Webinterpret Submission to the WMT2019 Shared Task on Parallel Corpus Filtering

Jesús González-Rubio
WebInterpret Inc.
jesus.gonzalez-rubio@webinterpret.com

Abstract

This document describes the participation of Webinterpret in the shared task on parallel corpus filtering at the Fourth Conference on Machine Translation (WMT 2019). Here, we describe the main characteristics of our approach and discuss the results obtained on the data sets published for the shared task.

1 Task Description

Parallel corpus filtering task at WMT19 tackles the problem of cleaning noisy parallel corpora. Given a noisy parallel corpus (crawled from the web), participants develop methods to filter it to a smaller size of high quality sentence pairs.

In comparison to the German-English task last year, the organizers now pose the problem under more challenging low-resource conditions including Nepali and Sinhala languages. The organizers provide very noisy 40.6 million-word (English token count) Nepali-English and a 59.6 million-word Sinhala-English corpora. Both raw corpora were crawled from the web as part of the Paracrawl project[1]. Participants are asked to select a subset of sentence pairs that amount to (a) 5 million, and (b) 1 million English words. The quality of the resulting subsets is determined by the quality of a statistical and a neural Machine Translation (MT) systems trained on the selected data. The quality of the translation systems is measured on a held-out test set of Wikipedia translations. Despite the known origin of the test set, the organizers make explicit that the task addresses the challenge of data quality and not domain-relatedness of the data for a particular use case.

For our submission, we propose a variation of coverage augmentation ranking (Haffari et al., 2009; Gascó et al., 2012; González-Rubio, 2014). The main idea underlying our approach is to minimize the amount of unseen events for the model. In MT, these unseen events are words or sequences thereof. These unseen events result in a loss of model coverage and, ultimately, of translation quality. The main difference of our submission respect to previous approaches is that we do not rely on an in-domain corpus to identify underrepresented events. Instead, we look for the subset of sentences that provide the most coherent coverage among themselves. One of the advantages of this approach is that it does not rely on pre-trained models requiring additional data to train. This characteristic fits perfectly with the focus on low-resource languages of this year's task.

The rest of this document is organized as follows. First, we describe the details of our approach. Next, we present the results of our submission. Finally, we close with the conclusions and some ideas for future developments.

2 Sentence Pairs Ranking

Our goal is to rank the sentence pairs in the raw corpora such that the pairs in the top of the ranking are better candidates for training data. As pre-processing, we only apply tokenization via the TokTok tokenizer in the NLTK python package.

First, we filtered out some of the pairs $(\mathbf{x}, \mathbf{y})$ in the raw corpus according to several heuristic rules (Section 2.1). Then, for the remaining pairs, we computed a ranking value $r(\mathbf{x}, \mathbf{y})$ for each of them. This ranking, was the result of the combination of several different ranking functions aiming at capturing the ”value” of the sentence pair according to different criteria (Section 2.2 and Section 2.3). Finally, we used the final ranking of each pair to compute its corresponding score as required for the shared task (Section 2.4).

2.1 Initial Rule-based Filtering

We start by describing the set of filtering rules implemented to reduce the amount of candidates to be ranked by the more sophisticated methods Sections 2.2, and 2.3. These rules have been previously proposed and successfully implemented in the literature, for instance (Junczys-Dowmunt, 2018; Rossenbach et al., 2018).

[1] https://paracrawl.eu/

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 271–276
Florence, Italy, August 1-2, 2019. ©2019 Association for Computational Linguistics

	Nepali-English (2.2M)		**Sinhala-English** (3.4M)	
Method	**Sent. pairs**	**Ratio**	**Sent. pairs**	**Ratio**
Language Identification	1.65M	74.0%	2.27M	67.7%
Length Ratio	0.86M	38.6%	1.13M	33.8%
Max. Sentence Length	0.24M	10.9%	0.27M	8.1%
Combined	2.11M	94.4%	2.92M	86.8%

Table 1: Amount of sentence pairs (in Millions) filtered out by each filtering method. ”Combined” denotes the final amount of sentence pairs filtered out after applying the three methods in sequence.

