Filtering and rescoring the CCMatrix corpus for Neural Machine Translation training

Antoni Oliver, Sergi Álvarez
Universitat Oberta de Catalunya (UOC)
{aoliverg,salvarezvid}@uoc.edu

Abstract

There are several parallel corpora available for many language pairs, such as CCMatrix, built from mass downloads of web content and automatic detection of segments in one language and the translation equivalent in another. These techniques can produce large parallel corpora, but of questionable quality. In many cases, the segments are not in the required languages, or if they are, they are not translation equivalents. In this article, we present an algorithm for filtering out the segments in languages other than the required ones and re-scoring the segments using SBERT. A use case on the Spanish–Asturian and Spanish–Catalan CCMatrix corpus is presented.

1 Introduction

1.1 Parallel corpora crawled from the web

There are several web-derived very large parallel corpora available for a high number of language pairs. Paracrawl[1] (Bañón et al., 2020) is a parallel corpus created crawling the web searching for multilingual pages. At the moment it offers parallel corpora from English to 38 languages and 6 additional language pairs not including English. Wikimatrix[2] (Schwenk et al., 2021a) is created using Wikipedia to automatically find translated sentences. It includes 96 languages, totalling 16,720 language pairs. CCAligned[3] (El-Kishky et al., 2020) is a corpus formed by parallel or comparable web-document pairs in 137 languages aligned with English. From this document corpus, parallel segments are extracted using similarity scores of LASER[4] (Artetxe and Schwenk, 2019) embeddings from the document pairs. OSCAR[5] (Abadji et al., 2022) is also a parallel corpus crawled from the web covering 166 languages. The CCMatrix[6] (Schwenk et al., 2021b) corpus has the particularity that no document information has been used. Instead, all the segments in a given language are compared with all the segments in another language in order to detect parallel segments. To do so, they also use LASER and calculate a *margin score*, defined as the ratio between the cosine distance between the two sentence embeddings, and the average cosine similarity of its nearest neighbours in both directions. This results in very large parallel corpora for 90 languages, totalling 1,197 language pairs.

Some of these corpora, and CCMatrix in particular, suffer from low quality, especially for language pairs with fewer resources. Two main problems are easily detected by a simple visual inspection: segments are not in the correct language, and source and target segments are not translation equivalents. In this paper we present a program that verifies the languages and assesses the translation equivalence of the source and target segments. We evaluate the performance of the program on the CCMatrix corpus for Spanish–Asturian and Spanish–Catalan.

1.2 Automatic language detection

Several language detection libraries implemented in Python are available. Among them, we can

[1] https://paracrawl.eu/
[2] https://github.com/facebookresearch/LASER/tree/main/tasks/WikiMatrix
[3] https://www.statmt.org/cc-aligned/
[4] https://github.com/facebookresearch/LASER
[5] https://oscar-project.org/
[6] https://github.com/facebookresearch/LASER/tree/main/tasks/CCMatrix

highlight the following:[7] (1) langdetect[8] able to detect 55 languages; (2) Spacy-langdetect[9] that in fact uses langdetec, being able to detect by default the same number of languages; (3) fastText,[10] a tool for text classification developed by the Facebook AI Research (FAIR) lab that includes a language identification model able to detect 176 languages; and (4) gcld3,[11] a neural network model for language identification developed by Google that can detect 107 languages.

We have selected fastText language identification module because it is the one detecting more languages and it provides a confidence score for the detected languages. Furthermore, fastText allows training your own models very easily.

1.3 Multilingual models for sentence embeddings

Two libraries for the calculation and use of multilingual sentence embeddings, that also provide ready-to-use models for a lot of languages, can be highlighted. The LASER[12] (Language-Agnostic SEntence Representations) (Schwenk and Douze, 2017) provides models for over 200 languages. This library is the one used to create the CCMatrix corpus. Sentence-Transformers (Reimers and Gurevych, 2019) (SBERT)[13] is a library for sentence, text and image embeddings, offering support for more than 100 languages. Both libraries offer a lot of code examples for different tasks, and they can be used indistinctly.

