Webinterpret Submission to the WMT2018 Shared Task on Parallel Corpus Filtering

Marina Fomicheva*
AT Language Solutions
mari.fomicheva@gmail.com

Jesús González-Rubio
WebInterpret
jesus.g.rubio@gmail.com

Abstract

This paper describes the participation of Webinterpret in the shared task on parallel corpus filtering at the Third Conference on Machine Translation (WMT 2018). The paper describes the main characteristics of our approach and discusses the results obtained on the data sets published for the shared task.

1 Task description

Parallel corpus filtering task at WMT18[1] tackles the problem of cleaning noisy parallel corpora. Given a noisy parallel corpus (crawled from the web), participants develop methods to filter it to a smaller size of high quality sentence pairs.

Specifically, the organizers provide a very noisy 1 billion word German–English corpus crawled from the web as part of the Paracrawl project[2]. Participants are asked to select a subset of sentence pairs that amount to (a) 100 million words, and (b) 10 million words. The quality of the resulting subsets is determined by the quality of a statistical and a neural Machine Translation (MT) systems trained on the selected data. The quality of the translation systems is measured computing the BLEU score on the (a) official WMT 2018 news translation test set and (b) another undisclosed test set.

The organizers make explicit that the task addresses the challenge of *data quality* and *not domain-relatedness* of the data for a particular use case. Hence, they discourage participants from sub-sampling the corpus for relevance to the news domain despite being one of the evaluation test sets. Organizers thus place more emphasis on the second undisclosed test set, although they report both scores.

The provided raw parallel corpus is the outcome of a processing pipeline that aimed for high recall at the cost of precision, which makes it extremely noisy. The corpus exhibits noise of all kinds (wrong language in source and target, sentence pairs that are not translations of each other, bad language, incomplete or bad translations, etc.).

We address this problem under the framework of quality estimation (QE) (Blatz et al., 2004). QE aims at assessing MT quality in the absence of reference translation, based on the features extracted from the source sentence and from the MT output. We consider parallel corpus filtering as a QE task where the goal is to estimate to what extent a pair of sentences in two languages correspond and, therefore, can be considered as translations of each other.

The rest of this paper is organized as follows. First, we describe our submission. Next, we present our experiments and the results of the shared task. Finally, we close the paper with the conclusions and some ideas for future work.

2 Corpus filtering as QE task

We frame the corpus filtering task within the QE framework. Given a pair of sentences $(\mathbf{s}, \mathbf{t})$, we first compute a set of features indicating to what extent the sentences correspond to each other. Then, these features are used to predict a binary score indicating if the sentences in the pair can be considered translations of each other.

In order to make the training process effective, any binary classification model needs to use both positive and negative examples. In our context positive examples are pairs of original and translated sentences, whereas negative examples are sentence pairs that cannot be considered transla-

*Marina Fomicheva worked at Webinterpret at the time of preparation of this submission.

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

[2] https://paracrawl.eu/

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 890–894

https://doi.org/10.18653/v1/W18-6477

tions of each other. Positive examples can be easily obtained from clean parallel corpora, and, while there is no explicit corpus with negative examples, these can be generated on demand.

We use the confidence score from our binary classifier as the final score for our submission to the shared task. As described in the previous Section, based on this score, the sentence pairs in the original noisy corpus provided by the organizers will be sorted and then the first N pairs will be selected and used to train the MT systems.

Note that this approach may be sub-optimal since it considers each individual pair of sentences in isolation from the rest. In exchange for this, we end up with a much more efficient method, linear in the size of the noisy data.

Next, we describe in detail the features we used for our submission (Sec. 2.1), the process we followed for generating negative examples (Sec. 2.2) and the classification model we chose (Sec. 2.3).

2.1 Features

We use a rich variety of features intended to capture what it means to be an adequate training pair of sentences. For simplicity, we split them into three categories.

Adequacy These features measure how much of the meaning of the original is expressed in the translation and vice versa. We use probabilistic lexicons with different formulations of word alignment to estimate the extent to which the words in the original and translated sentences correspond to each other.

- *Average Max lexical probability (2 f.)*: originally proposed by (Ueffing and Ney, 2005) for word-level QE. It measures the average maximum probability of translation for each word in the sentence. We apply it in both source-to-target and target-to-source directions. Formally, source-to-target is given by:

$$\frac{1}{n}\sum_{1}^{n}\max_{j=0}^{m} P(t_i \mid s_j)$$

  where the source word $\mathbf{s} = s_1 \dots s_m$ has m words, the target sentence $\mathbf{t} = t_1 \dots t_n$ has n words and the word s_0 indicates the NULL word (Brown et al., 1993). For target-to-source, source and target words swap their roles.

- *Cross-entropy (2 f.)*: proposed by (Xu and Koehn, 2017), it measures a "distance" between the sentence pairs based on a bag-of-words translation model. Specifically, the "distance" is measured as the cross-entropy between the bag-of-words of the actual sentence and the bag-of-words estimated from the other sentence in the pair via the probabilistic lexicon. We apply it in both source-to-target and target-to-source directions.

