Prompsit's submission to WMT 2018 Parallel Corpus Filtering shared task

Víctor M. Sánchez-Cartagena, Marta Bañón,
Sergio Ortiz-Rojas, Gema Ramírez-Sánchez
Prompsit Language Engineering
Av. Universitat s/n. Edifici Quorum III
E-03202 Elx, Spain
{vmsanchez,sortiz,gramirez,mbanon}@prompsit.com

Abstract

This paper describes Prompsit Language Engineering's submissions to the WMT 2018 parallel corpus filtering shared task. Our four submissions were based on an automatic classifier for identifying pairs of sentences that are mutual translations. A set of hand-crafted hard rules for discarding sentences with evident flaws were applied before the classifier. We explored different strategies for achieving a training corpus with diverse vocabulary and fluent sentences: language model scoring, an active-learning-inspired data selection algorithm and n-gram saturation. Our submissions were very competitive in comparison with other participants on the 100 million word training corpus.

1 Introduction

This paper describes the systems submitted by Prompsit Language Engineering[1] to the parallel corpus filtering shared task (Koehn et al., 2018) featured in the Third Conference on Machine Translation (WMT 2018).

Given a very noisy 1 billion-word German-English parallel corpus crawled from the web,[2] participants have to subselect sentence pairs that amount to (a) 10 million words (*10M dataset*), and (b) 100 million words (*100M dataset*). In this shared task, performance of the sentence filtering is estimated as the translation quality (as measured by BLEU) of phrase-based statistical machine translation (SMT) and neural machine translation (NMT) systems built from the subselected data. Evaluation sets belong to different domains, which discourages strategies based on domain relatedness.

Our submission is built upon the assumption that a training set that maximizes the quality of machine translation (MT) must meet the following requirements:

- Parallel sentences must be mutual translations.
- Sentences must be *fluent* in the corresponding language in order to build a reliable language model/NMT decoder. We work under the hypothesis that the sentence `D0006 Tooth brush A NOELL / F945J 0,21` is less useful for a language model than `I brush my teeth and look in the mirror`, despite containing a similar amount of tokens.
- Vocabulary must be diverse, since the MT systems are evaluated with test sets from different domains.

We built a training corpus that meets the aforementioned requirements in a sequential process that comprises the following steps:

1. As a preprocessing step, deletion of parallel sentences by means of a set of hand-crafted *hard rules* implemented in the translation memory cleaning tool `Bicleaner`.[3] These rules are addressed at detecting evident flaws such as languages different from English and German, encoding errors, very different lengths in parallel sentences, etc. and speeding up the subsequent steps.
2. Detection of misaligned parallel sentences by means of an automatic classifier.
3. Scoring of sentences based on fluency and diversity: four different approaches were tested and submitted.

[1] http://www.prompsit.com

[2] As part of the Paracrawl project: https://paracrawl.eu/.

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

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 955–962

https://doi.org/10.18653/v1/W18-64115

The remainder of the paper is organized as follows: Section 2 outlines related approaches, Sections 3 and 4 respectively describe the steps 2 and 3 of our processing pipeline. Section 5 confirms the positive impact of our processing pipeline on translation quality by comparing it with other baseline approaches. Finally, the paper ends with some concluding remarks and the suggestion of potential future research directions.

2 Related work

The WMT 2018 parallel corpus filtering shared task partially shares its objectives with the First Automatic Translation Memory Cleaning Shared Task (Barbu et al., 2016), where participants had to automatically classify translation memory segments according to whether the target language (TL) side was translation of the source language (SL) side or not. This task is, in turn, very similar to the detection of parallel sentences in comparable corpora, that can be tackled by combining bilingual data and automatic classifiers (Munteanu and Marcu, 2005), machine translation (Abdul-Rauf and Schwenk, 2009) or, more recently, word embeddings (España-Bonet et al., 2017). In fact, the approach we follow to detect sentences that are mutual translations is similar to the work of Munteanu and Marcu (2005). Their approach differs from ours in the fact that we make use of a larger set of shallow features not related to lexical similarity.

However, since the size of the data sets that participants must produce in this task is smaller than the number of parallel sentences that are mutual translations, this task is also related to the *data selection*: selection of a subset of data that maximizes translation quality, avoiding redundancy and matching a given domain (Eetemadi et al., 2015). Instead of the widespread language-model based data selection methods (Axelrod et al., 2011), we replaced words with placeholders in order to not take into account the domain of the text.