The filtering rules we implemented for our submission are not language specific, and moreover, they only place very mild assumption on what constitutes a ”good” sentence pair. In particular, *maximum sentence length* is a technical restrictions implemented by many MT systems. Given that the translation system is most probably going to ignore them in any case, it makes no sense for us to even rank them. Table 1 displays the amount of sentences pairs filtered out by each method.

Language Identification

We implemented a very straightforward language identification using the Python LangID package. Specifically, we filtered out all those pairs not belonging to the desired pair of languages. For example, each pair $(\mathbf{x}, \mathbf{y})$ in the Nepali-English corpus should satisfy: $\text{LangID}(\mathbf{x}) =$ ”ne” and $\text{LangID}(\mathbf{y}) =$ ”en”, otherwise the sentence pair is filtered out. For Sinhala-English, we require Sinhala as source language: $\text{LangID}(\mathbf{x}) =$ ”si”.

Length Ratio

As our second heuristic filtering, we chose the ratio between the number of tokens of $\mathbf{x}$ and $\mathbf{y}$. This is a very simple criterion, but efficient to identify mispaired sentences. We limited this ratio to be under 1.7 and smoothed the counts by adding 1 to them. That is, we rejected the sentence pair if:

$$\frac{|\mathbf{x}|+1}{|\mathbf{y}|+1} \text{ or } \frac{|\mathbf{y}|+1}{|\mathbf{x}|+1}$$

where $|\mathbf{x}|$ and $|\mathbf{y}|$ are the number of tokens of $\mathbf{x}$ and $\mathbf{y}$ respectively.

Maximum Sentence Length

Most translation systems have an upper bound for the sentence length. These sentences will be ignored in any case during training so we decided to filter them out directly. If either the source $(\mathbf{x})$ or destination $(\mathbf{y})$ sentence in a pair was over 50 tokens, we filtered out the pair.

2.2 Coverage Ranking

Sparse data problems are ubiquitous in MT (Zipf, 1935). In a learning scenario, this means that some rare events will be missing completely from a training set, even when it is very large. Missing events result in a loss of coverage, a situation where the structure of the model is not rich enough to cover all types of input. An extreme case of this are out-of-vocabulary words for which the MT system will have no information on how to translate them. Therefore, words (or sequences thereof) that do not appear in the training set cannot be adequately translated (Haddow and Koehn, 2012; Sennrich et al., 2016).

According to these considerations, we propose to explicitly measure how well represented are the different words on a potential training corpus $\mathcal{T}$ as a proxy of the actual ”value” of such corpus. We define this corpus ”value”, $V(\mathcal{T})$, as:

$$V(\mathcal{T}) = \sum_{s \in \text{tokens}(\mathcal{T})} \frac{\min(N, c(s, \mathcal{T}))}{N} \qquad (1)$$

where function $\text{tokens}(\mathcal{T})$ returns the set of tokens that appear in $\mathcal{T}$, $\text{c}(s, \mathcal{T})$ counts how many times a token s appears in $\mathcal{T}$, and N denotes a count above which we consider a token to be adequately represented. After some initial experiments, we used $N = 50$ in our submission.

In order to rank the different sentences in the raw corpora, we implemented a greedy algorithm to create a training corpus $\mathcal{T}$ by iteratively adding sentences to it taken from a given pool. At start, $\mathcal{T} = \varnothing$ and the pool is equal to the sentences that passed the filtering rules in the previous section. The sentence to be added at each step is the one that resulted in a new $\mathcal{T}$ with the highest value as measured by Equation 1. This selected sentence is then removed from the pool and definitely added to $\mathcal{T}$. This process repeats until the pool is empty.

This algorithm has a complexity of $\mathcal{O}(R^2)$ where R is the number of sentences initially in the pool. In Section 3, we describe how we modify this algorithm for the final submission in order to improve its time performance.

In our submission, we considered as tokens n-grams of sizes from one up to four, and computed them for both the source and destination sentences. This resulted in a total of eight ranks per sentence pair. We denote each of them as $r_{\mathrm{c}}(\mathbf{s}, n)$ where $\mathbf{s} \in \{\mathbf{x}, \mathbf{y}\}$, and $1 \leq n \leq 4$.

