An Unsupervised System for Parallel Corpus Filtering

Viktor Hangya and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich, Germany
{hangyav, fraser}@cis.lmu.de

Abstract

In this paper we describe LMU Munich's submission for the *WMT 2018 Parallel Corpus Filtering* shared task which addresses the problem of cleaning noisy parallel corpora. The task of mining and cleaning parallel sentences is important for improving the quality of machine translation systems, especially for low-resource languages. We tackle this problem in a fully unsupervised fashion relying on bilingual word embeddings created without any bilingual signal. After pre-filtering noisy data we rank sentence pairs by calculating bilingual sentence-level similarities and then remove redundant data by employing monolingual similarity as well. Our unsupervised system achieved good performance during the official evaluation of the shared task, scoring only a few BLEU points behind the best systems, while not requiring any parallel training data.

1 Introduction

Machine translation is important for eliminating language barriers in everyday life. To train systems which can produce good quality translations large parallel corpora are needed. Mining parallel sentences from various sources in order to train better performing MT systems is essential, especially for low resource languages. Previous efforts[1] showed that it is possible to crawl parallel data from the web, but also showed that additional steps are necessary to filter noisy sentence pairs. In this paper we introduce our approach to filter noisy parallel corpora without the need of any initial bilingual signal to train the filtering system.

We participate in the *WMT 2018 Parallel Corpus Filtering* shared task with our system which tackles the problem of selecting the best quality sentence pairs for training both statistical and neural MT systems (Koehn et al., 2018). A lot of previous work has studied the problem of parallel data cleaning. Esplà-Gomis and Forcada (2010) proposed BiTextor which filters data based on sentence alignment scores and URL information. Similarly, word alignments and language modeling were used in (Denkowski et al., 2012) to select sentence pairs that are useful for training an MT system. Xu and Koehn (2017) proposed Zipporah, a logistic regression based model that uses bag-of-words translation features to measure fluency and adequacy in order to score sentence pairs. Another line of work is to select data based on the target domain. A static sentence-selection method was used for domain adaptation based on the internal sentence embedding of NMT (Wang et al., 2017) while van der Wees et al. (2017) used domain-based cross-entropy as a criterion to gradually fine-tune the NMT training in a dynamic manner. In contrast with previous work, we do not rely on any bilingual supervision, making our approach applicable to language pairs which lack initial parallel resources. Similarly to the work of Kajiwara and Komachi (2016), where word embeddings were used to mine monolingual sentence pairs for text simplification, we use a word level metric to compute sentence pair similarity in a computationally efficient way.

Our approach consists of three steps. Due to the noisiness of the input data we use a pre-filtering step which detects sentences which are not useful. We developed a simple rule-based method which looks for sentence pairs which for example came from the wrong languages or have significantly different lengths. As a second step, we calculate sentence pair similarities using bilingual word embeddings and orthographic information. In the third step, we perform post-ranking where we counterweight source language sentences which

[1] https://paracrawl.eu

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 882–887

https://doi.org/10.18653/v1/W18-64104

are less fluent or redundant using language modeling and monolingual document similarity respectively. Our system is fully unsupervised, i.e., we do not use any parallel data for the training of our methods. We show results on the official test sets of the shared task which includes six datasets from different sources. Although, our method is fully unsupervised it achieves good performance on the extrinsic task of training MT systems on the filtered parallel data, scoring only 2.17 BLEU points behind the best systems.

2 Approach

In this section we introduce our approach for the filtering task. Since parallel sentence mining is most crucial for resource-poor languages our goal was to develop a system that does not need any bilingual signal for training. Our approach is based on recent developments in the field of bilingual word embeddings, i.e., it was shown that good quality bilingual embeddings can be trained using only source and target language monolingual data (Conneau et al., 2017). As was mentioned in the previous section our approach consists of three steps which we introduce below. In each step we score the input candidate sentence pairs which are used at the sampling step to select sentence pairs before the training of MT systems. Higher score means higher probability for being selected during the sampling process. For more detail about the data, the preprocessing and the sampling procedure see section 3.