3 Sentence alignment classifier

After applying the hard rules aimed at detecting evident flaws introduced in Section 1, 22 229 462 parallel sentences (21%) out of the initial 104 002 521 were kept. In order to discard pairs of sentences that are not mutual translations, we applied an automatic classifier to the sentence pairs that passed the hard rule filter. The classifier produces a score for each pair of sentences that represents the probability that they are mutual translations. This score is used in different ways depending on the scoring strategy chosen for achieving vocabulary diversity and fluency (see next section).

The features we used can be split in two groups: those that represent the lexical similarity of the two sides of a parallel sentence by making use of probabilistic bilingual dictionaries, and those that are based on shallow properties such as sentence length, capitalized words, punctuation marks, etc.

Given a bilingual probabilistic dictionary whose SL is *L1* and TL is *L2* and a pair of sentences (s_1, s_2), written in languages *L1* and *L2* respectively, we computed the four lexical similarity features described next. The feature `DICT-QMAX-L1` is defined as $\prod_{w \in s_2} \max_{w' \in s_1} p(w', w)$, where $p(w', w)$ is the translation probability from the *L1* word w' to the *L2* word w according to the bilingual dictionary. That is, `DICT-QMAX-L1` is the product, for each word w in s_2, of the maximum translation probability from any word in s_1 to w. The feature `DICT-QMAX-L2` is computed in the opposite direction (with the help of a bilingual dictionary whose SL is *L2* and TL is *L1*). We also used two additional features that account respectively for the proportion of words in s_1 and s_2 that can be found in the bilingual dictionaries.

Shallow features include, among others:

- For each language, probability of the sentence length according to a Poisson distribution, given the sentence length ratio observed in the positive examples of the classifier training set.[4]
- Number of tokens in each segment.
- Average token length (in characters) in each segment.
- Number of punctuation marks in each segment.
- Number of numerical expressions in each segment that can be found in the other segment of the pair.

[4] Let l_s be the length of the SL sentence, l_t the length of the TL sentence and r the average length of TL sentence to length of SL sentence ratio observed in the training corpus. The probability of the TL sentence length is computed as $e^{-l_s r} \frac{l_s r}{l_t !}$.

- Number of capitalized tokens in one segment that can be found in the other segment of the pair.

We trained a Random Forest classifier (Breiman, 2001) with 200 trees and a maximum depth of 2. The remaining parameters were the default ones in the Random Forest implementation of the `Scikit-learn` library.[5]

The bilingual dictionaries were obtained from all the available English–German parallel corpora from WMT 2018 news translation shared task (with the exception of 60 000 sentences randomly removed from *news-commentary-v13*, which were used for training the classifier, as explained in the next paragraph). After concatenating the corpora, they were word-aligned by means of `MGIZA++`,[6] alignments were symmetrized with the heuristic *grow-diag-final* and the probabilities in the bilingual dictionaries were estimated by maximum likelihood from the symmetrized alignments. Before building the dictionaries and computing the lexical features, compounds in German were segmented with the maximum entropy classifier proposed by Dyer (2009).[7]

The training set for the classifier was built as follows. From the 60 000 parallel sentences randomly removed from the *news-commentary-v13* parallel corpus, 50 000 were used for actually training the classifier while the remaining 10 000 were used as a validation set. From the training set, 50 000 positive instances were obtained. 50 000 negative instances were also obtained from the training set, after randomly shuffling their English side, i.e., synthetically generating pairs of sentences that are not mutual translations. The same strategy was built for obtaining negative instances for the validation set. The accuracy of the resulting classifier with the score threshold at 0.5 was 0.98.

4 Scoring for fluency and diversity

From the three main issues that need to be tackled for obtaining a good training corpus for machine translation, the classifier dealt with sentences that are not mutual translations. In this section, we describe the four scoring strategies we submitted to the shared task and how they tackle the two remaining issues: vocabulary diversity and fluency.

4.1 N-gram saturation

This scoring strategy aims to increase the vocabulary diversity by removing sentence pairs that are too similar to other pairs in the training corpus. Each sentence pair is assigned the score returned by the classifier, with the exception of those sentences deemed as too similar, which are discarded. The 10M and 100M datasets are just obtained by selecting the not discarded (not deemed as too similar) sentences, sorted in descending classifier score, until the desired token count is achieved.

Too similar sentences are identified by a simple n-gram saturation algorithm. First, some tokens are replaced with placeholders. Fully alphabetic tokens written either in lowercase (all characters are lowercase) or in titlecase (the first character is uppercase and the remaining ones are lowercase) are kept intact and every other token is replaced with one of the following placeholders:

- `ALPHA:UPPER`: all characters are uppercase.
- `ALPHA:MIXED`: all characters are alphabetic, but the token is neither written in lowercase, nor in titlecase, nor in full uppercase.
- `NUMERIC`: all the characters are digits.
- `PUNCTUATION`: all the characters are punctuation marks.
- `MIXED`: none of the previous conditions are met.

