Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 824–831
Marseille, 20-25 June 2022

Bicleaner AI: Bicleaner Goes Neural

Jaume Zaragoza-Bernabeu, Marta Bañón, Gema Ramírez-Sánchez, Sergio Ortiz-Rojas
Prompsit Language Engineering, SL (PLE), Spain
{jzaragoza, mbanon, gramirez, sortiz}@prompsit.com

Abstract

This paper describes the experiments carried out during the development of the latest version of Bicleaner, named Bicleaner AI, a tool that aims at detecting noisy sentences in parallel corpora. The tool, which now implements a new neural classifier, uses state-of-the-art techniques based on pre-trained transformer-based language models fine-tuned on a binary classification task. After that, parallel corpus filtering is performed, discarding the sentences that have lower probability of being mutual translations. Our experiments, based on the training of neural machine translation (NMT) with corpora filtered using Bicleaner AI for two different scenarios, show significant improvements in translation quality compared to the previous version of the tool which implemented a classifier based on Extremely Randomized Trees.

Keywords: Machine Translation, Corpus, Tools, Evaluation Methodologies

1. Introduction

Parallel corpus filtering has become an important subtask of machine translation, specially on web-crawled corpora, due to the noisy nature of raw crawls. In the past few years, the WMT Parallel Corpus Filtering Shared Task (Koehn et al., 2018; Koehn et al., 2019; Koehn et al., 2020) has shown the importance of data cleaning for neural machine translation (NMT). Previous studies were warning us about this specific need, in contrast to what happens with statistical machine translation (Khayrallah and Koehn, 2018).

Bicleaner[1] is a tool that aims at detecting noisy sentences in parallel corpora, indicating, for each sentence pair, the likelihood of being mutual translations. In previous submissions (Sánchez-Cartagena et al., 2018; Esplà-Gomis et al., 2020), Bicleaner achieved comparable results to the state-of-the-art. Also, it has been the default cleaning step in the ParaCrawl (Bañón et al., 2020) pipeline since the third release of the corpus. The ParaCrawl pipeline is based on Bitextor, [2] a tool to automatically harvest bitexts from multilingual websites including Bicleaner as a component. Despite discarding a considerable amount of noisy sentences from ParaCrawl and improving machine translation quality, manual observation of the corpus still indicated that further cleaning was needed to bring out the full potential from the data.

In order to take a step forward and catch up with the state of the art, we have developed a new version of the tool that significantly improves its accuracy compared to previous versions. Following the steps of Açarçiçek et al. (2020), who approached the problem similarly to how Bicleaner does (a supervised task where already existing parallel sentences are positive samples, and synthetically generated misaligned sentences are negative samples), we replaced a classifier based on handcrafted features and extremely randomized trees with a fine-tuned XLM-RoBERTa classifier. The new version has been called *Bicleaner AI* and it is publicly available in a separated repository. [3]

To assess the effect of Bicleaner AI as a filtering tool, we carry out an evaluation that follows a similar methodology to the one used at the WMT shared task. To this end, we score each sentence in a noisy web-crawled corpus and extract sub-samples of different sizes that contain sentences with the highest score. We then train neural machine translation systems on each sub-sample and measure their performance on WMT News test sets by computing automatic metrics.

2. Related Work

Machine translation for low-resourced languages was defined as one of the six main challenges of the field in Koehn and Knowles (2017), and, subsequently parallel corpus filtering, has started to focus on them. In this low-resource setting, techniques like transfer learning and multilingual models have become more important for such tasks. Indeed, machine translation for low-resource languages takes benefit from massively multilingual models and can outperform the models that are trained solely on resources in the target languages. Accordingly, in the last Parallel Corpus Filtering task at WMT20[4] one can see the rise of pre-trained transformer language models performing a number of different tasks. Three out of the four top performing submissions used scoring methods based on these. To briefly describe these submissions, the first one by Lu et al. (2020), consisted on training a custom GPT-2 for each translation direction to compute dual cross-entropy scores. The second one, by Lo and Joanis (2020) used YiSi-2 that relies on XLM-RoBERTa word representations to compute cross-lingual lexical simi-

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

[2] `https://github.com/bitextor/bitextor`

[3] `https://github.com/bitextor/bicleaner-ai`

[4] `https://www.statmt.org/wmt20/parallel-corpus-filtering.html`

larity. And, finally, the third one by Açarçiçek et al. (2020) implemented a fine-tuned XLM-RoBERTa as a classifier to filter-out noisy parallel sentences. This last approach, along with ElNokrashy et al. (2020) and Esplà-Gomis et al. (2020), showed the importance of creating more elaborated negative samples instead of simply pairing a sentence in the source language with another random sentence in the target language, as we were doing in previous versions of Bicleaner. Therefore, classifiers are trained with negative samples that mimic a more realistic scenario, where one of the sides is similar to the "correct" translation, but it lacks some words because of segmentation or alignment flaws.