The main shortcoming of this ranking scheme is that it ignores how the source and destination sentences in a pair relate to each other. Long sentences with multiple tokens will most surely rank high even when the other sentence in the pair carry completely different meaning. In order to counterbalance these undesired effects, we implement a secondary adequacy ranking to measure such correspondence between the sentences on each pair.

2.3 Adequacy Ranking

This ranking function measures how much of the original meaning is expressed in the translation and vice versa. Specifically, we estimate to which extent the words in the original and translated sentences correspond to each other.

We compute this ranking from a simple (but fast) word-to-word translation model (Brown et al., 1993). Given a sentence pair $(\mathbf{x}, \mathbf{y})$, we compute a *source-given-target* score according to the geometric average probability over the words for the IBM model 1 formulation:

$$P_{\mathrm{M1}}(\mathbf{x}, \mathbf{y}) = \sqrt[|\mathbf{x}|]{\frac{\prod_{i=1}^{|\mathbf{x}|} \sum_{j=0}^{|\mathbf{y}|} P(x_i \mid y_j)}{(|\mathbf{y}| + 1)^{|\mathbf{x}|}}} \quad (2)$$

where $P(x_i \mid y_j)$ is the lexical probability of the i^{th} source word in $\mathbf{x}$ given the j^{th} target word in $\mathbf{y}$. For the *target-given-source* direction, source and target sentences swap their roles. We denote these two rankings as $r_{\mathrm{a}}^{\mathrm{M1}}(\mathbf{x}, \mathbf{y})$ and $r_{\mathrm{a}}^{\mathrm{M1}}(\mathbf{y}, \mathbf{x})$.

Additionally, we compute another two rankings based on a *Viterbi* implementation of Equation 2:

$$P_{\mathrm{Mv}}(\mathbf{x}, \mathbf{y}) = \sqrt[|\mathbf{x}|]{\frac{\prod_{i=1}^{|\mathbf{x}|} \max_{j=0}^{|\mathbf{y}|} P(x_i \mid y_j)}{(|\mathbf{y}| + 1)^{|\mathbf{x}|}}} \quad (3)$$

where we replace the summation ($\sum_{j=0}^{|\mathbf{y}|}$) in Equation 2 by a maximization. Again, we calculate both source-given-target and target-given-source directions: $r_{\mathrm{a}}^{\mathrm{Mv}}(\mathbf{x}, \mathbf{y})$ and $r_{\mathrm{a}}^{\mathrm{Mv}}(\mathbf{y}, \mathbf{x})$ respectively.

2.4 Ranking Aggregation

Finally, we combined the different rankings described in previous sections to obtain the final ranking of our submission.

Aggregation of Coverage Rankings

We start combining the eight coverage rankings described in Section 2.2. First, we average the four rankings for $\mathbf{x}$ into a source coverage ranking. Then, we repeat the process for the four destination rankings. Finally, we got the final coverage ranking $r_{\mathrm{C}}(\mathbf{x}, \mathbf{y})$ as the average between the source and destination coverage rankings:

$$r_{\mathrm{c}}(\mathbf{x}, \mathbf{y}) = \frac{\dfrac{\sum_{n=1}^{4} r_{\mathrm{c}}(\mathbf{x}, n)}{4} + \dfrac{\sum_{n=1}^{4} r_{\mathrm{c}}(\mathbf{y}, n)}{4}}{2} \quad (4)$$

where $r_{\mathrm{c}}(\mathbf{x}, n)$ denotes the ranking of sentence $\mathbf{x}$ using n-grams of size n as tokens.