2.1 Pre-Filtering

The input data, released by the shared task organizers, contain a large amount of erroneous candidate sentence pairs which can be filtered out based on some simple heuristics. For detecting these instances we use the following rules and set the weight of these noisy candidate pairs to zero. Note, we ignore the candidates selected here in later steps for reasons of speed.

1. **Hunalign** scores of the sentence pairs were released with the data. We ignore candidates if the initial score is less then 0.0.

2. If either of the sentences has a **length** of less then 3 tokens we consider it as noise.

3. A good indicator of bad alignment of sentences is their **length difference**. If this value is greater than 15 tokens we set its weight to zero.

4. We also consider a candidate as noise if the **number and URL ratio**, compared to the number of all tokens, is greater than 0.6.

5. In many cases the **language** of the sentences is incorrect. We use the system of Sarwar et al. (2001) to detect these instances.

2.2 Scoring

In the main step of our approach we calculate the score of a candidate sentence pair based on the similarities of the contained words. First, we describe how we train bilingual word embeddings and then we describe the method for sentence similarity.

Bilingual word embeddings Recently, Conneau et al. (2017) showed that good quality bilingual embeddings can be produced by training monolingual word embedding spaces for both source and target languages and mapping them to a shared space without any bilingual signal. We follow this approach and use bilingual word embeddings, trained in an unsupervised fashion. For this we use the system released by (Conneau et al., 2017). We discuss the used data and parameters in section 3.

Sentence pair similarity Given a candidate pair of source and target sentences S and T, the similarity score is calculated by iterating over the words in S from left to right and pairing each word $s \in S$, in a greedy fashion, with the word $t \in T$ that has the highest cosine similarity based on our dictionary. We then greedily eliminate t from T, so that it cannot be matched by a later word "s". Then, the averaged word-pair similarity gives the final score. We remove stopwords, digits and punctuation from texts before calculating similarity. Note, this idea is similar to *Word Movers Distance* introduced in (Kusner et al., 2015) but simpler due to runtime considerations on huge corpora.

As was shown in previous work (Braune et al., 2018), the quality of bilingual word similarity can be significantly improved by using orthographic cues, especially for rare words. We extend this idea to the sentence level by using a dictionary containing orthographically similar source-target

language word pairs and their similarity. We define orthographic similarity as one minus the normalized Levenshtein distance. We use this orthographic dictionary together with the BWE-based dictionary when mining parallel sentences by using the higher value from the two dictionaries. If the given word pair is not in a dictionary we consider their similarity as 0.0 for that dictionary. One issue with orthographic similarity of words is that it tends to give high scores to sentences which contain many orthographically similar words, e.g., a sentence with a list of named entities, which are often not useful for MT systems. To overcome this issue, we multiply the orthographic word similarities with 0.2.

2.3 Post-Ranking

In the third step we re-rank candidates from the previous step in order to reduce the number of redundant sentence pairs and to ensure that we have more fluent sentences. We apply these steps only to the source sentences due to speed considerations.

Monolingual Document Similarity The input corpus contains redundant sentences, i.e., sentences which have similar structure and meaning, and which are often generated based on predefined sentence templates. It is enough to use only one element from these clusters of redundant sentences since the rest does not have a big impact on the translation quality. Due to the huge size of the input data we used a simple thus fast approach to detect redundant sentences and decrease their score. First, we embed each source side sentence to a fixed sized sentence embedding by simply averaging the word embeddings of the words in the sentence. We calculate sentence similarities of each possible pairs which can be done efficiently even for large inputs (Johnson et al., 2017). We use cosine as the similarity metric and we consider those sentences as redundant which have lower difference than 0.02 between the similarity value of its top two most similar sentences. We multiply the original score of redundant sentences by 0.5.

Language model It is beneficial to use fluent sentences for training MT systems. To take this aspect into consideration we used KenLM language model (Heafield et al., 2013) to change the score of a candidate pair based on the source side sentence's normalized language model probability. We multiply scores if the given sentence has higher (lower) probability than 1×10^{-3} (5×10^{-6}) by 1.5 (0.5).

