Bicleaner at WMT 2020: Universitat d'Alacant–Prompsit's submission to the parallel corpus filtering shared task

Miquel Esplà-Gomis[⋆] **Víctor M. Sánchez-Cartagena**[⋆]
Jaume Zaragoza-Bernabeu[†] **Felipe Sánchez-Martínez**[⋆]

[⋆]Dep. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant
E-03690 Sant Vicent del Raspeig (Spain)
{mespla,vmsanchez,fsanchez}@dlsi.ua.es

[†]Prompsit Language Engineering
Av. Universitat s/n. Edifici Quorum III, E-03202 Elx (Spain)
jzaragoza@prompsit.com

Abstract

This paper describes the joint submission of Universitat d'Alacant and Prompsit Language Engineering to the WMT 2020 shared task on parallel corpus filtering. Our submission, based on the free/open-source tool Bicleaner, enhances it with Extremely Randomised Trees and lexical similarity features that account for the frequency of the words in the parallel sentences to determine if two sentences are parallel. To train this classifier we used the clean corpora provided for the task and synthetic noisy parallel sentences. In addition we rescore the output of Bicleaner using character-level language models and n-gram saturation.

1 Introduction

This paper describes the joint submission of Universitat d'Alacant and Prompsit Language Engineering to the parallel corpus filtering shared task at the Fifth Conference on Machine Translation (WMT 2020). Our submission is built upon Bicleaner (Sánchez-Cartagena et al., 2018),[1] a widely-used free/open-source tool for detecting noisy parallel sentences that participated in the 2018 edition of this shared task and ranked fourth out of 17 submissions on one of the sub-tasks. We provide quality scores for the sentence pairs provided by the organiser without re-aligning them.

The 2020 edition of the parallel corpus filtering shared task focuses on two under-resourced Asian languages paired with English: Khmer and Pashto. Khmer (km) is the official language of Cambodia and is spoken circa 16 million people in Cambodia, Vietnam and Thailand.[2] There are about 500k English–Khmer parallel sentences in OPUS,[3] mainly belonging to narrow domains like software products and religion. Pashto (ps) is spoken by around 40 million people in Pakistan and in Afghanistan, where it is official together with Persian.[4] There are around 100k English–Pastho parallel sentence in OPUS, most of which belong to the software domain.

Detecting noisy parallel sentences for under-resourced language pairs, like those addressed in this shared task, is challenging. Pastho is not directly supported by LASER (Schwenk and Douze, 2017), although it supports other Iranian languages, and there are few bilingual resources for building the Bicleaner's models.

Bicleaner is based on a classifier that assesses whether a pair of sentences are mutual translations or not. It is trained on a parallel corpus (positive samples) and on an automatically corrupted version of the same corpus (negative samples). The most important features used by the classifier are lexical similarity scores obtained with the help of probabilistic bilingual dictionaries, which are also extracted from the parallel corpus. Our submission improves the performance of the version of Bicleaner that took part in the 2018 shared task in multiple ways: a new classification algorithm, new lexical features that account for the frequency of the words in the parallel sentences, and a novel way of generating corrupted pairs of sentences. In addition, we re-score the output of Bicleaner combining character-level language models and an n-gram saturation scorer in a linear combination whose parameters are determined by fine-tuning the MBART model provided by the organisers of the shared task.

The rest of the paper is organised as follows. Section 2 describes the Bicleaner classifier whereas Section 3 explains how the score produced by the

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

[2] Wikipedia: https://en.wikipedia.org/wiki/Khmer_language

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

[4] Wikipedia: https://en.wikipedia.org/wiki/Pashto

Proceedings of the 5th Conference on Machine Translation (WMT), pages 950–956
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

classifier is combined with the information provided by character-level language models and an n-gram saturation algorithm to produce the submitted score. Section 4 then describes the process followed to build the submission, and Section 5 lists related approaches. The paper ends with some concluding remarks.

2 Bicleaner classifier

Bicleaner is based on an automatic classifier that produces a score for a pair of sentences representing the probability that they are mutual translations. Random Forests (Breiman, 2001), the classification algorithm used in the 2018 submission, has been replaced by Extremely-Randomised Trees (Geurts et al., 2006) because the latter performed best on preliminary experiments.

The Extremely Randomised Trees classification algorithm works by selecting at each internal node the *best* feature from a sub-set of features selected at random from the whole set of features F, and using a random cut-off point. The hyper-parameters controlling the training of these classifiers are therefore the method used to rank the features and select the best one, the size of the subset of features selected at random, and the number of trees to be used. To select the best hyper-parameters we performed a grid search with the following hyper-parameter values. For the ranking we tried with Gini importance (Breiman et al., 1984, Ch. 4) and information gain; for the size of the sub-set of features we tried with $|F|$, $log_2|F|$ and $\sqrt{|F|}$; for the number of trees we tried with 100, 200, 300, 400 and 500.