As we mentioned before, our previous version of the tool, Bicleaner, used a classifier based on Extremely Randomised Trees using a wide range of handcrafted features. Features were grouped in two types, lexical features (based on bilingual dictionaries and word frequencies) and shallow features (based on sentence length, character distribution and other heuristics). This approach was used in the submission of the last shared task for corpus filtering, described in Esplà-Gomis et al. (2020) and ended up being among the top four performers in the final results, being the only one among them that was not based on deep-learning techniques. Although this was not a bad result, we wanted to explore if a different type of classifier could improve the quality of the corpus produced by ParaCrawl and, as a consequence, the quality of the neural machine translation models built with them.

3. Methodology

3.1. Classifier training data

To train a Bicleaner AI model, a clean parallel corpus is needed. To create the training corpus, we have downloaded several corpora for each language using MT-Data,[5] selecting only the ones that do not come from websites or are translation memories for software like Ubuntu or PHP. Despite the fact that some non web-crawled corpora available at OPUS are cleaner than web-crawled ones, they still have some issues that need to be addressed in order to mitigate possible flaws in the resulting classifier.

To address these issues, we have applied some fixes:

- Detokenize corpora that are distributed in a tokenized format, like JW300 and TED2020.
- Remove non printing characters and normalize punctuation.
- Apply character and orthography fixing with Bifixer (Ramírez-Sánchez et al., 2020).

We also discarded sentences that matched any of the following criteria:

- Word-based sentence length ratio larger than 2.5 or smaller than 0.4.
- More than 50% of non-alphabetic characters on either side.
- Longer than 200 words or shorter than 2 words on either side.
- Identified by FastText[6] as being in a different language than the expected one.
- Different amount of simplified sentence-ending conditions —a period, question mark or exclamation mark, followed by a word starting by capital letter— on either side.

The last rule is very aggressive but has helped to remove most of the glued sentences in OpenSubtitles, that are often the result of an incorrect alignment or segmentation. It has to be noted that removing too many sentences is not a problem in our scenario as, according to our experience, we will not need more than half a million sentences.

After applying all these fixes and filters, we have randomly sampled subsets from each corpus so that none of them takes more than a third of the total to make sure that we have a balanced representation of each domain.

3.2. Synthetic Noise

We use a classifier to give sentence pairs a score between 0 and 1, the higher the score, the higher the probability of a sentence pair to be an actual translation. Therefore, the training of the classifier follows a supervised learning framework where the positive samples are sentences from already existing parallel corpora, and the negative samples are created by corrupting the same positive samples. With the synthetic noise, we try to emulate common errors introduced by the sentence segmentation and alignment tools used to produce web-crawled corpora.

To corrupt the samples, we use the same types of noise that has been used in Bicleaner `0.14`, described in section 2.3 of Esplà-Gomis et al. (2020):

- Random alignment: parallel segments are randomly re-aligned to produce pairs of segments that are not parallel.
- Word omission: some words are omitted; the fraction and the chosen words are random. This noise replaces wrong segmentation noise in the cited paper to emulate a more generic noise where some parts of the sentence are missing, either because of bad segmentation or other similar issues.
- Frequency-based noise: some words of the target sentence are replaced by words with similar monolingual frequency.

[5] https://github.com/thammegowda/mtdata

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

In addition, we employ a 1:10 positive to negative ratio introduced by Açarçiçek et al. (2020). The training and development sets are built using sentence pairs from a parallel corpus provided by the user as positive samples. Then, for each positive example we apply random alignment three times, word omission three times and frequency noise four times.

3.3. Architecture

Bicleaner AI comes with two types of neural network classifiers: lite models and full models. Lite models provide high-speed inference, while full models are intended for high-performance inference.