Additionally, titlecased words that can be found in the other sentences of the pair are replaced with `ALPHA:PROPER`.

For instance, the sentence `the Kari EL22 electrode switch is designed for the control of conductive liquids .` becomes `the ALPHA:PROPER MIXED electrode switch is designed for the control of conductive liquids PUNCTUATION` after the replacement is made.[8]

Once placeholders are introduced in sentences, sentence pairs are traversed in descending classifier score order, and those whose full set of 4-grams can be found in sentences with higher

[5] http://scikit-learn.org/

[6] https://github.com/moses-smt/mgiza.git

[7] https://github.com/redpony/cdec/tree/master/compound-split; pre-trained models from this implementation were used.

[8] The word `Kari` also appears in the German sentence and it is thus considered as a proper noun.

scores are classified as too similar and discarded. Placeholders prevent sentences which differ from other sentences only in proper nouns, codes, figures, punctuation, etc. from being accepted.

The number of sentences retained after applying n-gram saturation was 10 100 275, from which the top 433 760 and the top 5 121 715 with the highest classifier scores were respectively selected to build the 10M and 100M datasets.

4.2 Active learning data selection

A potential limitation of the scoring strategy based on n-gram saturation is that, when building the 10M word training set, a large proportion of the sentences which passed the saturation filter were not considered. From the 6 798 687 sentences resulting from applying n-gram saturation with a classifier score above 0.5 (i.e., very likely to be mutual translations), 433 760 were greedily chosen without even considering the remaining ones. These sentences could contain useful words or expressions that have been ignored.

In order to overcome that limitation, we designed a data selection strategy that considers the vocabulary of the whole corpus. Our approach is an adaptation of the active learning strategy used for building training corpora for SMT proposed by Haffari et al. (2009) and it is outlined in Algorithm 1. This algorithm is applied only to sentences with a classifier score ≥ 0.55; those below that score are discarded.

Algorithm 1 Data selection via active learning

Require: Bilingual corpus C
Ensure: Sorted bilingual corpus S
  $S \leftarrow \emptyset$
  $blocksize \leftarrow 100\,000$
  while $|C| > 0$ **do**
    $S_{new} \leftarrow select(C, S)$
    $C \leftarrow C - S_{new}$
    $S \leftarrow S + S_{new}$
    $blocksize \leftarrow increaseBlockSize(blocksize)$
  end while

It iteratively selects a sequence of sentence pairs S_{new} and appends it to the sorted corpus S until no sentences are available in the corpus C. The function $select(C, S)$ scores the sentences in C with the *Geom n-gram* function (Haffari et al., 2009, Sec. 3.1.2), sorts them by decreasing score, applies the n-gram saturation filter described previously (with a small modification: a sentence pair is discarded if at least half of the 4-grams have been observed in not discarded sentence pairs from C with higher score) and returns the top $blocksize$ sentences. The *Geom n-gram* scoring function assigns the highest scores to sentences with n-grams that are frequent in C and infrequent in S. The function $increaseBlockSize$ doubles the block size every 5 iterations. The datasets were built by traversing the sorted corpus S until desired token counts were achieved.

4.3 Language modeling

While the two previous approaches aimed at increasing the diversity of the vocabulary, the corpora selected following these approaches may contain pairs of sentences that are not useful to build a powerful language model, such as: `Brush for Acrylic - blue #06` ↔ `Pinsel für Acryl Falten - Rot #6`.

In order to include only *fluent* sentences in the training sets, we made use of language models. As we did not want to include a bias towards news data in the language models, placeholders were used in a similar way to what has been described in Section 4.1. The following types of tokens were replaced with placeholders:

- Tokens made fully of alphabetical characters. They were replaced with a placeholder that represents its capitalization: lowercase (`ALPHA:LOWER`), titlecase (`ALPHA:TITLE`), uppercase (`ALPHA:UPPER`) or mixed case (`ALPHA:MIXED`).
- Tokens made fully of numeric characters (`ALPHA:NUM`).
- Tokens that contain a numeric or alphabetical character but do not fall into any of the two previous groups (`MIXED`).