3 Experimental Setup

The goal of the shared task is, given a noisy parallel corpus, to filter candidate sentence pairs that are most useful for training MT systems. Candidate pairs have to be scored based on the predicted quality of the corresponding candidate where the scores do not have a special meaning except that higher values indicate better quality. To produce the actual training data for the MT systems the scored corpus is sampled using an official tool, released by the organizers, which samples sentences with a probability proportional to their scores.

3.1 Data

A German-English dataset was released containing 1 billion (English) tokens. The corpus was crawled from the web as part of the *ParaCrawl* project. After extracting texts from web pages with BiTextor (Esplà-Gomis and Forcada, 2010), documents and sentences were aligned using (Buck and Koehn, 2016) and Hunalign (Varga et al., 2007) respectively. The aligned sentence pairs are the candidates which have to be scored for the sampling process and used as training parallel data for the MT systems. The alignment scores of the candidate sentence pairs were also released which do not by themselves correlate strongly with sentence pair quality which we show in section 4. For more details of the data see the overview paper of the shared task (Koehn et al., 2018). As an additional data source we use monolingual German and English NewsCrawl sentences from the time period between 2011 and 2014 (Bojar et al., 2014) which we use to train word embeddings and the language model.

3.2 Evaluation

To evaluate systems two setups were performed: (i) sampling 10M tokens and (ii) 100M tokens from the scored corpus using the released sampler tool. The quality of the resulting subsets is determined by the quality of a German-English SMT (Koehn et al., 2007) and an NMT (Junczys-Dowmunt et al., 2018) system trained on this data and using BLEU to measure translation quality. We will refer to these setups as SMT 10M, SMT 100M, NMT 10M and NMT 100M. As development set newstest 2017 was used, while newstest

		newstest 2017	newstest 2018	iwslt2017	Acquis	EMEA	Global Voices	KDE	avg
SMT 10M	lmu-ds-lm	21.73	28.03	20.61	17.97	26.95	21.45	24.73	23.29
	lmu-ds	21.71	28.03	20.57	17.96	26.96	21.46	24.58	23.26
	lmu	19.62	25.35	19.67	15.30	25.32	20.03	23.08	21.46
SMT 100M	lmu-ds-lm	24.86	30.14	22.42	21.47	30.08	23.09	26.20	25.57
	lmu-ds	24.86	30.00	22.31	21.25	30.18	23.19	26.11	25.51
	lmu	25.09	30.34	22.37	20.98	30.44	23.27	26.24	25.61
NMT 10M	lmu-ds-lm	26.17	31.89	22.40	18.51	27.01	24.60	17.46	23.65
	lmu-ds	26.22	31.79	22.09	18.43	27.14	24.53	17.94	23.65
	lmu	23.03	28.79	21.06	16.01	26.98	23.30	21.64	22.96
NMT 100M	lmu-ds-lm	29.14	36.99	25.48	25.19	33.46	27.52	28.17	29.47
	lmu-ds	29.33	36.71	25.48	25.25	34.15	27.67	27.95	29.54
	lmu	30.82	37.78	25.95	25.77	35.61	28.48	29.62	30.54

Table 1: BLEU scores of our setups on the different datasets. We underline best results for each setup and dataset.

Insitution	SMT 10M	SMT 100M	NMT 10M	NMT 100M
RWTH	24.58	26.21	28.01	31.29
Microsoft	24.45	26.50	28.62	32.06
Alibaba	24.11	26.44	27.60	31.93
NRC	23.89	26.40	27.41	31.88
Speechmatics	23.88	25.85	27.97	31.00
NICT	23.46	25.98	25.94	30.04
AFRL	23.36	25.32	27.09	30.28
Vicomtech	23.29	25.91	26.35	30.40
LMU	23.29	25.61	23.65	30.54
Tilde	23.03	26.19	26.56	31.24
Prompsit	22.94	26.41	26.05	31.83
ARC	22.68	26.13	25.79	31.34
JHU	22.61	25.84	25.41	30.16
MAJE	22.53	26.07	24.81	31.20
Univ. Tartu	22.31	25.70	25.17	30.60
Systran	21.83	25.44	24.30	29.91
UTFPR	20.81	22.35	21.75	22.23
DCU	15.67	21.19	6.27	18.60

Table 2: Best systems of participants on the four setups averaged over all test sets.