3.3.1. Decomposable Attention

Lite models are based on *decomposable attention* (Parikh et al., 2016), an attention-based classifier that has shown good performance with a significant small number of parameters. Compared to the original paper, instead of using pre-trained English embeddings, we added a bilingual SentencePiece joint vocabulary whose embeddings are trained with GloVe (Pennington et al., 2014). This allows us to eliminate the tokenisation dependency and to have a more compact vocabulary that uses less memory.

The training is performed using a batch size of 1024 sentences during 200 epochs with a learning rate of 5×10^{-3} and an inverse time decay schedule. The embedding size is 300 dimensions and the maximum length of each side is 100 tokens.

3.3.2. XLM-RoBERTa

Full models are based on XLM-RoBERTa (Conneau et al., 2020) pre-trained language model, subsequently fine-tuned by adding a hidden layer of 2,048 ReLU units and trained with a 10% dropout rate. Fine-tuning is performed with a learning rate of 2×10^{-6}, a batch size of 128 sentences, 1,000 warm-up steps, followed by checkpoints every 2,000 steps and stopping after three checkpoints without improvement on a development set, or a maximum of 30,000 steps.

4. Experiments

4.1. Classifier Evaluation

In order to perform validation checkpoints during training and have an idea of the performance of the model, we use the development set to compute the Matthews correlation coefficient ϕ between positive and negative class. This score is very similar to the F-score but more informative, and generally recommended as in Chicco and Jurman (2020). Its values range between -1 and $+1$, where +1 means perfect prediction, 0 means random prediction and -1 means inverse prediction.

As it can be seen in Table 1, full models clearly outperform lite models.[7] Furthermore, full models achieve quite good performance for very low-resourced languages (like Irish or Icelandic), and on languages that were not seen during pre-training (like Maltese)[8]. We have also trained a multilingual model (en-xx) that achieves worse performance than the individual full models, but still very competitive compared to the average performance of lite models. This multilingual model also has the advantage of being able to deal with languages that have not been seen during fine-tuning. To train it, we concatenate all of the training sets generated for the other languages. The multilingual development set is made of a concatenation of random samples from each bilingual development set.

Language pair	ϕ_{full}	ϕ_{lite}
en-bg	0.879	0.588
en-cs	0.864	0.555
en-da	0.879	0.665
en-de	0.898	0.675
en-el	0.837	0.581
en-es	0.872	0.651
en-et	0.836	0.585
en-fi	0.875	0.614
en-fr	0.880	0.681
en-ga	0.685	0.586
en-hr	0.881	0.646
en-hu	0.854	0.555
en-is	0.787	0.681
en-it	0.862	0.628
en-lt	0.856	0.477
en-lv	0.839	0.581
en-mt	0.850	0.716
en-nb	0.859	0.703
en-nl	0.849	0.659
en-nn	0.806	0.695
en-pl	0.890	0.636
en-pt	0.863	0.683
en-ro	0.889	0.609
en-sk	0.858	0.712
en-sl	0.858	0.609
en-sv	0.869	0.694
en-xx	0.710	-
es-ca	0.915	0.810
es-eu	0.657	0.615
es-gl	0.845	0.644

Table 1: Matthews correlation coefficient of full models (ϕ_{full}) and lite models (ϕ_{lite}) for each language pair covered in ParaCrawl project.

4.2. NMT Evaluation

In order to evaluate the impact of cleaning, we train neural machine translation systems with the resulting corpora. We do this in two ways. Firstly, by extracting sub-samples of different sizes with the best scores and

[7] Full and lite models are publicly available at https://github.com/bitextor/bicleaner-ai-data/releases/tag/v1.0

[8] According to XLMR paper (Conneau et al., 2020), Maltese was not present in the training data

training NMT systems on each sub-sample. Secondly, by filtering out all the sentence pairs with a score under the 0.5 threshold and combining with other existing corpora to train the NMT systems.

4.2.1. Sub-sample Evaluation

This evaluation is performed in the same way as in the WMT Parallel Corpus Filtering shared task. Firstly, we score the noisy corpus with Bicleaner AI; secondly, we sort all the sentences by score; thirdly, we extract sub-samples[9] of different sizes containing the top scoring sentences; and finally, we train a NMT system with each sub-sample using development and test sets from the WMT News Translation Task. [10].

The main objective of this method is to be able to study the performance without the need of choosing a score threshold, and with different sizes emulating different amounts of resources.

