# NICT's Corpus Filtering Systems for the WMT18 Parallel Corpus Filtering Task

**Rui Wang** **Benjamin Marie*** **Masao Utiyama** **Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Souraku-gun, Kyoto, 619-0289, Japan
{wangrui, bmarie, mutiyama, eiichiro.sumita}@nict.go.jp

## Abstract

This paper presents the NICT's participation in the WMT18 shared parallel corpus filtering task. The organizers provided 1 billion words German-English corpus crawled from the web as part of the Paracrawl project. This corpus is too noisy to build an acceptable neural machine translation (NMT) system. Using the clean data of the WMT18 shared news translation task, we designed several features and trained a classifier to score each sentence pairs in the noisy data. Finally, we sampled 100 million and 10 million words and built corresponding NMT systems. Empirical results show that our NMT systems trained on sampled data achieve promising performance.

## 1 Introduction

This paper describes the corpus filtering system built for the participation of the National Institute of Information and Communications Technology (NICT) to the WMT18 shared parallel corpus filtering task.

NMT has shown large gains in quality over Statistical machine translation (SMT) and set several new benchmarks (Bojar et al., 2017). However, NMT is much more sensitive to domain (Wang et al., 2017) and noise (Khayrallah and Koehn, 2018). The reason is that NMT is a single neural network structure, which would be affected by each instance during the training procedure (Wang et al., 2017). In comparison, SMT is a combination of distributed models, such as a phrase-table and a language model. Even if some instances in the phrase-table or the language model are noisy, they can only affect part of the models and would not affect the entire system so much. To the best of our knowledge, there are only few works investigating the impact of the noise problem in NMT (Xu and Koehn, 2017; Belinkov and Bisk, 2017).

In this paper, we focus on the performance of NMT trained on noisy parallel data. We adopt the clean data of WMT18 News Translation Task to train a classifier and compute informative features. Using this classifier, we score each sentence in the noisy data and sample the top ranked sentences to construct the pseudo clean data. The new pseudo clean data are used to train a robust NMT system.

The remainder of this paper is organized as follows. In Section 2, we introduce the task and data. In Section 3, we introduce the features that we designed to score sentences in the noisy corpus. We use these features to train a classifier and the sentences in the noisy corpus are scored by this classifier. Empirical results produced with our systems are showed and analyzed in Section 4, and Section 5 concludes this paper.

## 2 Task Description

WMT18 shared parallel corpus filtering task[1] (Koehn et al., 2018) provides a very noisy 1 billion words (English word count) German-English (De-En) corpus crawled from the web as a part of the Paracrawl project. Participants are asked to provide a quality score for each sentence pair in the corpus. Computed scores are then evaluated given the performance of SMT and NMT systems trained on 100M and 10M words sampled from data using the quality scores computed by the participants. *newstest2016* is used as the development data and the test data include *newstest2018*, *iwslt2017*, *Acquis*, *EMEA*, *Global Voices*, and *KDE*.[2] The statistics of the noisy data to filter are shown in Table 1.

The participants may use the WMT18 News

*The first two authors have equal contributions.

[1] http://www.statmt.org/wmt18/parallel-corpus-filtering.html

[2] Note that, except for *newstest2018*, all testsets remained unknown from the participants until the submission deadline.



*Proceedings of the Third Conference on Machine Translation (WMT)*, Volume 2: Shared Task Papers, pages 963–967

https://doi.org/10.18653/v1/W18-64116

| Language | #lines | #words | #tokens |
|---|---|---|---|
| En | 104.00 M | 1.00B | 1.66B |
| De | 104.00 M | 0.96B | 1.62B |

Table 1: Statistics of the noisy data to filter. "#words" indicates the word count before tokenization.

Translation Task data[3] for German-English (without the Paracrawl parallel corpus) to train components of their method. In addition, to participate in the shared task, participants have to submit a file with quality scores, one score per line, corresponding to the sentence pairs. The scores do not have to be meaningful, except that higher scores indicate better quality.

## 3 Sentence Pairs Scoring

The task requires to give a score to each sentence pair in the corpus to filter. We performed first an aggressive filtering (Section 3.1) to avoid scoring sentence pairs that are clearly too noisy to be used during the training of MT systems. Then, we computed informative features (Section 3.2) for each one of the remaining sentence pairs. Then, according to the feature scores, a classifier computes a global score for each sentence pair that can be used to rank them.

### 3.1 Aggressive Filtering

After a quick observation of the data, we first decided to perform an aggressive filtering since it appeared that many of the sentence pairs are obviously too noisy to be used to train MT systems. For instance, many sentences in the corpus are made of long sequences of numbers or punctuation marks. We decided to give a score of 0.0 to all the sentence pairs that contain a sentence made of tokens that are, for more than 25% them, numbers or punctuation marks. We also had to take into account the sentence length: very short source sentences are more likely to be paired with a good translation in the corpus, and our classifier may give to such pairs very high scores. Then, in order to avoid a filtering that keeps sentences made in majority of very short and redundant sentences, that are not very useful to train NMT systems, we also give a score of 0.0 to all the sentence pairs that contain a source or a target sentence that contains less than four tokens. We also give a score of 0.0 to all the sentence pairs that contain a sentence longer than 80 tokens since the default parameters of the SMT system used for evaluation filter out sentences longer than that.