2018, iwslt2017, Acquis, EMEA, Global Voices and KDE were the undisclosed test sets (Koehn et al., 2018).

3.3 Parameter setup

We preprocessed all data using the tokenizer from Moses with aggressive mode (Koehn et al., 2007) and lower casing. To train monolingual word embeddings we used FastText (Bojanowski et al., 2016) with default parameters except the dimension of the vectors which is 300. As input the concatenation of the shared task data and NewsCrawl was used. For the unsupervised mapping we ran (Conneau et al., 2017) using the source and target language monolingual spaces. As a language model we used KenLM (Heafield et al., 2013), with n-gram size 5 and using default values for the rest of the parameters, on the source side of our data. All other parameters introduced earlier are based on manual analysis of the data and non-exhaustive tuning on the development set. During development we only run SMT 10M due to time constraints.

4 Results

We present official BLEU scores of our systems on the four setups and seven datasets in table 1. Our default system *lmu* applies pre-filtering and scoring and we incrementally add monolingual document similarity and language modeling post-ranking steps. During development we calculated the performance of only applying the pre-filtering step on newstest 2017 with SMT 10M which resulted in a score of 15.53 BLEU while the released hunalign scores resulted in a score of 6.88. This result shows the noisiness of the data and the importance of pre-filtering.

Based on table 1 it can be seen that our default system, without post-ranking, could already achieve good performance. The additional post-ranking steps were most helpful for the setups with only 10M tokens in the training data. This indicates that giving less weight to redundant and not fluent sentences is especially important in the low resource setups. During the development we also performed an ablation study on the post-ranking methods. Using only the language model on top of pre-filtering and scoring gave 20.67 BLEU points while activating only the document similarity module we got 21.66 with SMT 10M. This shows that the latter method is more important because it removes more redundant data from the training set and makes space for sentence pairs that contain additional lexical information. On the other hand, language modeling causes lower performance increase because the rule-based pre-filtering step could already detect and remove some of the less fluent candidates. By combining the two techniques we could achieve

the best performance on the newstest 2017 dataset. In contrast, post-ranking steps only helped for the iwslt2017 and Acquis datasets in the case of the 100M token setups. We conjecture that the down-weighting of candidates by these steps was too heavy which resulted in lower importance of these candidates comparing to candidates which are not even parallel. This issue could be overcome by better fine tuning of hyperparameters.

In table 2 we show the averaged results over all test sets of the best system of the official participants. Our systems performs better then the average in three out of four cases and scores below the best system by only 2.17 BLEU points on average. Our results are less competitive with NMT which is because we only used SMT during development. Our results show that competitive performance can be achieved without the use of any bilingual signal for the parallel corpus filtering task.

5 Conclusion

In this paper we introduced LMU Munich's submission to the WMT 2018 Parallel Corpus Filtering shared task. Such systems are especially useful in low resource setups, so we proposed a fully unsupervised system which is built on three modules: (i) we apply a pre-filtering step to remove noisy data (ii) we score sentences based on bilingual word embeddings and (iii) as a post-ranking step we penalize sentence pairs which are redundant or not fluent enough. We achieved good results with all setups which shows the competitiveness of our unsupervised system.

Acknowledgments

We would like to thank Fabienne Braune, Yuliya Kalasouskaya and the anonymous reviewers for their valuable input. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58.

Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. 2018. Evaluating bilingual word embeddings on the long tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 188–193.

Christian Buck and Philipp Koehn. 2016. Quick and reliable document alignment via tf/idf-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *CoRR*, abs/1710.04087.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 261–266.

Miquel Esplà-Gomis and Mikel Forcada. 2010. Combining content-based and url-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathematical Linguistics*, pages 77–86.

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

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 116–121.

Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics on interactive poster and demonstration sessions*, pages 177–180.

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

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, page 247.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. *arXiv preprint arXiv:1708.00712*.

Hainan Xu 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.