The features we used can be split in two groups: those that account for the lexical similarity of the two sentences, and those based on shallow properties of the sentences.

2.1 Lexical features

Bilingual lexical similarity is assessed by means of the lexical feature $\mathrm{Qmax}(S, \Theta, d)$, which was first described by Sánchez-Cartagena et al. (2018) and is inspired by the translation probabilities used in statistical machine translation (Koehn, 2009). It is defined as:

$$\mathrm{Qmax}(S,\Theta,d) = \left(\prod_{t\in\Theta} \max_{s\in\mathcal{S}\cup\{\mathrm{NULL}\}} p(t|s;d)\right)^{\frac{1}{|\Theta|}}$$

where, S is a source-language (SL) sentence, $\mathcal{S}$ is a set with the tokens in S, Θ is a set with the tokens in the target-language (TL) sentence T that appear at least once in the SL-to-TL probabilistic bilingual dictionary d, and $p(t|s;d)$ stands for the translation probability of the target token t given the source token s according to the bilingual dictionary d. A smoothing is applied if, for a token t, $\max_{s\in\mathcal{S}\cup\{\mathrm{NULL}\}} p(t|s;d)$ equals zero; in that case, this expression is set to the value of the smallest probability in d divided by 10. One can interpret that, in this case, the dictionary is providing evidence that t is unlikely to be the translation of any of the tokens in S. It is worth noting that this case differs from the case in which a token $t \in T$ does not appear in the dictionary at all; in that case, no evidence, either positive or negative, is available for it. This is why Qmax is only computed for the tokens in Θ instead of doing so for all the tokens in T.

The informativeness of Qmax strongly depends on the coverage of the probabilistic bilingual dictionary used. To measure the coverage of this dictionary, the feature Qmax is complemented with two additional features: $\mathrm{CoverT}(T, d)$, which returns the percentage of unique tokens in T appearing in d, and $\mathrm{CoverTS}(S, T, d)$, which returns the percentage of unique tokens in T that appear in d associated with at least one token in S. All these features are also computed in the reverse direction: $\mathrm{Qmax}(\Theta, S, d')$, $\mathrm{CoverS}(S, d')$, and $\mathrm{CoverST}(T, S, d')$, where d' is a TL-to-SL probabilistic bilingual dictionary.

Even though low-frequency words usually have more discriminatory power (Ramos, 2003), the original formulation of the Bicleaner lexical features did not take into account word frequency in any way. In order to allow the classifier to give different weights to words from different frequency ranks, we re-formulated the lexical features: Qmax now becomes a set of features $\{\mathrm{Qmax}_q(S, \Theta, d, R) \mid q \in [1, 4]\}$. While the summation in the original Qmax was computed for all the tokens in Θ, in Qmax_q it is only computed for those tokens in Θ that appear in the quartile $q \in [1, 4]$ of the ranking of tokens R. R sorts tokens by the logarithm of their relative frequency in a monolingual corpus; in this way, quartile $q = 1$ contains a large amount of tokens with low frequency, while quartile $q = 4$ contains fewer tokens with high frequency.[5] The same adaptation is applied to obtain

[5] Preliminary experiments showed that no gain is obtained by dividing word frequencies in more than four groups.

the set of features $\{\mathrm{CoverS}_q(T, d) \mid q \in [1, 4]\}$ and $\{\mathrm{CoverST}_q(S, T, d) \mid q \in [1, 4]\}$. As in the original Bicleaner, these features were also computed in the reverse direction.

2.2 Shallow features

Shallow features do not make use of bilingual lexical information and are aimed at complementing the lexical features, which may not be reliable enough in sentence pairs with poor dictionary coverage. The shallow features used can be further split into those that model sentence length and those that identify tokens and characters that give hints about the parallelness of a pair of sentences.

Features that model sentence length are based on the assumption that the ratio between the lengths of a pair of parallel sentences is fairly constant for a given language pair. Hence, sentence pairs that deviate too much from this ratio are not likely to be parallel. We measure how close is the ratio of a given pair of sentences to the expected one as the probability mass function of a Poisson distribution. We also provide the raw lengths to the classifier. The complete list of features based on sentence length is the following. Each of these features is computed independently for the SL sentence S and for the TL sentence T of the pair.