2 Previous works

The idea of using multilingual sentence embeddings for parallel corpus cleaning is not new. In Chaaudary et al. (2019), LASER is used to create representations of the segments and to score them and filter the noisy parallel segments. They used this technique in a low–resource scenario, but the authors state that it is promising even in no–resource scenarios. In Zhang et al. (2020), the degree of parallelism of the segments is measured using BERT and a domain filter is used to avoid the adverse effect of the domain of the training data. A recent study (de Gibert Bonet et al., 2022) designs a filtering strategy based on a trained classifier. To train the classifier, they use a labelled dataset of parallel segments annotated as valid or invalid. They apply the filtering algorithm to English–Catalan and Catalan–English and achieve improvements between 1.3 and 2.9 BLEU points when training NMT systems on the clean corpus. The resources and algorithms are freely available, but their use is not simple and straightforward.

Few of these works end in a ready-to-use algorithm. Among these, we can mention the following. Zipporah[14] (Xu and Koehn, 2017) uses a bag-of-words translation feature, and needs to train a logistic regression models to filter the parallel corpus. The user has to train the system providing a bad corpus (containing noisy data that should be filtered), a good or training corpus and development data, that should be a clean corpus. Bifixier[15] (Ramírez-Sánchez et al., 2020) performs a restorative cleaning consisting on the following steps: removing of the parallel segments having an empty segment in any of the parts; character fixing; orthography fixing; respliting of the segments and duplicate identification. Bicleaner[16] (Zaragoza-Bernabeu et al., 2022) is a parallel sentence noise filter and classifier tool. The process is done in three steps: (1) pre-filtering based on a set of rules; (2) language model fluency scoring, a language-dependent step using character-based language models; and (3) classification based on a random-forest machine learning model.

3 Description of the resorting and filtering tool

The tool is implemented in two Python programs that can be freely downloaded from GitHub[17]: the rescorer and the selector.

The rescorer algorithm performs two actions:

- It detects the language of the source and target segments using fastText. By default, it uses the lid.176.bin model, that is able to detect 176 languages, but the user can select any other model and even train and use his/her own models.
- It represents the source and target languages using a multilingual sentence embedding

[7] https://towardsdatascience.com/4-nlp-libraries-for-automatic-language-identification-of-text-data-in-python-cbc6bf664774
[8] https://github.com/Mimino666/langdetect
[9] https://pypi.org/project/spacy-langdetect/
[10] https://fasttext.cc/
[11] https://pypi.org/project/gcld3/
[12] https://github.com/facebookresearch/LASER
[13] https://www.sbert.net/
[14] https://github.com/hainan-xv/zipporah
[15] https://github.com/bitextor/bifixer
[16] https://github.com/bitextor/bicleaner
[17] https://github.com/aoliverg/MTUOC-PCorpus-rescorer

model. The implementation uses Sentence-Transformers.[18] By default the LaBSE model is used, that supports 109 languages, but any other model can be used.

These actions are implemented in `MTUOC-PCorpus-rescorer.py`, that uses the following parameters:

- The input corpus. It should be a parallel corpus in TSV format with the source segment, the target segment and, optionally, a score. For example, CCMatrix corpora provides a margin score, that can be used as a third field in the TSV file.
- A path and name for the Sqlite database that will be created. See the description of this database below in this section.
- The source language code.
- The target language code.
- Optionally, a SentenceTransformer model can be provided. By default, the LaBSE model is used.
- Optionally, a fastText language detection model can be provided. By default, the lid.176.bin model is used.

The algorithm creates a Sqlite database with the following structure:

- segmment identifier.
- source segment.
- target segment.
- the score provided by the corpus, if any.
- the detected source language.
- the confidence for the detection of the source language.
- the detected target language.
- the confidence for the detection of the target language.
- the score calculated with the SentenceTranformer, the cosine similarity between the source and the target segments.

[18] https://www.sbert.net/

While reading the input corpus, the Sqlite database is filled with the required information. As the calculation of the SentenceTransformer and the cosine similarity are slow, they are only calculated for those source and target segments with the expected detected languages. Please note that along with the detected language, the confidence scores are stored in the database.

Once the Sqlite database is created, a selection program is used (`MTUOC-PCorpus-selector.py`) to select the parallel segments satisfying a minimum source and target language detection confidence and a minimum SBERT score (the cosine similarity).

