora specially made for WMT2019[12] are used, namely a crawled corpus (WMT19 Crawl) and a localisation corpus extracted from OPUS[13] (WMT Localisation). Lastly, the *Translation Quality Estimation* (TQE) dataset for Indian languages (Nisarg et al., 2018), which essentially is the concatenation of two corpora by the *Indian Languages Corpora Initiative*, which focus on the health and tourism domain each. For development, we use the first 999 sentences from the English-Gujarati version of *newsdev2019*. Further, we report results on the final *newstest2019* corpus.

Pre-processing. All English corpora (excluding the evaluation corpora) undergo the same preprocessing. After being sentence split, the corpora are normalized, tokenized and truecased using standard Moses scripts (Koehn et al., 2007a). A byte-pair-encoding (BPE) (Sennrich et al., 2016b) of 40 k merge operations trained jointly on *en–gu* data respectively is applied accordingly. Duplicates are removed and sentences with more than 50 tokens are discarded. In order to enable a multilingual setup, language tokens indicating the designated target language are prepended to each source sentence. As the English–Gujarati setting is bilingual, this reduces to each Gujarati sentence starting with the language token `<en>`, and each English sentence with `<gu>`.

Gujarati corpora are normalized and romanized

[8] September-November 2018

[9] Downloaded from https://dumps.wikimedia.org/ on January 2019.

[10] https://github.com/cristinae/WikiTailor

[11] http://christos-c.com/bible/

[12] http://www.statmt.org/wmt19/translation-task.html

[13] http://opus.nlpl.eu/

	# sentences
Monolingual	
ssNewsCrawl *en*	176,220,479
CommonCrawl *gu*	3,729,406
NewsCrawl *gu*	244,919
WP Edition *gu*	4,280,531
Crawled	
Divya Bhaskar *gu*	563,072
News18 *en*	460,097
News18 *gu*	193,455
Gujarat Samachar *gu*	121,349
Sandesh *gu*	892,196
Zeenews *en*	466,449
Zeenews *gu*	244,191
Parallel	
Bible *en–gu*	7,807
WMT19 Crawl *en–gu*	10,650
WMT19 Localisation *en–gu*	107,637
TQE *en–gu*	50,000
WP Reference *en–gu*	18,033
Comparable	
WP Comparable *en*	546,924
WP Comparable *gu*	143,120

Table 3: Size of the corpora used for the *en–gu* models.

using the `Indic NLP Library`.[14] The romanized corpora are then tokenized using Moses. As the romanization is case sensitive, no true-casing is performed. The shared BPE is applied.

Cross-lingual word embeddings. We initialize the unsupervised NMT model using *cross-lingual embeddings*. These are trained using monolingual data only. For the English embeddings, we use *ssNewsCrawl*, as well as the English crawled data. For Gujarati all Gujarati data available in Table 3 is used. The initial monolingual embeddings (of size 512) are trained using `word2vec`[15]. The two embeddings are then projected into a common multilingual space using `vecmap`[16] (Artetxe et al., 2017) . We extract all numerals that occur in both monolingual corpora in order to supply a small seed dictionary for training that is not linguistically motivated. After having projected the embeddings into the same space, they are merged into a single cross-lingual embedding. Whenever a word in the two languages is a homograph, one of the two was chosen randomly.

3.2 Neural Machine Translation System

For training our models, we use both SMT and a transformer architecture. While the SMT is used to provide a first model for back-translations as well as to train the final model submitted, the transformer is used in-between to extract additional data from Wikipedia.

The transformer is trained using `OpenNMT-py` (Klein et al., 2017) and is defined as follows: 6-layer encoder-decoder with 8-head self-attention and 2048-dim hidden feed-forward layers. Adam optimization with λ=2 and $beta2$=0.998; *noam* learning rate decay (as defined in Vaswani et al. (2017)) with 8000 warm-up steps. Labels are smoothed (ϵ=0.1) and a dropout mask (p=0.1) is applied. As is common for transformers, position encodings and *Xavier* parameter initialization (Glorot and Bengio, 2010) are used.

3.3 Statistical Machine Translation System

The second family of systems we use in this setting is statistical machine translation (SMT). We expect these systems to perform better when the number of parallel sentences is small. SMT systems are trained using standard freely available software. We estimate a 5-gram or 4-gram language model using interpolated Kneser–Ney discounting with `SRILM` (Stolcke, 2002) depending on the language and the size of the monolingual corpus. Word alignment is done with `GIZA++` (Och and Ney, 2003) and both phrase extraction and decoding are done with the `Moses` package (Koehn et al., 2007b). The optimisation of the feature weights of the model is done with Minimum Error Rate Training (MERT) (Och, 2003) against the BLEU (Papineni et al., 2002) evaluation metric. Our model considers the language model, direct and inverse phrase probabilities, direct and inverse lexical probabilities, phrase and word penalties, and a lexicalised reordering.

3.4 Results

We train our SMT and NMT in four steps, yielding the following models:

1. **SMT**$_{\text{base}}$: Train an SMT model on the concatenation of all parallel training data listed in Table 3 ($\sim$194 k pairs). This is then used to back-translate 4 k (2 k per language direction) pairs of the monolingual data available.