The experiments carried on with this method are performed in three language combinations: English→ Finnish, English→Latvian and English→Romanian. The noisy corpus to be scored is the raw ParaCrawl v7 after filtering those sentences with a Bicleaner score of 0 and removing duplicates and near duplicates (See Table 4 for sizes after deduplication and filtering). The sizes of the sub-samples are measured in number of tokens chosen in relation to the total size of the raw corpus. These range from 5M tokens to 100M tokens depending on the language combination.

At the end, the NMT systems are trained to translate from English into the other three languages and evaluated computing BLEU on the WMT News test set.

Embedding similarity scoring In order to compare to other state-of-the-art methods we use LASER (Artetxe and Schwenk, 2019b) embeddings to compute sentence similarity. We use as score the ratio between the cosine of the candidate and the average of its k nearest neighbours, as proposed by Artetxe and Schwenk (2019a).

Bicleaner scoring The Bicleaner versions to be compared in this evaluation scenario are *Bicleaner 0.14* [11] and *Bicleaner AI* with full models. Since both flavours of Bicleaner perform scoring of sentences in an independent way, and LASER performs the scoring of each sentence taking into account all of the other sentences, sub-samples containing the top scores for Bicleaner can contain very repetitive sentences. Therefore, previously to the sub-sampling, we applied an n-gram saturation re-scoring. To perform the re-scoring, we sort the sentence pairs by score in descending order and apply a β penalty of 0.8 to sentence pairs whose all word 2-grams are present in sentences with higher score. This re-scoring method was applied similarly to our last submission (Esplà-Gomis et al., 2020).

4.2.2. ParaCrawl Evaluation

ParaCrawl corpora have been evaluated through machine translation by carrying out an experiment that has been reproduced for all versions of the corpora. The experiment comprises the training of baseline NMT models using the corpora available for the WMT news translation task for 5 language combinations.[14] After that, we add the clean ParaCrawl corpus (all the sentence pairs over a set Bicleaner score) to the baseline corpus and we train again NMT models with the concatenation of both. We, then, compare the performance of the baseline and the baseline plus ParaCrawl models using automatic metrics like BLEU.

Bicleaner is an important component in the ParaCrawl production pipeline, probably the one having the biggest impact on the corpora final quantity and quality. However, other components in the pipeline can also influence the final corpora and these were also continuously improved during the project. Despite having this into account, as Bicleaner AI was introduced between versions 8 and 9 of the ParaCrawl corpora, we find interesting reporting the details of this type of evaluation and the results focusing on versions 8 and 9 to observe the possible positive impact of Bicleaner AI already observed in the sub-sample evaluation method. Better alignment and additional data may also have contributed to the reported results.

NMT Training Details All NMT systems trained are transformers with a 32,000-piece SentencePiece [15] joint vocabulary trained with the Marian NMT toolkit[16] in a Nvidia 3080Ti using parameters specified at Listing 1. BLEU scores are computed with SacreBLEU.[17]

4.3. Results and Analysis

4.3.1. Subsample Evaluation

Table 2 shows the results of the sub-sample evaluation. As we can observe, Bicleaner AI lite models has given better or equal results compared to Bicleaner. Bicleaner AI full models clearly outperform lite models,

[9] The sorting and extraction has been done with the subselect.perl script from the WMT (https://www.statmt.org/wmt20/parallel-corpus-filtering.html).

[10] WMT17 for Finnish and Latvian, and WMT16 for Romanian, downloaded with SacreBLEU.

[11] Last software version at the moment of writing the paper available at: https://github.com/bitextor/bicleaner/releases/tag/bicleaner-v0.14

[13] Those sentences discarded by Hardrules (https://github.com/bitextor/bicleaner-hardrules/), which is the same configuration for Bicleaner and Bicleaner AI.

[14] WMT16 for Romanian and WMT17 for Czech, German, Finnish and Latvian.