4 Experimental part

4.1 Corpora

In the experiments we worked with the CCMatrix for two language pairs involving three Romance languages of the project TAN-IBE: Spanish–Catalan and Spanish–Asturian. This setting is interesting because it involves similar languages (causing difficulties for the automatic language detection) and includes one low resource language: Asturian. In table 1 we can observe the size of these corpora.

Languages	Segments
spa–ast	6,438,281
spa–cat	65,369,659

Table 1: Sizes of the CCMatrix corpus for Spanish–Asturian and Spanish–Catalan.

To automatically evaluate the algorithm we used the Flores-200 corpus (Goyal et al., 2022) for the following languages: Spanish, Portuguese, Catalan, Galician, Occitan and Asturian. For Asturian, a complete revision by a native speaker has been performed in the TAN-IBE project. This corpus has a total of 2,009 segments. Two evaluation corpora have been created from these Flores corpora:

- A monolingual corpus containing all these Flores corpora concatenated and shuffled. This corpus has been used to evaluate the language detection algorithm,
- A parallel corpus with mixed language pairs and directions of these Flores corpora, including: Spanish–Asturian, Asturian–Spanish,

Spanish–Portuguese, Spanish–Catalan and Spanish–Occitan. It also included incorrectly aligned Spanish–Asturian and Asturian–Spanish segments. This corpus has been used to evaluate the capability of the algorithm to select the correct parallel segments.

4.2 Evaluation of the language detection algorith

The evaluation has been performed using the language detection model provided by fastText: lid.176.bin, capable of detecting 176 languages. The detection algorithm can provide a confidence score. In table 2 we can observe the values of precision, recall and L_1 for Asturian, Catalan and Spanish for different values of confidence (the same minimum confidence assigned to both languages).

As we can observe, for any value of confidence we get a 100% precision for Asturian, but very low recall and therefore F_1. This may mean that most of the Asturian segments are detected as other languages, and only very few of the segments are detected as written in this language. This is probably due to the fact that Asturian is underrepresented in the corpus used to train the language detection module. For Catalan, the best F_1 is reached for a confidence of 0.7 and for Spanish for a confidence of 0.9.

The evaluation results for language detection using the existing lid.176.bin model were no satisfactory for Asturian. Using this model will result in rejecting a lot of Asturian segments due to the incorrect language detection. For this reason we decided to train a new language detection model including the languages of the project plus French and English and using the same number of segments for training for all languages. We have included English because a lot of content collected from the web contains segments in English, and we want this content to be detected and filtered out. The inclusion of French is motivated by its similarity to Occitan, and to the fact that a lot of web content in Occitan contains information in French. To do so, we extracted the text from the Wikipedia dumps for Spanish, Portuguese, Galician, Catalan, Asturian, Aragonese, Occitan, English and French. We randomly selected 1,000,000 segments larger than 50 characters from each Wikipedia texts and labeled them with the language code. For the Aragonese Wikipedia we could only select 273,458 segments and for the Occitan Wikipedia 664,728. With this corpus we trained a fastText model using character n-grams of length 2. 3 and 4. In table 3 we can observe the results of the evaluation of the language detection task using the newly trained model. As we can see, the precision for Asturian is kept in very high values with no lack of recall, resulting in very good values of F_1 for all the levels of confidence. The values for Catalan and Spanish are also very good.

4.3 Evaluation of the rescoring algorithm

In this section the results of the evaluation of the rescoring algorithm are showed. We used the parallel corpus with mixed language pairs and directions from the Flores corpora. The task consists on detecting the correct segment pairs for two directions: Spanish–Asturian and Spanish–Catalan. In table 4, we can observe the results of the evaluation, using the confidences for language detection with higher confidence of 0.5 for all the languages and using the lid.176.bin model. As we can see, the values for precision for a SBERT score of 0.6 or higher are very good (100% for Spanish–Asturian and 84.36% for Spanish–Catalan. But for Spanish–Asturian the recall values are very low, of about 21%. Using this configuration in a real scenario would probably lead to missing a lot of correct parallel segments, at least for the Spanish–Asturian language pair.

If we now observe the results in table 5, where the newly trained language detection model is used, we can see that the recall problems in the Spanish–Asturian language pair now disappear, with no degradation of the precision figures. As far as the Spanish–Catalan language pair is concerned, we now observe a significant improvement in the precision values, while the recall values are maintained and even improved.