Consequently, tokens made only of punctuation characters were kept unchanged. The previous pair of sentences was hence processed as follows: `ALPHA:TITLE ALPHA:LOWER ALPHA:TITLE - ALPHA:LOWER MIXED` ↔ `ALPHA:TITLE ALPHA:LOWER ALPHA:TITLE ALPHA:TITLE - ALPHA:TITLE MIXED`

Each 5-gram language model (one for each language) was estimated from 20 000 000 sentences randomly chosen from the news and Europarl

monolingual corpora with KenLM (Heafield, 2011) and Knesser-Ney smoothing (Heafield et al., 2013).

Language models were used to score pairs of sentences as follows:

1. Pairs of sentences with a classifier score lower than 0.55 were discarded.

2. Remaining pairs of sentences were sorted in ascending sum of (English plus German) perplexity per word.

3. The n-gram saturation algorithm described in Section 4.1 was applied. As similar sentences have similar perplexities, the algorithm is needed in order to decrease the degree of repetition in the resulting corpus.

Two submissions were based on language model scoring. In the first one, `prompsit-lm`, sentences were truecased before training the language model and the saturation algorithm was applied exactly as described in Section 4.1, i.e. with the same placeholder replacement strategy. In the alternative submission, `prompsit-lm-nota`, sentences were not truecased for language model scoring and the saturation algorithm was applied without placeholder replacement.[9]

In the submission `prompsit-lm`, 5 868 776 sentences passed the n-gram saturation filter, from which the 4 492 314 sentence pairs with the lowest perplexity per word were selected for building the 100M tokens training set. In the submission `prompsit-lm-nota`, since the saturation filter is less aggressive, 7 016 169 sentence pairs passed that filter and 4 491 269 were selected for the 100M tokens training set.

5 Machine translation experiments

We built MT systems from the four scoring alternatives presented and compared them with two baseline systems: one in which the sentences were randomly chosen from the noisy, crawled data (`random`) and another one in which the hard-rule filtering was applied and each sentence was simply scored by the classifier (`only-classifier`; 10M and 100M datasets were built by selecting sentences in descending classifier score order).

Systems were trained following the official instructions from the shared task.[10] SMT systems were built with Moses and tuned with Batch MIRA (Cherry and Foster, 2012). A 5-gram language model was estimated from the TL side of the training corpus. NMT systems followed the Transformer architecture (Vaswani et al., 2017) and were built with Marian (Junczys-Dowmunt et al., 2018). 49 500 byte pair encoding merge operations (Sennrich et al., 2016) were applied to segment the words in the NMT training corpus. The development set (used for tuning the parameters of the log-linear model in SMT and for early stopping in NMT) was *newstest2016*, while the test set was *newstest2017*. Table 1 presents the (cased) BLEU scores obtained by the MT systems built.

It can be observed that the scores of NMT systems trained on random subsamples (`random` baseline) are very low if we compare them with SMT. This confirms that NMT is very sensitive to noisy training data (Belinkov and Bisk, 2017). An important increase in BLEU for all systems can be observed when filtering with hard rules and classifier (`only-classifier` system). After this filtering, NMT outperforms SMT for both training set sizes.

Concerning our submissions, results show that adding n-gram saturation (`prompsit-sat`) slightly improves the results in the four datasets, which confirms that vocabulary diversity is relevant for this task. We can also observe in Table 3 that the number of unknown words in the test set was slightly reduced. Our active learning strategy for achieving vocabulary diversity (`prompsit-al`), however, brought a degradation in the 10M dataset and a light improvement in the 100M one. If we analyze vocabulary sizes (displayed in Table 2), it was reduced (in comparison with `prompsit-sat`) only for the 10M dataset, and the number of unknown words in the test set increased. A potential solution for this issue could be reducing the block size for the first iterations of the active learning algorithm, so that more itera-

[9] Note that, in the `prompsit-lm` submission, two different placeholders replacement strategies were applied. Firstly, that described in Section 4.3 was applied in order to obtain language model perplexities. Afterwards, the one described in Section 4.1 was applied in order to discard similar sentences. In the `prompsit-lm-nota` submission, only the first one was applied. Concerning truecasing, preliminary experiments showed that it has a limited impact for language model scoring, hence the main difference between the submissions is the strength of n-gram saturation: fewer sentences are discarded if placeholders are disabled.

[10] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

System	SMT 10M	SMT 100M	NMT 10M	NMT 100M
random	14.92	18.51	7.70	7.66
only-classifier	20.22	23.96	21.46	29.32
prompsit-sat	20.77	24.12	22.82	29.55
prompsit-al	20.02	24.46	22.50	29.64
prompsit-lm	19.09	24.37	18.50	29.79
prompsit-lm-nota	18.61	24.36	18.60	29.85

Table 1: BLEU scores obtained by our 4 submissions and two baseline approaches.

tions are executed before obtaining the 10M training set.