[15] https://github.com/google/sentencepiece

[16] https://marian-nmt.github.io/

[17] SacreBLEU signatures for BLEU and chrF2 scores: nrefs:1|bs:1000|seed:12345|case:mixed |eff:no|tok:13a|smooth:exp|version:2.0.0 nrefs:1|bs:1000|seed:12345|case:mixed |eff:yes|nc:6|nw:0|space:no|version:2.0.0

	en→fi			**en→lv**			**en→ro**		
scoring	**5M**	**50M**	**100M**	**5M**	**30M**	**60M**	**5M**	**50M**	**100M**
LASER	**16.5**	21.5	23.2	12.5	17.5	18.9	24.6	30.3	30.4
Bicleaner	14.3	21.0	22.7	12.2	17.1	18.5	21.8	28.7	29.5
Bicleaner AI lite	14.2	22.4	23.7	12.5	17.7	18.9	21.4	28.8	29.6
Bicleaner AI full	13.7	**25.6**	**26.2**	**15.6**	**19.5**	**20.0**	**25.3**	**31.0**	**30.8**

Table 2: BLEU scores on WMT news tests of different sub-samples of ParaCrawl v7.0, sizes are in million tokens. Best scores are in bold.

size	**lang**	**scoring**	**BLEU (μ ± 95% CI)**	**chrF2 (μ ± 95% CI)**
5M	en→fi	LASER	**16.5 (16.5 ± 0.6)**	**51.4 (51.4 ± 0.4)**
	en→fi	Bicleaner AI full	13.7 (13.7 ± 0.5) (p = 0.0010)*	47.5 (47.5 ± 0.4) (p = 0.0010)*
50M	en→fi	LASER	21.5 (21.5 ± 0.7)	56.8 (56.8 ± 0.5)
	en→fi	Bicleaner AI full	**25.6 (25.6 ± 0.7) (p = 0.0010)***	**59.9 (59.9 ± 0.5) (p = 0.0010)***
100M	en→fi	LASER	23.2 (23.2 ± 0.7)	58.0 (58.0 ± 0.5)
	en→fi	Bicleaner AI full	**26.2 (26.2 ± 0.7) (p = 0.0010)***	**60.2 (60.2 ± 0.5) (p = 0.0010)***
5M	en→lv	LASER	12.5 (12.5 ± 0.6)	45.5 (45.5 ± 0.6)
	en→lv	Bicleaner AI full	**15.6 (15.6 ± 0.7) (p = 0.0010)***	**48.0 (48.0 ± 0.6) (p = 0.0010)***
30M	en→lv	LASER	17.5 (17.5 ± 0.7)	50.1 (50.1 ± 0.6)
	en→lv	Bicleaner AI full	**19.5 (19.5 ± 0.8) (p = 0.0010)***	**51.6 (51.6 ± 0.6) (p = 0.0010)***
60M	en→lv	LASER	18.9 (19.0 ± 0.8)	51.3 (51.3 ± 0.7)
	en→lv	Bicleaner AI full	**20.0 (20.0 ± 0.8) (p = 0.0010)***	**51.9 (51.9 ± 0.6) (p = 0.0010)***
5M	en→ro	LASER	24.6 (24.6 ± 0.7)	53.3 (53.3 ± 0.6)
	en→ro	Bicleaner AI full	**25.3 (25.3 ± 0.8) (p = 0.0030)***	**53.9 (53.9 ± 0.6) (p = 0.0010)***
50M	en→ro	LASER	30.3 (30.3 ± 0.9)	58.1 (58.1 ± 0.6)
	en→ro	Bicleaner AI full	**31.0 (31.0 ± 0.9) (p = 0.0040)***	58.3 (58.3 ± 0.6) (p = 0.0779)
100M	en→ro	LASER	30.4 (30.4 ± 0.9)	57.9 (57.9 ± 0.6)
	en→ro	Bicleaner AI full	30.8 (30.8 ± 0.9) (p = 0.0579)	**58.5 (58.5 ± 0.6) (p = 0.0010)***

Table 3: Statistical significance tests for Bicleaner AI full paired with the best scoring baseline (LASER) from Table 2. For each systems pair, the best score is highlighted in bold when the difference is statistically significant (p-value is higher than 0.05).

corpus	**fi**	**lv**	**ro**
raw	1,373	397	1,123
near-dedup filtered 0	15	8	14

Table 4: Paracrawl v7.0 sizes used for the sub-sample evaluation. Corpus sizes are in million sentences. The second corpus comes from the raw corpus after removing duplicates, near-duplicates and sentences with Bicleaner score equal to 0.[13]

training corpus	**cs-en**	**en-cs**	**de-en**	**en-de**	**fi-en**	**en-fi**	**lv-en**	**en-lv**	**ro-en**	**en-ro**
WMT	28.1	21.7	33.4	27.2	24.8	21.3	18.1	15.2	33.4	28.3
WMT + PC v8	28.8	22.1	35.4	29.7	32.2	25.8	22.9	20.4	40.2	32.6
WMT + PC v9	**28.9**	***22.8**	***36.0**	***30.6**	***33.0**	***27.8**	***24.0**	***20.8**	**40.5**	***33.5**

Table 5: BLEU scores for the NMT models trained with WMT16/17 training corpora and adding Paracrawl v8 and v9. Best scores are in bold, indicated with an asterisk if paired bootstrap resampling test with the second best score, shows statistical significance.