- Likelihood of having a TL segment T with length (in tokens) l_T given l_S, the length of the SL segment S, and r_{ts}, the ratio between the length of TL and SL computed on a training parallel corpus; likelihood is computed as $Pr(X = l_T; \lambda = l_S \cdot r_{ts})$. This feature is also computed for S: $Pr(X = l_S; \lambda = l_T \cdot r_{st})$. Note that $Pr(X = k; \lambda = L) = \frac{e^{-L} \cdot L^k}{k!}$.
- Number of tokens in the sentence.
- Number of characters in the sentence.
- Average token length (in characters) in the sentence.

Parallel pairs of sentences are also likely to share numerical expressions, punctuation marks and proper nouns. The following features aim at leveraging that information. Each of these features is computed independently for S and T.

- Number of punctuation marks of each type.
- Proportion of numerical expressions in the sentence that can be found in the other sentence of the pair.
- Proportion of capitalised tokens in the sentence that can be found in the other sentence of the pair.

Finally, character counts can also be considered hints for parallelness. They are taken into account by the following features, which are computed independently for S and T:

- Number of characters in each of the main Unicode classes.
- Number of different characters.
- Number of occurrences of the three most frequent characters, normalised by sentence length.
- Entropy of the string, considering each character as an event whose probability is proportional to the number of occurrences of the character in the sentence.
- Maximum number of consecutive repetitions of the same character.

Overall, 92 shallow features are used.

2.3 Modelling noise

For training the Bicleaner classifier, positive and negative samples are used. The positive samples are those found in the original parallel corpus. The negative samples are generated by corrupting the sentences in that corpus as explained next.

Three types of synthetic noise are applied for corrupting the sentences:

- wrong alignment: parallel segments are randomly re-aligned to produce pairs of segments that are not parallel;
- wrong segmentation: one of the sentences in the pair is truncated: a suffix starting from a random position is removed, therefore emulating an error in sentence segmentation; and
- word replacement: a random number of words in one of the sentences of the pair is replaced by other words with similar frequency as computed on a monolingual corpus.[6]

The amount of corrupted sentences we generated equals the the size of the original parallel corpus,

[6] The ranking of token frequencies R described in Section 2.1 was used for this replacements.

and the three types of synthetic noise were applied in the same proportion. The classifier is therefore trained on a set of sentences twice as large as the original parallel corpus. This strategy differs from the one followed in the 2018 submission (Sánchez-Cartagena et al., 2018) for generating corrupted sentences, where only the "wrong alignment" type of noise was used.

3 Re-scoring

Subsampling 5 million words from the raw corpus based on the score described in the previous section ensures that NMT systems are trained on parallel data. However, some of the selected training parallel samples may not bring useful information and replacing them with other, more informative samples could improve the performance of the resulting NMT systems. We hypothesise that two main reasons could make a pair of sentences which are mutual translations non-informative: i) sentences are not fluent enough and hence very different from those that will be translated with the resulting NMT systems: lists of keywords or website menus are examples of such non-fluent sentences; and ii) the pair of sentences is too similar to other training samples.

To take into account these additional factors, the final score assigned to each sentence pair was computed as follows. First, each sentence received a preliminary score, $\mathrm{prescore}$, computed as:

$$\mathrm{prescore}(s,t) = \lambda \cdot \mathrm{bicleaner}(S,T) + (1-\lambda)\cdot \min(\mathrm{fluency}_s(S), \mathrm{fluency}_t(T)))$$

where S and T are respectively the SL and TL sentence, $\mathrm{bicleaner}$ is the score described in Section 2, and $\mathrm{fluency}_s$ and $\mathrm{fluency}_t$ denote, respectively, fluency scores in the SL and in the TL provided by character language models.[7]

Fluency scores were computed as the normalised perplexity of the sentence according to a 7-gram character language model estimated with KenLM (Heafield, 2011). Normalisation was aimed at placing the perplexities in the $[0, 1]$ interval and consisted on a linear transformation that ensured that the values in the raw corpus had a mean of 0.5 and standard deviation of 0.25. Assuming that the perplexities follow a normal distribution, 95% of the values fall into the desired range. Those values with score lower than 0 or higher than 1 after the transformation were set to 0 and 1, respectively.

After computing $\mathrm{prescore}$, sentence pairs were sorted by that score in descending order, and the score of those pairs for which all their 3-grams could be found in sentences with a higher score was multiplied by a penalty β to promote diversity in the subsampled corpus.

The values of the parameters λ and β, that control the contribution of parallelness, fluency and novelty to the final score were optimised so as to maximise the BLEU score obtained after fine tuning the MBART model provided by the task organisers. The Nelder-Mead algorithm (Nelder and Mead, 1965), which does not require gradient computations, was used.