Fluency This type of features aim at capturing if the sentences are well-formed grammatically, contain correct spellings, adhere to common use of terms, titles and names, are intuitively acceptable and can be sensibly interpreted by a native speaker. We use two different features, both based on language models:

- *Language model score (2 f.)*: given language models for the source and target languages, we use as features the log probability of each sentence in the pair computed with the corresponding model.

- *Perplexity (2 f.)*: is measured as the inverse probability of the sentence normalized by its number of words. Again, we apply it to both source and target sentences in the pair.

Shape features These features can be seen as an extension of adequacy since they measure the mismatch between the frequency of different tokens between the two sentences in the pair; these features are quite commonly used in the QE literature, (Specia et al., 2015) *inter alia*.

- *Counts (8 f.)*: count of words, numbers, alphanumeric tokens, and punctuation in both source and target sentences.

- *Jaccard index (4 f.)*: metric that measures the similarity and diversity of the sets of tokens between the source and target sentences. Formally it is defined as:

$$\frac{|\, A \cap B \,|}{|\, A \cup B \,|}$$

  where A and B are the set of tokens of the source and target sentences respectively. We apply it to words, numbers, alphanumeric tokens and punctuation.

- *Counts difference (16 f.)*: we compute four metrics from the counts of tokens: the ratio in both directions, the absolute difference, and the absolute difference normalized by the maximum number of tokens of both sentences. Each of these metrics is applied to four different types of tokens: words, numbers, alphanumeric tokens and punctuation.

- *Specific punctuation (12 f.)* same as the previous features, but in this case we only compute the absolute difference and the normalized difference for specific punctuation tokens: dot (.), comma (,), colon (:), semicolon (;), exclamation mark (!), and question mark (?).

2.2 Training regime

An important consideration for this task is how to obtain suitable examples to train the classification model. Positive examples are easy to obtain since any clean parallel corpus provide us with plenty of them. Negative examples, however, are not readily available -there exist no collection of "wrong" sentence pairs. Fortunately, they can be easily generated on demand. We mostly followed the approach described in (Xu and Koehn, 2017), perturbing one or both of the sentences in a pair to create a new synthetic pair that by construction constitutes a negative example.

We apply three different perturbation operations when generating negative pairs:

- *Swap*: exchange source and target sentences.

- *Copy*: two copies of the same string. We apply it to both source and target strings.

- *Randomization*: replace the source or target sentence by a random sentence from the same side of the corpus.

As can be seen from above, we focus on the perturbation operations that mess with the correct alignment between the sentences. Thus, we aim at identifying correctly aligned sentence pairs. A complementary approach would be to aim at detecting the actual "quality" of the sentence pair, or, in other words, how valuable a sentence pair is when used for training MT systems. However, this is left for future developments.

2.3 Classification model

We did some initial experiments testing the performance of different classifiers on the task of distinguishing between actual original-translation sentence pairs and the synthetically generated negative examples (see Sec. 3.2 for details on the data we used). Gradient boosting algorithm (Friedman, 2002) obtained the highest accuracy and, therefore, we used it for our final submission.

Gradient boosting (Gra) is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. Similar to other boosting methods, it builds the models in a stage-wise fashion and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

3 Submission

Next, we describe the tools and the data we exploited for feature extraction, the data used to train the classifier, and the results of our participation in the shared task.

3.1 Feature Extraction

We need to generate two types of models to extract our features: probabilistic lexicons and language models. We used the probabilistic lexicons that can be obtained as a sub product of the training of full statistical models. In particular, we used Moses (Koehn et al., 2007) with its default configuration with the News Commentary V13 parallel corpus as provided for the News translation shared task. We used the same corpora to train the language models. For this, we used Kenlm (Heafield et al., 2013) and estimated models of order 5.

3.2 Training the classifier

We also used News Commentary V13 parallel corpus for training the classifier. We generated as many negative examples as positive sentence pairs in the corpus for a total of almost 600k data points. The negative examples were evenly distributed among the three perturbation operations described in the previous section. We used the implementation of gradient boosting classifier from the scikit-learn library[3] to train our model. The model was then applied to each sentence pair in the noisy Paracrawl corpus from the shared task.

[3] http://scikit-learn.org/stable/index.html

We used the probability of the positive class as predicted by the classifier as the final scores in our submission.

We also conducted some initial experiments using the Common Crawl corpus, under the rationale that it would be closer to the domain of the noisy data from the Paracrawl corpus. However, Common Crawl data has quite a large number of misaligned sentences. To handle this issue we implemented an iterative training process which comprises the following steps: a) train the model using all available data as positive class and synthetically generated data as negative class (see Sec 2.2); b) use the trained model to clean the available data eliminating the sentence pairs assigned to the negative class with a very high probability; c) use the cleaned data to train a new model; d) repeat until no more sentence pairs can be eliminated with a given threshold. An advantage of this approach is that it allows to be less dependent on the quality of the initial training data. However, we had to stop exploring this direction due to time constraints.