Bicleaner and LASER, except for the smallest sized setting in English→Finnish. Furthermore, we can see at Table 3 that full models have better BLEU and chrF2 scores than LASER (the best scoring baseline) and being statistically significant in 6 out of 9 paired significance tests. In the case of English→Romanian for 50M and 100M tokens, both scores are better and at least for one of those is statistically significant.

4.3.2. Paracrawl Evaluation

The results from the Paracrawl evaluation are shown in Table 5. As we can see, the addition of Paracrawl corpora improved the BLEU scores for all the languages, specially for under-resourced languages like Romanian, Finnish and Latvian.

corpus	cs	de	fi	lv	ro
WMT	52.0	5.8	2.6	4.5	0.6
ParaCrawl v8 raw	3,305.6	26,655.6	1,570.2	490.2	1570.8
ParaCrawl v8 filtered	50.0	261.0	15.0	8.0	13.0
ParaCrawl v9 raw	2,996.9	9,662.1	1,792.1	621.1	1496.2
ParaCrawl v9 filtered	50.6	278.0	31.0	13.0	25.0

Table 6: Corpus sizes in million sentences from the WMT (baseline) and the ParaCrawl corpus versions 8 and 9. Filtered versions are the ones used in the evaluation. Raw versions are shown for comparison with the filtered ones and are all the sentences resulting from the alignment step, including duplicates.

model	1xCPU	1xGPU
Bicleaner	650	-
Bicleaner AI lite	600	10,000
Bicleaner AI full	2	200

Table 7: Speed comparison between Bicleaner versions and models using CPU or GPU and reported as the approximate number of sentence pairs per second.

The numbers also show significant improvements between Paracrawl v9 and v8 (being the introduction of Bicleaner AI one of the main changes). Concretely, we see almost 5 BLEU points on average of improvement between the baseline and Paracrawl v9, and almost 1 point if we compare v8 and v9. With the exception of Czech→English and Romanian→English, for which we got little improvements in BLEU scores with no statistical significance on the test. In the case of Czech, this could be due to the large size of training data included in the baseline (see Table 6) and the nature of the data, made of web-crawled content, being redundant with ParaCrawl data. This could explain the tight difference with the baseline regarding automatic metrics for all versions of ParaCrawl. In the case of Romanian, we believe that something similar is happening between ParaCrawl v8 and ParaCrawl v9. Despite doubling the size of the corpus, improvements are not significant, probably because newly added data is redundant. The WMT training data for Romanian is made of web-crawled news (same domain as the test set), so that could also explain the difficulty of improving the final results.

5. Conclusions

In this paper we have presented the details of Bicleaner AI, a new version of Bicleaner in which the binary classifier component is based on deep learning techniques. This new classifier has been trained for 33 language pairs in full and lite forms to maximise either quality or speed (See Table 7 for speed comparison) depending on the needs. It also provides a multilingual model. Experiments carried out mimicking the ones in the WMT Corpus Filtering Task or inside the ParaCrawl project NMT-based evaluation show that the shift from a machine learning-based classifier to a deep learning-based classifier bring improvements to the quality of the cleaned corpora resulting in improvement in the NMT systems trained with them.

Acknowledgments

We thank all those who worked with us on the ParaCrawl project, particularly Miquel Esplà-Gomis and Mikel L. Forcada, from the University of Alicante, for their useful contributions and comments. The work for this paper was and is still being funded by the Connecting Europe Facility of the Innovation and Networks Executive Agency of the European Commission through the ParaCrawl 3 project (Agreement number INEA/CEF/ICT/A2018/1762029), EuroPat project (Agreement number INEA/CEF/ICT/A2018/1761979) and MaCoCu project (Agreement number INEA/CEF/ICT/A2020/2278341).