This aggressive filtering excluded 69% of the sentence pairs, leaving us a much reduced quantity of sentence pairs to be scored by our classifier.

### 3.2 Features

We scored each of the remaining sentence pairs with four NMT transformer models, trained with Marian (Junczys-Dowmunt et al., 2018)[4], on all the parallel data provided for the shared news translation task (excluding the "paracrawl" corpus). We trained left-to-right and right-to-left models for German-to-English and English-to-German translation directions. We used these four model scores as features in our classifier.

We also trained lexical translation probability with Moses and used them to compute a sentence-level translation probability, for both translation directions, as proposed by Marie and Fujita (2017).

To evaluate the semantic similarity between the source and target sentence, we compute a feature based on bilingual word embeddings as follows. First, we trained monolingual word embeddings with FastText (Bojanowski et al., 2017)[5] on the monolingual English and German data provided by the WMT organizers. Then, we aligned English and German monolingual word embedding spaces in a bilingual space using the unsupervised method proposed by Artetxe et al. (2018).[6] Given the bilingual word embeddings, we computed embeddings for the source and target sentence by doing the element-wise addition of the bilingual embedding of the words they contain. Finally, we computed the cosine similarity between the embeddings of source and target sentence for each sentence pair, and used it as a feature.

Other features are computed to take into account the sentence length: the number of tokens in the source and target sentences, and the difference, and its absolute value, between them. We summarize the features that we used in Table 2.

[3] http://www.statmt.org/wmt18/translation-task.html

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

[5] We used the default parameters for skipgram, with 512 dimensions.

[6] We used the implementation provided by the authors, with default parameters, at: https://github.com/artetxem/vecmap.

| Feature | Description |
|---|---|
| L2R (2) | Scores given by the left-to-right German-to-English and English-to-German NMT models |
| R2L (2) | Scores given by the right-to-left German-to-English and English-to-German NMT models |
| LEX (4) | Lexical translation probabilities, for both translation directions |
| WE (1) | Bilingual sentence embedding similarity |
| LEN (4) | Length-based features |

Table 2: Set of features used by our classifier.

### 3.3 Classifier

We chose a logistic regression classifier to compute a score for each sentence pair using the features presented in Section 3.2. We trained our classifier on *Newstest2014*, that we used as positive examples of good sentence pairs, and created the same number of negative examples using the following procedure. We created three-type of negative examples, each of which contains one third of the sentence number of *Newstest2014*:

- Misaligned: The target sentences are wrongly aligned to the previous or following source sentences.
- Wrong translation: some words in a sentence are replaced by random words from the vocabulary.
- Misordered words: we shuffled the words in a sentence.

We used the same procedure to create training data with *Newstest2015*, and used it to tune the regularization parameter of our classifier. The classifier accuracy is 78.9% on *Newstest2015*.

We used the probability returned by the classifier for each sentence pair as the score to be used to perform filtering.

## 4 NMT Systems and Results

For this task, we did not conduct experiments with a state-of-the-art NMT system, because the organizers fixed the data and systems settings for a fair comparison.

### 4.1 NMT Systems

For the data preprocessing, we strictly followed the data preparation (including tokenization, truecasing, and byte pair encoding) provided by the organizers. To train NMT systems, we used the provided official settings of Marian, which can be found at the WMT offical website[7] and the Appendix A. All our NMT systems were trained on four Nvidia Tesla P100 GPUs.

Our settings were the same for all of the NMT systems. For each method, we use their score to select the top 100M and 10M sentences to train the corresponding NMT systems. In Table 4, "Original" means the original corpus without any filtering. "Aggressive Filtering" is the method which we introduced in Section 3.1. "Hunalign" indicates the baseline corpus filtering method (Varga et al., 2007)[8] given by the organizers. "Classifier" indicates the classifier that we proposed in Section 3.3. "Classifier + LangID" indicates that we also use a language identification tool, LangID (Lui and Baldwin, 2012)[9], to filter the sentence pairs containing sentences that are not German or English. The results were evaluated on the development data *newstest2016*.