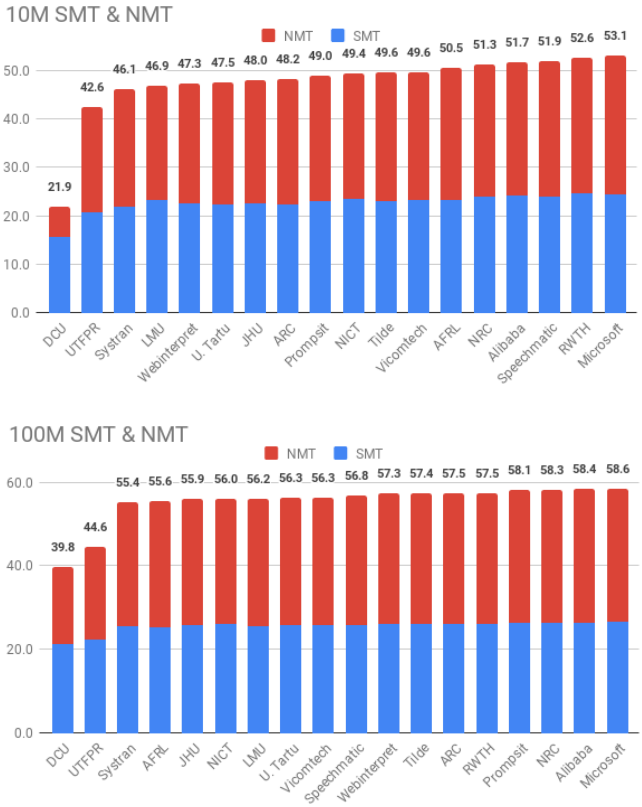

Figure 1: Best submission of each participant institution. We display BLEU [%] results stacked for SMT (blue) and NMT (red).

3.3 Evaluation and results

Participants in the shared task have to submit a file with quality scores, one per line, corresponding to the sentence pairs on the 1 billion word German-English Paracrawl corpus. Scores do not have to be meaningful, except that higher scores indicate better quality. The performance of the submissions is evaluated by sub-sampling 10 million and 100 million word corpora based on these scores, training statistical (Koehn et al., 2007) and neural (Junczys-Dowmunt et al., 2018) MT systems with these corpora, and assessing translation quality on six blind test sets[4] using the BLEU (Papineni et al., 2002) score.

Figure 1 displays the score of the best submission of each individual participant institution. The top plot shows the results for the 10 million token sub-sampled corpus, and the bottom plot shows the results for the 100 million token corpus. Scores are the aggregation of the BLEU scores of the statistical and neural systems averaged over the six blind test sets.

One first observation we can make is that (almost) all scores are quite close to each other with little variation between them; particularly in the 100 million condition. Also, the scores for the statistical and neural systems tend to follow the same pattern. We do not have confidence intervals available which makes difficult to interpret the observed differences between systems. Still, in the case of 100 million tokens sub-sampling, it seems quite clear that all the systems except for the DCU and UTFPR submissions are of the same quality. There is only a 5% relative improvement between the last system of this group and the best submission to the task. Scores are a bit more spread out in the 10 million tokens sub-sampling. This indicates that 100 million sample neutralizes the differences between the data cleaning methods and allows (almost) all systems to reach a theoretical maximum.

Our submission (Webinterpret) scored 22.5 for statistical and 24.8 for neural MT systems on the 10 million tokens sub-sampling, in comparison to the corresponding scores of 24.5 and 28.6 achieved by the best submission. For the 100 million condition, we scored 26.1 and 31.2, in comparison to the best system with the respective scores of 26.5 and 32.1.

[4]Tests: newstest 2018, iwslt 2017, Acquis, EMEA, Global Voices, and KDE.

4 Conclusions

We have presented our submission to the WMT18 shared task on parallel corpus filtering. We frame the task as a QE problem, where we estimate how well two sentences correspond to each other to be part of a training sample for MT models. Our approach is computationally light, takes advantage of well-known methods used for QE, and exploits a general training regime that allows to customize it by defining under demand samples of negative examples.[5]

There are several directions that can be explored to extend this approach:

- Use a neural model to automatically estimate the features relevant for the system instead of hand-crafting them.
- Extend the training regime with new perturbation operations, in particular those that degrade the quality of the pair so it is less valuable as training data for MT.
- Implement an iterative training procedure where steps of model training and data cleaning are repeated over the available training data until convergence. This will make training more robust and less dependent on the quality of available training data.

Acknowledgments

Work funded by WebInterpret.

References

Gradient boosting. https://en.wikipedia.org/wiki/Gradient_boosting. Accessed: 2018-08-15.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.

[5]The code developed to prepare our submission is available at https://github.com/mfomicheva/parallel_data_cleaning.

Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378.

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.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007, System Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

Nicola Ueffing and Hermann Ney. 2005. Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the European Association for Machine Translation conference*, pages 262–270.

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.