The contents of this publication are the sole responsibility of the implementing partners and do not necessarily reflect the opinion of the European Union.

6. Bibliographical References

Artetxe, M. and Schwenk, H. (2019a). Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy, July. Association for Computational Linguistics.

Artetxe, M. and Schwenk, H. (2019b). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Açarçiçek, H., Çolakoğlu, T., aktan hatipoğlu, p. e., Huang, C. H., and Peng, W. (2020). Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation*, pages 940–946, Online, November. Association for Computational Linguistics.

Bañón, M., Chen, P., Haddow, B., Heafield, K., Hoang, H., Esplà-Gomis, M., Forcada, M. L., Kamran, A., Kirefu, F., Koehn, P., Ortiz Rojas, S., Pla Sempere, L., Ramírez-Sánchez, G., Sarrías, E., Strelec, M., Thompson, B., Waites, W., Wiggins, D., and Zaragoza, J. (2020). ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online, July. Association for Computational Linguistics.

Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, 01.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July. Association for Computational Linguistics.

ElNokrashy, M., Hendy, A., Abdelghaffar, M., Afify, M., Tawfik, A., and Hassan Awadalla, H. (2020). Score combination for improved parallel corpus filtering for low resource conditions. In *Proceedings of the Fifth Conference on Machine Translation*, pages 947–951, Online, November. Association for Computational Linguistics.

Esplà-Gomis, M., Sánchez-Cartagena, V. M., Zaragoza-Bernabeu, J., and Sánchez-Martínez, F. (2020). Bicleaner at wmt 2020: Universitat d'alacant-prompsit's submission to the parallel corpus filtering shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 952–958, Online, November. Association for Computational Linguistics.

Khayrallah, H. and Koehn, P. (2018). On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia, July. Association for Computational Linguistics.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August. Association for Computational Linguistics.

Koehn, P., Khayrallah, H., Heafield, K., and Forcada, M. L. (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*, pages 739–752, Belgium, Brussels, October. Association for Computational Linguistics.

Koehn, P., Guzmán, F., Chaudhary, V., and Pino, J. (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 56–74, Florence, Italy, August. Association for Computational Linguistics.

Koehn, P., Chaudhary, V., El-Kishky, A., Goyal, N., Chen, P.-J., and Guzmán, F. (2020). Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online, November. Association for Computational Linguistics.

Lo, C.-k. and Joanis, E. (2020). Improving parallel data identification using iteratively refined sentence alignments and bilingual mappings of pre-trained language models. In *Proceedings of the Fifth Conference on Machine Translation*, pages 972–978, Online, November. Association for Computational Linguistics.

Lu, J., Ge, X., Shi, Y., and Zhang, Y. (2020). Alibaba submission to the wmt20 parallel corpus filtering task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 979–984, Online, November. Association for Computational Linguistics.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *CoRR*, abs/1606.01933.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Ramírez-Sánchez, G., Zaragoza-Bernabeu, J., Bañón, M., and Ortiz-Rojas, S. (2020). Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal, November. European Association for Machine Translation.

Sánchez-Cartagena, V. M., Bañón, M., Ortiz Rojas, S., and Ramírez, G. (2018). Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 968–975, Belgium, Brussels, October. Association for Computational Linguistics.

Listing 1: MarianNMT YAML configuration parameters.

```
seed: 0
mini-batch-fit: True
workspace: 8000
shuffle-in-ram: true
early-stopping: 5
exponential-smoothing: 0.0001
keep-best: True
valid-freq: 10000
valid-mini-batch: 32
save-freq: 10000
overwrite: True
disp-freq: 1000
valid-metrics:
    - ce-mean-words
    - perplexity
    - bleu-detok
beam-size: 6
normalize: 1
cost-type: ce-mean-words
type: transformer
enc-depth: 6
dec-depth: 6
```

```
dim-emb: 512
transformer-heads: 8
transformer-dim-ffn: 2048
transformer-ffn-depth: 2
transformer-ffn-activation: swish
transformer-decoder-autoreg: self-attention
transformer-dropout: 0.1
label-smoothing: 0.1
layer-normalization: True
tied-embeddings-all: True
learn-rate: 0.0003
lr-warmup: 16000
lr-decay-inv-sqrt: 16000
lr-report: True
optimizer-params:
    - 0.9
    - 0.98
    - 1e-09
clip-norm: 0
sync-sgd: true
optimizer-delay: 4
```