The submissions that aimed at increasing the fluency of the training corpus brought a light improvement in translation quality for the NMT system trained on the 100M dataset. On the contrary, they further reduced the vocabulary sizes and increased the unknown rate for the 10M dataset. We believe this is due to the fact that, with this approach, fluency had a stronger influence than vocabulary diversity in the criterion for selecting sentences for the small dataset. Only the top 836 520 sentences with smallest perplexity were explored for building the final 10M training corpus obtained with `prompsit-lm`, which contained 551 098 sentences.[11] A manual inspection of the sentences included in the 100M dataset but not in the 10M one showed that they were perfectly fluent. This means fluent sentences which are more interesting (from a vocabulary point of view) have been ignored when building the 10M dataset, since the process is mainly guided by perplexity. This problem disappears in the large data set, that is large enough to contain diverse vocabulary.

The BLEU scores reported in this section do not exactly match those published in the official results (Koehn et al., 2018) because, unlike the scores reported in this paper, the official scores were averaged over multiple training runs and multiple evaluation corpora. Nevertheless, the relative performance of our four submissions remains the same. Our active learning and language model scoring strategies were very competitive for the 100M dataset and were ranked very close to the top performing systems, while our best performing submissions for the 10M dataset were in the middle of the ranking.

[11] The difference between these two numbers is the amount of sentences removed by the n-gram saturation algorithm.

6 Concluding remarks

This paper described Prompsit Language Engineering's submissions to the WMT 2018 parallel corpus filtering shared task. Our four submissions stemmed from a strategy based on handcrafted filtering rules and an automatic classifier that selects those sentences that are mutual translations. Our submissions explored different ways of achieving vocabulary diversity and fluency in the selected training corpora. The strategies based on an active learning algorithm (aimed at achieving vocabulary diversity) and language model perplexity combined with n-gram saturation (aimed at achieving fluency and vocabulary diversity) allowed our submissions to be ranked close to the top performing system for the 100M dataset.

Our strategies were less successful for the 10M tasks, as they were placed in the middle of the ranking. An analysis of out of vocabulary words in the test set for the language model-based approaches suggests that fluency has a stronger influence than vocabulary diversity. A scoring scheme that balances them better should improve the results and designing it could be a future research direction. The active learning algorithm could also be tuned for smaller datasets by decreasing the block size parameter.

Acknowledgments

We would like to thank the anonymous reviewers for their suggestions on how to improve the paper. Work funded by EU project 2016-EU-IA-0114 "Provision of web-scale parallel corpora for official European languages".

References

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve smt performance. In *Proceedings of the 12th Conference of the*

System	de 10M	en 10M	de 100M	en 100M
random	904K	789K	3 810K	3 364K
only-classifier	561K	382K	2 197K	1 274K
prompsit-sat	585K	365K	2 246K	1 174K
prompsit-al	403K	228K	2 329K	1 162K
prompsit-lm	359K	99K	2 022K	910K
prompsit-lm-nota	364K	103K	1 969K	879K

Table 2: Vocabulary sizes, expressed in thousands of words, after tokenization with the Moses tokenizer, of the training corpora produced with our four submissions and two baseline approaches.

System	# unks 10M	# types 10M	# unks 100M	# types 100M
random	2 580	2 012	1 132	913
only-classifier	2 852	2 207	1 199	921
prompsit-sat	2 639	2 027	1 148	877
prompsit-al	3 084	2 266	1 114	848
prompsit-lm	4 307	2 744	1 178	882
prompsit-lm-nota	4 183	2 682	1 182	896

Table 3: Unknown words in the source language (German) size of the *newstest2017* test set. The columns labeled as `# unks` represent the number of instances of unknown words, while `# types` stands for the number of unique unknown words.

European Chapter of the Association for Computational Linguistics, EACL '09, pages 16–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eduard Barbu, Carla Parra Escartín, Luisa Bentivogli, Matteo Negri, Marco Turchi, Constantin Orasan, and Marcello Federico. 2016. The first automatic translation memory cleaning shared task. *Machine Translation*, 30(3):145–166.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 427–436, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 406–414, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3):189–223.

Cristina España-Bonet, Ádám Csaba Varga, Alberto Barrón-Cedeño, and Joseph van Genabith. 2017. An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1340–1350.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 415–423, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

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, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31(4):477–504.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

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.