This experiment leads us to conclude that the language detection model plays a very important role in the filtering and rescoring process of the corpus. The use of a language detection model tailored to the corpus to be cleaned leads to a much better performance.

4.4 Filtered CCMatrix corpora

In table 6 we can observe the number of sentences after the filtering process for the CCMatrix Spanish–Asturian and Spanish–Catalan using the lid.176.bin with confidence 0.5 for both languages and for several values of the SBERT score. In table

	Asturian			Catalan			Spanish		
conf.	P	R	F_1	P	R	F_1	P	R	F_1
0.9	100	1.24	2.46	98.83	75.79	85.90	92.11	94.12	93.13
0.8	100	4.83	9.21	96.30	89.35	92.69	81.09	98.86	89.10
0.7	100	9.31	17.03	92.97	93.48	93.22	73.08	99.45	84.25
0.6	100	15.88	27.41	89.04	96.27	92.51	67.60	99.70	80.57
0.5	100	21.75	35.73	84.12	97.81	90.45	62.67	99.95	77.04
0.4	100	27.82	43.54	78.98	98.95	87.85	58.78	99.85	74.03
0.3	100	30.91	47.22	76.74	99.35	86.59	57.19	99.95	72.75
0.2	100	31.86	48.32	75.62	99.45	85.95	56.69	99.95	72.35
0.1	100	31.86	48.43	75.60	99.45	85.90	56.66	99.95	72.32
0	100	32.01	48.49	75.57	99.45	85.88	56.66	99.95	72.32

Table 2: Evaluation of language detection with model lid.176.bin

	Asturian			Catalan			Spanish		
conf.	P	R	F_1	P	R	F_1	P	R	F_1
0.9	100	98.11	99.05	100	99.00	99.50	100	97.76	98.873
0.8	99.95	98.66	99.30	100	99.45	99.73	100	99.95	99.47
0.7	99.95	99.30	99.63	100	99.75	99.88	100	99.40	99.70
0.6	99.95	99.45	99.70	99.95	99.85	99.90	99.80	99.70	99.75
0.5	99.95	99.60	99.78	99.95	99.95	99.95	99.70	99.90	99.80
≤ 0.4	99.95	99.65	99.80	99.95	99.95	99.90	99.65	99.90	99.78

Table 3: Evaluation of language detection with the newly trained model

7 we can observe the same figures when using the newly trained language detection model.

For the Spanish–Asturian corpus, the number of segments of the filtered corpus is much larger for the newly trained language detection model, by a factor of almost 3 for all SBERT scores. This may mean that, with the lid.176.bin model, many segments written in Asturian are detected as being written in another language, and thus filtered out, regardless of the SBERT score.

On the other hand, the number of segments of the filtered corpus is smaller for the Spanish–Catalan corpus when using the newly trained language detection model, by a factor of about 1.4 for most of the SBERT scores. This fact demonstrates the importance of selecting the appropriate language model when filtering parallel corpora with the proposed methodology.

In future experiments, we plan to manually evaluate the resulting filtered corpora. We also plan to evaluate this method in the task of training neural machine translation systems with several of the filtered corpora and the original one. The trained NMT systems will be evaluated using automatic metrics. These evaluation results will shed light on the quality-quantity in relation to the training corpora for NMT systems.

5 Conclusion and future work

In this paper, we have presented a simple strategy to select the higher quality segments from a large parallel corpus. This strategy is based on verifying the languages of the segments and on scoring the parallel segments with SBERT. The methodology has been implemented in a Python script holding a free licence that can be downloaded from Github.[19]. Filtered versions of the CCMatrix corpus for several language pairs are available for download.

In a future work we plan to further evaluate this strategy training and evaluating neural machine translation systems with the raw and cleaned versions of the corpora for several language pairs.