Aggregation of Adequacy Rankings

First, we averaged the two (source-to-destination and destination-to-source) rankings computed with Equation 2. Then, we repeated the process for the two rankings computed with Equation 3. The final adequacy ranking $r_{\mathrm{a}}(\mathbf{x}, \mathbf{y})$ was then obtained as the average of these two rankings:

$$r_{\mathrm{a}}(\mathbf{x}, \mathbf{y}) = \left(\frac{r_{\mathrm{a}}^{\mathrm{M1}}(\mathbf{x}, \mathbf{y}) + r_{\mathrm{a}}^{\mathrm{M1}}(\mathbf{y}, \mathbf{x})}{2} + \frac{r_{\mathrm{a}}^{\mathrm{Mv}}(\mathbf{x}, \mathbf{y}) + r_{\mathrm{a}}^{\mathrm{Mv}}(\mathbf{y}, \mathbf{x})}{2} \right) /2 \quad (5)$$

Final Submission Scores

Once we had computed for each sentence pair (x, y) its coverage ($r_{\mathrm{c}}(\mathbf{x}, \mathbf{y})$) and adequacy ($r_{\mathrm{a}}(\mathbf{x}, \mathbf{y})$) rankings, we averaged these two to obtain the final ranking $r(\mathbf{x}, \mathbf{y})$ of the pair:

$$r(\mathbf{x}, \mathbf{y}) = \frac{r_{\mathrm{c}}(\mathbf{x}, \mathbf{y}) + r_{\mathrm{a}}(\mathbf{x}, \mathbf{y})}{2} \quad (6)$$

For the final submission however, the organizers ask to provide a *score* for each pair. Scores do not have to be meaningful, except that higher scores indicate better quality. To do this, we take the simple solution of computing the score $s(\mathbf{x}, \mathbf{y})$ as the number of sentences in the raw corpus (R)[2] divided by the final ranking of the sentence pair.

[2] $R = 2235512$ for Nepali-English, and $R = 3357018$ for Sinhala-English.

Additionally, in order to break potential ties, and to provide a smoothing score for filtered out sentences (see Section 2.1), we added to the score the average word probability as described in Equation 2. The final scores in our submission were:

$$s(\mathbf{x},\mathbf{y}) = \frac{R}{r(\mathbf{x},\mathbf{y})} + P_{\mathrm{M1}}(\mathbf{x},\mathbf{y}) \tag{7}$$

Note that filtered pairs were considered to have an ”infinite” ranking which results in $\frac{R}{r(\mathbf{x},\mathbf{y})} = 0$; for unfiltered pairs the value of this fraction is assured to be greater than one.

3 Submission

We submitted three different score files to the shared task. All employ the same score function Equation 7 but use different ranking functions:

- PRIMARY: computed using as ranking function the combination of coverage and adequacy rankings in Equation 6.
- SECONDARYCOV: computed using only the aggregated coverage ranking in Equation 4.
- SECONDARYADE: computed using only the aggregated adequacy ranking in Equation 5.

3.1 Coverage Rankings Computation

As described in Section 2.2, we implemented a greedy algorithm to compute coverage ranking. At each step, the algorithm selects the sentences that provide a largest increase of ”value” (Equation 1) to a iteratively increasing training corpus.

The computational cost of this approach is $\mathcal{O}(R^2)$ where R is the number of sentences under consideration. The initial filtering partially alleviates this cost by drastically reducing the amount of sentences to rank. However, it is still a slow process that took about one second per iteration with our Python implementation[3]. To further reduce the computational time of the algorithm, we implemented a batch approach where at each step we selected not a single sentence but a batch of the most ”valuable” ones. After some experiments, we chose to select 1000 sentences at each step as a good compromise; running time was reduced by a factor of 1000 while the ”value” of the selected training corpus was barely affected.

	Ne–En		Si–En	
	1M	5M	1M	5M
PRIMARY	3.4 3.1	3.3 **2.6**	3.7 2.1	**4.1** **1.7**
SECONDARYCOV	2.9 0.5	4.2 2.4	2.6 0.1	4.0 1.2
SECONDARYADE	**3.5** **3.6**	**4.3** 2.4	**3.9** **2.9**	**4.1** 1.4

Table 2: Results of our submissions, in BLEU [%]. SMT figures are in blue while NMT is in red. Best results are in bold.

3.2 Adequacy Rankings Computation

The cornerstone of the adequacy ranking described in Section 2.3 is the probabilistic lexicons in Equations 2 and 3. In our submissions, we used the probabilistic lexicons that can be obtained as a sub-product of the training of full statistical MT models. For this end, we used Moses (Koehn et al., 2007) with its default configuration and the parallel data provided by the organizers as training data.