4 Building the submission

This section describes the process followed to build our submission, which comprised selection of training data, corpora preprocessing, classifier training and evaluation of different alternatives for some of the steps.

4.1 Data used

For both language pairs, the classifier training data was built from the concatenation of all the clean parallel corpora provided by the shared-task organisers. The length ratios used in shallow features were computed on the same data, as well as the bilingual dictionaries. In order to build the dictionaries, the parallel sentences were word-aligned with `MGIZA++`.[8] Alignments were symmetrised with the heuristic *grow-diag-final* and the probabilities in the bilingual dictionaries were estimated afterwards by maximum likelihood.

The Wikipedia monolingual corpus provided by the organisers was used to compute the word frequencies for word ranking R as described in Section 2.1. The same monolingual data was used to train character language models. Pashto and Khmer models were trained on the complete data. A different English language model was trained for each language pair on a random sample of the

[7] Values of λ close to 1.0 make lists of keywords or website menus that are mutual translations to have the highest scores. Value of λ around 0.5 make the top scored segment pairs to be fluent, complete grammatical sentences. Values of λ close to 0.0 make fluent but non-parallel sentences to receive the highest scores.

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

English Wikipedia corpus that matched the size of the Pashto/Khmer Wikipedia corpus.

4.2 Pre-filtering

The clean parallel data provided by the organisers was filtered before their use. Those parallel sentences in which at least one side contain less than 20% of characters in the Unicode range of the corresponding language were discarded. The remaining parallel sentences were deduplicated.

The raw sentence pairs to be scored were also pre-processed with a series of heuristic rules: the score was set to zero if any of the conditions was met. These rules were aimed at detecting segments with evident flaws and speeding up the subsequent steps. The rules were aimed at detecting the following defects in the parallel sentences:

- Wrong language: same Unicode filtering applied to the clean corpora (see above).
- Too long sentences: those with more than 1024 characters.
- Untranslated: SL and TL segments are identical after removing numerical expressions and punctuation marks.
- Not fluent: the sentence contain elements such as URLs, arithmetic operators, too many parentheses, escaped Unicode characters, and other common defects that arise when crawling parallel corpora from the web. These elements were detected by means of regular expressions.

4.3 Tokenisation and word segmentation

Tokenization and subword segmentation have shown to improve the recall of the probabilistic dictionaries used to obtain the lexical features described in Section 2.1. We experimented with the following tokenisation and subword segmentation methods, which were applied to the clean data as well as to the raw sentences to be scored:

- Rule-based tokenisation (`tok`) for Pashto, Khmer and English, as provided by the tool Polyglot (Al-Rfou et al., 2013);
- Rule-based tokenisation plus word morphological segmentation with Morfessor (`tok-morph`). For this we used, after tokenisation, the pre-trained models for Morfessor (Virpioja et al., 2013) included in Polyglot.

4.4 Training Bicleaner

As previously mentioned, the probabilistic bilingual dictionaries were obtained from the same parallel corpus used to train the classifier. This strategy has an important drawback. While almost all words would be found in the bilingual dictionaries when training the classifier, the coverage would be much smaller when classifying the raw sentences because of the small amount of parallel data available. In order to close the gap between training and classification, we removed some dictionary entries during training. Specifically, we removed the least frequent entries so as to ensure that the coverage of the truncated dictionaries on the training data matches the coverage of the full dictionaries on the raw sentences to be scored.

4.5 Results

Table 1 depicts the results obtained on the development environment during the preparation of the submission. The system that produced the scores for our final submission is shown in bold.

We firstly evaluated the different tokenisation alternatives described in Section 4.3, and applied the re-scoring scheme described in Section 3 on top of the best performing one. The results show that the tokenisation with Polyglot without any kind of subword segmentation (`tok`) leads to the best results. It is also worth mentioning the poor performance obtained with morphological segmentation, which needs to be studied more carefully. Moreover, re-scoring for increased fluency and diversity further improved the results.

Table 1 also shows the results obtained by the baseline LASER model,[9] which was consistently outperformed by Bicleaner. Comparing the results of the version of Bicleaner used in this submission with that used in 2018 also shows that the changes introduced bring a positive impact.

5 Related work

A shared task on parallel corpus filtering was part of the WMT conference programme for the first time in 2018 (Koehn et al., 2018). That year the task was targeted at a high-resource scenario. NMT models, which already provide the probability of a

[9] These results do not exactly match those published at `http://www.statmt.org/wmt20/parallel-corpus-filtering.html`, probably because of differences in the GPU hardware or random initialization seed.