### 4.2 NMT Performance

From the results in Table 4, we have the following observations:

- The proposed "Aggressive Filtering" reduced 69% sentences and improved 1.5 BLEU compared to using the original corpus. This indicates that most of the noisy data can be filtered by the aggressive filter.
- The baseline "Hunalign" did not perform very well, the performance decreased to 3.6/0.03 by selecting 100/10M sentences. Especially when selecting 10M sentences, the NMT system nearly did not work.
- The proposed "Classifier" significantly improved NMT performance by more than 20

[7] http://www.statmt.org/wmt18/parallel-corpus-filtering-data/dev-tools.tgz

[8] http://mokk.bme.hu/resources/hunalign/

[9] https://github.com/saffsd/langid.py

| System | newstest2018 | iwslt2017 | Acquis | EMEA | Global Voices | KDE | average |
|---|---|---|---|---|---|---|---|
| SMT-10M | 27.79 | 20.94 | 19.27 | 25.89 | 21.38 | 25.51 | 23.46 |
| SMT-100M | 30.79 | 22.76 | 21.98 | 30.39 | 23.63 | 26.55 | 25.98 |
| NMT-10M | 32.93 | 23.67 | 21.67 | 27.60 | 25.13 | 24.65 | 25.94 |
| NMT-100M | 37.28 | 25.83 | 26.11 | 34.13 | 27.62 | 29.25 | 30.04 |

Table 3: WMT official results.

| Methods | #tokens (En) | #lines | #BLEU |
|---|---|---|---|
| Original | 1.6B | 104.0M | 7.4 |
| Aggressive Filtering | 584M | 31.9M | 8.8 |
| Hunalign | 100M | 8.7M | 3.6 |
| Classifier | 100M | 9.1M | 26.1 |
| Classifier + LangID | 100M | 6.7M | 31.6 |
| Hunalign | 10M | 2.6M | 0.03 |
| Classifier | 10M | 1.2M | 25.6 |
| Classifier + LangID | 10M | 0.9M | 27.8 |

Table 4: Results on the development data.

| Methods | #tokens (En) | #Time |
|---|---|---|
| Original | 1.6B | 43 hours |
| Aggressive Filtering | 584M | 47 hours |
| Classifier + LangID | 100M | 55 hours |
| Classifier + LangID | 10M | 11 hours |

Table 5: Training efficiency.

BLEU. This indicates that the proposed classifier can rank sentence by a proper order and the more useful sentences are selected.

- The "Classifier + LangID" achieved further approximately 2∼5 BLEU improvement. This indicates there are several sentences which are not proper languages and they can be detected by the LangID.

- For the proposed method, the systems built from 100M sentences performed much better than the ones built from 10M sentences. This indicates that filtering too many sentences will harm the NMT performance.

### 4.3 Training Efficiency

Besides the NMT performances, we also showed the training efficiency in Table 5.

The results in Table 5 showed:

- The training time of using 1.6B, 584M, and 100M sentences was very close.

- The training time of using 10M sentences was quite faster than the other ones. Together with the performance results in Table 4, it show that these 10M contains most of the useful information in the entire corpus and can accelerate NMT training significantly.

### 4.4 Official Results

We reported the official results of our submitted system "Classifier + LangID" in Tables 3. In the official results, both SMT and NMT results were reported.

From the results in Table 3, we have the following observations:

- The NMT system performed much better than corresponding SMT systems. This indicates that the proposed method can help NMT in overcoming the noise problem.

- The systems built from 100M sentences performed much better than the ones built from 10M sentences. This is consistent with the results obtained on the development data.

- Compared with other teams, the rankng of our SMT systems performed better than our NMT systems. The reason may be that we used several features from SMT. We ranked the first in the KDE SMT-10M task.

## 5 Conclusion and Future Work

In this paper, we investigated the noisy data problem in NMT. We designed a classification system to filter the noisy data for the WMT18 shared parallel corpus filtering task and built NMT systems using the selected data.

The empirical results showed that most of the sentence pairs in the corpus are noisy. By removing these sentence pairs, the training corpus can be reduced up to 1% of the original one while training a significantly better NMT system than the original NMT system trained on all the data. In our future work, we would like to investigate the impact of each type of noise and the effect of each feature used by our classifier.

In this paper, we focused on supervised classification methods. That is, we used clean data as a gold standard. In our future work, we would like

to investigate this task using unsupervised methods. That is, we only use the noisy data and let NMT itself detect noisy sentence pairs.

## Acknowledgments

This work is partially supported by the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of MIC, Japan.

## A Marian Settings

To train NMT systems, we used the provided settings of Marian:

```
--sync-sgd -T --devices 0
1 2 3 --mini-batch-fit -w
3000 --dim-vocabs 50000
50000 --layer-normalization
--dropout-rnn 0.2 --dropout-src
0.1 --dropout-trg 0.1
--learn-rate 0.0001
--after-epochs 0 --early-stopping
5 --max-length 80 --valid-freq
20000 --save-freq
20000 --disp-freq 2000
--valid-mini-batch 8
--valid-metrics cross-entropy
perplexity translation --seed
1111 --exponential-smoothing
--normalize=1 --beam-size=12
--quiet-translation.
```

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer. 2017. Proceedings of the second conference on machine translation. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics.

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 ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 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. Association for Computational Linguistics.

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*, Brussels, Belgium. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2017. Efficient extraction of pseudo-parallel sentences from raw monolingual data using word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 392–398. Association for Computational Linguistics.

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*, 292:247.

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, Copenhagen, Denmark. Association for Computational Linguistics.

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. Association for Computational Linguistics.