3.3 Evaluation and Results

Participants in the shared task were asked to submit a file with quality scores, one per line, corresponding to the sentence pairs on the Nepali-English and Sinhala-English corpora. The performance of the submissions is evaluated by subsampling 1 million and 5 million word corpora based on these scores, training statistical (Koehn et al., 2007) and neural [4] MT systems with these corpora, and assessing translation quality on blind tests using BLEU (Papineni et al., 2002).

Table 2 shows the scores of our three submissions for each language pair and condition. Of the three, the one based on coverage rankings (SECONDARYCOV) showed a lower performance consistently being outperformed, particularly in the 1 million condition, by both our PRIMARY and SECONDARYADE submissions.

We were surprised by the ”poor” performance of coverage ranking. Previous works (Haffari et al., 2009; Gascó et al., 2012) showed quite promised results. However, in contrast to our case, all these assume the availability of a sample of the domain to be translated. We hypothesize that the lack of this in-domain data in conjunction with the eclectic domains of the data to be filtered are the causes of the poor results of this approach. Moreover, the greedy selection implemented may aggravate this issue by taking not-optimal initial decisions from which the algorithm cannot recover.

Another interesting observation is the unintu-

[3] After filtering about 176k pairs remained for Nepali-English, and 442k pairs remained for Sinhala-English.