	Khmer–English		Pashto–English	
System	**fairseq**	**MBART**	**fairseq**	**MBART**
LASER (baseline)	6.80	10.33	9.55	11.50
Bicleaner 2018 tok	7.45	10.16	10.11	11.85
Bicleaner 2020 tok	7.76	10.66	10.10	12.35
Bicleaner 2020 tok-morph	7.33	10.56	8.64	10.94
Bicleaner 2020 tok + re-score	8.25	11.18	10.53	12.80

Table 1: BLEU scores obtained by the different configurations evaluated for Khmer–English and Pashto–English on the development environment provided by the organisers.

TL sentence given an SL sentence, emerged as the dominant approach (Junczys-Dowmunt, 2018).

Last year's edition was focused on a low-resource scenario (Koehn et al., 2019), where parallel data big enough to build NMT models that provide reliable TL probability distributions was not available. The best performing model was LASER, a method based on multilingual sentence embeddings (Chaudhary et al., 2019) that takes advantage of the data available for multiple language pairs. In fact, a LASER model trained on 93 languages is the baseline model published by the organisers for this edition of the shared task.

Unlike LASER, our submission is mainly based on lexical similarity scores analogous to those used in statistical machine translation. They are computed only on parallel data, without any kind of transfer learning from other language pairs. The approach we follow to detect sentences that are mutual translations is similar to the one by Munteanu and Marcu (2005) for detecting parallel sentences in comparable corpora. However, we use a larger set of shallow features not related to lexical similarity and follow a more sophisticated method for generating negative samples.

Concerning our re-scoring strategy for including information about fluency and diversity, participants from past editions also used these attributes to score sentences. For instance, Axelrod et al. (2019) and Vázquez et al. (2019) devised a scoring strategy under the assumption that parallel sentences should have similar monolingual language model perplexities, and many other submissions included a penalty for repetitive sentences (González-Rubio, 2019; Erdmann and Gwinnup, 2019; Bernier-Colborne and Lo, 2019). Nevertheless, to the best of our knowledge, our approach is the first one that directly optimises the weight of these attributes towards an automatic translation evaluation metric.

6 Concluding remarks

We described the joint submission of Universitat d'Alacant and Prompsit Language Engineering to the parallel corpus filtering shared task at the Fifth Conference on Machine Translation (WMT 2020). Our submission is based on Bicleaner, an open source tool based on a classifier that uses lexical similarity features inspired in the translation probabilities used in statistical machine translation.

We presented a series of improvements over the version of Bicleaner that participated in the 2018 edition of the shared task, namely a better classifier, more sophisticated generation of negative samples and a reformulation of the lexical similarity scores which takes into account word frequency. We showed that these improvements are effective and they allowed our submission to outperform LASER, a state-of-the-art method based on multilingual sentence embeddings. Moreover, combining Bicleaner scores with scores that account for fluency and diversity further improved the results.

We plan to keep exploring subword segmentation algorithms that help to fight data sparseness when computing lexical similarity scores with the help of bilingual dictionaries. We also aim at integrating word embeddings into lexical similarity scores, which would allow us to leverage monolingual data in a more effective way.

Acknowledgments

Work funded by the European Union through projects GoURMET and ParaCrawl. GoURMET — Global Under-Resourced Media Translation, grant agreement number 825299— is funded through the H2020 research and innovation programme. ParaCrawl —actions numbers 2017-EU-IA-0178 and 2018-EU-IA-0063– is funded under the Automated Translation CEF Telecom instrument managed by INEA at the European Commission.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Amittai Axelrod, Anish Kumar, and Steve Sloto. 2019. Dual monolingual cross-entropy delta filtering of noisy parallel data. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 245–251, Florence, Italy. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy. Association for Computational Linguistics.

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

Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. Taylor & Francis.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Grant Erdmann and Jeremy Gwinnup. 2019. Quality and coverage: The AFRL submission to the WMT19 parallel corpus filtering for low-resource conditions task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 267–270, Florence, Italy. Association for Computational Linguistics.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Jesús González-Rubio. 2019. Webinterpret submission to the WMT2019 shared task on parallel corpus filtering. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 271–276, Florence, Italy. 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.

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.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

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

John A. Nelder and Roger Mead. 1965. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313.

Juan Ramos. 2003. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. New Jersey, USA.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Raúl Vázquez, Umut Sulubacak, and Jörg Tiedemann. 2019. The University of Helsinki submission to the WMT19 parallel corpus filtering task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 294–300, Florence, Italy. Association for Computational Linguistics.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report.