2. **NMT**$_{\text{extract}}$: Initialize Transformer with the pre-trained word-embeddings. The transformer is used to extract additional data from *en–gu* Wikipedias as well as the crawled

[14] https://github.com/anoopkunchukuttan/indic_nlp_library

[15] https://github.com/tmikolov/word2vec

[16] https://github.com/artetxem/vecmap

Reference	BLEU dev		BLEU nt2019	
	en2gu	*gu2en*	*en2gu*	*gu2en*
$\text{NMT}_{\text{extract}}$	4.65	10.64	3.10	8.60
SMT_{base}	8.77	12.90	6.90	10.20
$\text{SMT}_{\text{extract}}$	9.15	13.08	**6.90**	10.50
SMT_{all}	8.93	14.08	7.10	**10.80**

Table 4: BLEU scores achieved on the internal development set and the official *newstest2019*. Scores on the development set are calculated using `multi-bleu` on the tokenized outputs, while the results on *newstest2019* are those calculated by the WMT matrix. Primary system submissions are in bold.

Zeenews and *News18* articles. It is also used to filter the back-translations produced by SMT_{base} as well as the parallel corpus available. The extraction is performed using the joint NMT learning and extraction framework described in Ruiter et al. (2019). There, we use the margin-based function (Artetxe and Schwenk, 2018) for scoring both word embedding and hidden-state representations. This results in an extracted and filtered corpus of $\sim$275 k sentences; a slight increase to the original parallel data available to us despite the filtering of less useful pairs.

3. **SMT**$_{\text{extract}}$: SMT model, trained on the corpus that resulted from the extraction and filtering performed by $\text{NMT}_{\text{extract}}$.

4. **SMT**$_{\text{all}}$: SMT model, trained on both the extracted and filtered corpus by $\text{NMT}_{\text{extract}}$, as well as the parallel data available, resulting in $\sim$475 k training pairs used.

Due to time constraints we could not apply any system combination technique on the individual systems. However, due to the big gap in performance between SMT and NMT we do not expect significant improvements.

Table 4 shows translation quality as measured by BLEU for both the neural and statistical systems with the different data configurations.

The filtering and extraction performed by $\text{NMT}_{\text{extract}}$ led to a small increase in BLEU for $\text{SMT}_{\text{extract}}$ and SMT_{all}, indicating that the filtering was based on positive decisions. However, when taking into account that the average number of extracted pairs from WP was steadily around 1.6 k pairs, and comparing them with the 18 k pairs in the *en–gu* WP reference, it becomes clear that extraction did not obtain high recall. This is most likely due to three difficulties that the system encounters in this setting: i) Not enough comparable data was available to adapt the internal representations (word embeddings and hidden states) to the data, meaning that the extraction performance, which is bound to the extraction decisions of the representations, stays below its potential. ii) The lack of monolingual data to train high-quality *gu* embeddings as well as iii) the rareness of homographs in this rather distant language pair makes the initialization difficult. Extraction in the first epochs is usually dependent on such homographs and a lack thereof reduces the number of identifiable pairs in the initialization phase of the model.

4 Conclusions

We presented two approaches for the WMT 2019 news translation shared task. We participated in the *en2de* task with a data-based coreference-aware NMT system. The corpus is enriched with this document-level information at sentence level so that the standard training procedure can be used. However, the amount of data we can use is smaller than in the standard pipeline and therefore the global quality can be damaged. We expect the manual evaluation to show improvements on the tackled phenomena such as gender translation.

For the *en-gu* task, we used a NMT architecture that can be trained on comparable corpora. In this case we downloaded news web pages as well as linked Wikipedia articles in Gujarati and English to extract and train on. Our experiments show that very few sentences could be used from this corpus and our results are close to the baseline one can get with the available parallel resources. Given the final amount of data, our state-of-the-art SMT system performed clearly better than our NMT one.

Acknowledgments

The project on which this paper is based was partially funded by the German Federal Ministry of Education and Research under the funding code 01IW17001 (Deeplee) and by the Leibniz Gemeinschaft via the SAW-2016-ZPID-2 project (CLuBS). Responsibility for the content of this publication is with the authors.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost)

no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe and Holger Schwenk. 2018. Margin-based parallel corpus mining with multilingual sentence embeddings. *arXiv preprint arXiv:1811.01136*.

Alberto Barrón-Cedeño, Cristina España-Bonet, Josu Boldoba, and Lluís Màrquez. 2015. A Factory of Comparable Corpora from Wikipedia. In *Proceedings of the Eighth Workshop on Building and Using Comparable Corpora*, pages 3–13. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the International Workshop on Spoken Language Translation*, Seattle, WA.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. 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 Linguist*, 5:339–351.

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.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 67–72.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007a. 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. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007b. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session*, pages 177–180.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*.

Jhaveri Nisarg, Manish Gupta, and Vasudeva Varma. 2018. Translation quality estimation for indian languages. *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 159–168.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association of Computational Linguistics*, pages 311–318.

Dana Ruiter, Cristina España-Bonet, and Josef van Genabith. 2019. Self-supervised neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (ACL 2016), Volume 1: Long Papers*, pages 1715–1725, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (ACL 2016), Volume 1: Long Papers*, pages 1715–1725.

A. Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904.

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 *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.