[4] https://github.com/facebookresearch/flores

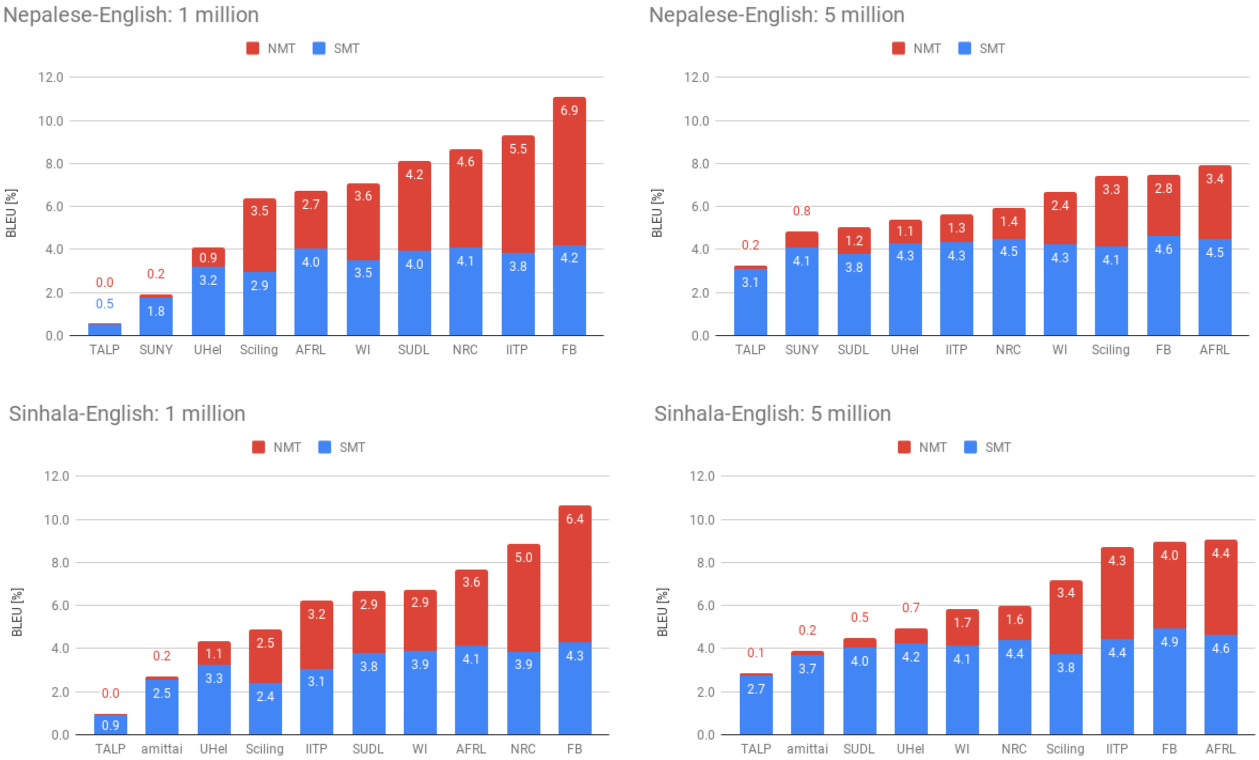

Figure 1: Best results for each team in the competition. We display the participants by increasing sum of BLEU scores for neural and statistical MT models.

itive results for NMT. While SMT results tend to go up as more data is selected, results for NMT tend to show the opposite trend. A fact to consider is that actual BLEU figures are quite low so the actual relevance of these trends are not clear. Additionally, given that this observation is valid other submissions as we will see next, we think this is an issue worthy of further investigation.

After discussing the performance of our submissions, we will compare our best submission on each condition to the rest of participants. Figure 1 summarizes the results of the shared task as reported by the organizers of the task (Bojar et al., 2019). Each sub-figure displays the best submission of each individual participant institution for a particular task and condition. Plots in the upper row show results for Nepalese-English while the bottom row does the same for Sinhala-English. Plots in the left column are for the 1 million condition while results for the 5 million condition are shown in the right column. Stacked bars displayed in the plots denote the BLEU scores for the statistical (blue) and neural (red) systems. We sort them in increasing order according to each system's sum of SMT and NMT scores.

The organizers do not provide confidence intervals for the reported scores so compare results is somehow difficult. Still, as we mention previously, it is surprising the degradation in translation quality for NMT when comparing the 5 million condition to the 1 million condition. Usually, a larger amount of data correlates with an increase in translation quality. In this case, however, scores for SMT barely changed while NMT results went down. This seems to indicate that our methods were not sophisticated enough to find adequate data, or that the really adequate data in the noise corpora amount for less than 5 million words.

Our submission (WI) lays in the upper half among the best submission of the different participants. Regarding Nepalese-English, it scored an aggregated of 7.1 and 6.7 BLEU points for the 1 million and 5 million conditions respectively. This represent respectively about a 64% of the best result submitted for the 1 million condition, and about a 85% of the best result for the 5 million condition. As for the Sinhala–English condition, we scored 6.8 and 5.8 BLEU points which represent a 64% of the best results respectively.

4 Conclusions

We have presented our submission to the WMT19 shared task on parallel corpus filtering. We have mostly explored the application of coverage augmentation ranking techniques with the aim at selecting the subset of sentence pairs that provide the best coherent coverage for the raw sentences.

Results have shown that our proposed coverage approach is not well suited for this particular task. Our secondary submission based on lexical scoring works better in all conditions, and even outperforms our primary submission that combines both coverage and lexical rankings.

One interesting effect seen in the results of the task is the reduced performance on NMT in the presence of more data that can be observed for all participants. Given this, we think that exploring methods able to decide when adding more data will be harmful for performance it is a good research direction to explore.

Acknowledgments

We want to thank the reviewers of the paper for their valuable comments and suggestions. Work funded by WebInterpret Inc.

References

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Christof Monz, Mathias Müller, and Matt Post. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 152–161.

Jesús González-Rubio. 2014. *On the Effective Deployment of Currrent Machine Translation Technology*. Ph.D. thesis, DSIC, U. Politècnica de València. Supervised by Dr. Daniel Ortiz-Martínez and Prof. Francisco Casacuberta.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 422–432, Montreal, Canada. Association for Computational Linguistics.

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*, pages 415–423.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 901–908, Belgium, Brussels. 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, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007, System Demonstrations*.

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.

Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graa, Aman Gokrani, and Hermann Ney. 2018. The rwth aachen university filtering system for the wmt 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 959–967, Belgium, Brussels. Association for Computational Linguistics.

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)*, pages 1715–1725.

George Kingsley Zipf. 1935. *The Psychobiology of Language*. Houghton-Mifflin.