We plan to use this strategy for further cleaning the parallel corpora available in the Opus Corpus collection[20] (Tiedemann, 2012) for the languages of the project TAN-IBE (Neural Machine Translation for the romance languages of the Iberian

[19]https://github.com/aoliverg/MTUOC-PCorpus-rescorer
[20]https://opus.nlpl.eu/

	Spanish–Asturian			Spanish–Catalan		
conf.	P	R	F_1	P	R	F_1
0.9	100	6.12	11.54	93.23	65.50	77.63
0.8	100	16.43	28.22	87.25	95.62	91.26
0.7	100	20.11	33.49	84.96	97.56	90.82
0.6	100	21.20	34.99	84.36	97.76	80.57
0.5	100	21.60	35.53	84.11	97.76	90.42
0.4	98.20	21.70	35.55	84.11	97.76	90.42
0.3	82.61	21.75	34.44	84.12	97.81	90.45
0.2	59.70	21.75	31.89	84.05	97.81	90.41
0.1	51.47	21.75	30.58	84.01	97.81	90.39

Table 4: Evaluation of SBERT capability to select correct translations. For language detection, lid.176.bin model is used with confidence 0.5 for both languages.

	Spanish–Asturian			Spanish–Catalan		
conf.	P	R	F_1	P	R	F_1
0.9	100	27.18	42.74	100	67.89	80.88
0.8	100	76.06	86.40	99.95	97.71	98.85
0.7	100	92.48	96.10	99.95	99.65	99.8
0.6	100	94.91	98.94	99.95	99.85	99.90
0.5	99.75	99.15	99.45	99.95	99.85	99.90
0.4	97.60	99.30	98.45	99.95	99.85	99.90
0.3	82.67	99.50	90.31	99.95	99.90	99.90
0.2	59.15	99.5	74.21	99.95	99.85	99.93
0.1	51.16	99.50	67.59	99.95	99.85	99.93

Table 5: Evaluation of SBERT capability to select correct translations. For language detection, a newly trained model is used with confidence 0.5 for both languages.

score	spa–ast	spa–cat
0.9	126,526	35,495,245
0.8	170,491	45,848,066
0.7	183,074	52,120,334
0.6	199,780	55,207,461
0.5	258,113	56,308,989
0.4	418,225	56,624,672
0.3	737,022	56,703,000
0.2	1,162,165	56,719,624
0.1	1,417,611	56,722,271

Table 6: Size of the filtered corpora using the lid.176.bin model

score	spa–ast	spa–cat
0.9	372,317	23,639,411
0.8	496,931	30,547,417
0.7	539,569	34,874,013
0.6	590,249	37,046,724
0.5	749,993	37,886,522
0.4	1,202,697	38,943,338
0.3	2,264,739	43,882,800
0.2	3,915,887	51,739,619
0.1	4,948,002	55,510.090

Table 7: Size of the filtered corpora using the lid.176.bin model

Peninsula): Spanish, Portuguese, Catalan, Galician Asturian, Aragonese and Aranese.

Acknowledgments

This work is partially supported by the project *TAN-IBE: Neural Machine Translation for the romance languages of the Iberian Peninsula*, founded by the Spanish Ministry of Science and Innovation Proyectos de generación de conocimiento 2021. Reference: PID2021-124663OB-I00.

References

Abadji, Julien, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642, January.

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

Bañón, Marta, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 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.

Chaudhary, Vishrav, 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, August. Association for Computational Linguistics.

de Gibert Bonet, Ona, Ksenia Kharitonova, Blanca Calvo Figueras, Jordi Armengol-Estapé, and Maite Melero. 2022. Quality versus quantity: Building Catalan-English MT resources. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 59–69, Marseille, France, June. European Language Resources Association.

El-Kishky, Ahmed, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, November.

Goyal, Naman, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Ramírez-Sánchez, Gema, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 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.

Reimers, Nils and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Schwenk, Holger 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, August. Association for Computational Linguistics.

Schwenk, Holger, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online, April. Association for Computational Linguistics.

Schwenk, Holger, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online, August. Association for Computational Linguistics.

Tiedemann, Jörg. 2012. Parallel data, tools and interfaces in opus. In Chair), Nicoletta Calzolari (Conference, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Xu, Hainan and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950, Copenhagen, Denmark, September. Association for Computational Linguistics.

Zaragoza-Bernabeu, Jaume, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz-Rojas. 2022. Bicleaner ai: Bicleaner goes neural. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 824–831.

Zhang, Boliang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8545–8554, Online, July. Association for Computational Linguistics.