SYSTRAN Participation to the WMT2018 Shared Task on Parallel Corpus Filtering

MinhQuang Pham, Josep Crego, Jean Senellart
SYSTRAN / 5 rue Feydeau, Paris (France)
FirstName.LastName@systrangroup.com

Abstract

This paper describes the participation of SYSTRAN to the shared task on parallel corpus filtering at the Third Conference on Machine Translation (WMT 2018). We participate for the first time using a neural sentence similarity classifier which aims at predicting the relatedness of sentence pairs in a multilingual context. The paper describes the main characteristics of our approach and discusses the results obtained on the data sets published for the shared task.

1 Introduction

Corpus-based approaches to machine translation rely on the availability and quality of parallel corpora. In the case of neural machine translation, a large neural network is trained to maximise the translation performance on a given parallel corpus. Therefore, the quality of an MT engine is heavily dependent upon the amount and quality of the training parallel sentences. Such resource is not naturally existing, and because of the process necessary to compile a parallel corpus, it may contain multiple sentence pairs that are often not as parallel as one might assume.

The primary objective of our approach is to assess whether we are able to identify parallel sentences using a flexible method that relies on deep learning architectures. Thus, eliminating the need for any domain specific feature engineering. We evaluate the feasibility of a model learnt over the same noisy data that must be cleaned. Using as few external tools as possible.

Hence, we tackle the filtering problem by means of a neural sentence similarity network, which aims at predicting the relatedness of sentence pairs. Pairs are selected according to their similarity score, thus filtering those sentences which are less likely to be translations of each other. The rest of this paper is organised as follows. After describing the filtering task we outline our similarity classifier. Next, we present experiments and results of the shared task. Finally, we draw some conclusions.

2 Task description

In the context of the third conference on machine translation (WMT18), the parallel corpus filtering shared task[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 organisers provide a very noisy 1 billion word (English token count) German-English corpus crawled from the web as part of the Paracrawl project[2]. Participants must subselect 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 statstical and a neural machine translation system trained on this data. The quality of the machine translation system is measured by BLEU score on the (a) official WMT 2018 news translation test and (b) another undisclosed test set.

The organisers 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 subsampling the corpus for relevance to the news domain despite being one of the evaluation test sets. Organisers 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, bad language,

[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 947–951
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-6486

incomplete or bad translations, etc.).

3 Neural Similarity Classifier

Our network architecture is very much inspired by the work on Word Alignment in (Legrand et al., 2016). Figure 1 illustrates the network. In the following, we consider a source-target sentence pair (s,t) with $s=(s_1,...,s_I)$ and $t=(t_1,...,t_J)$.

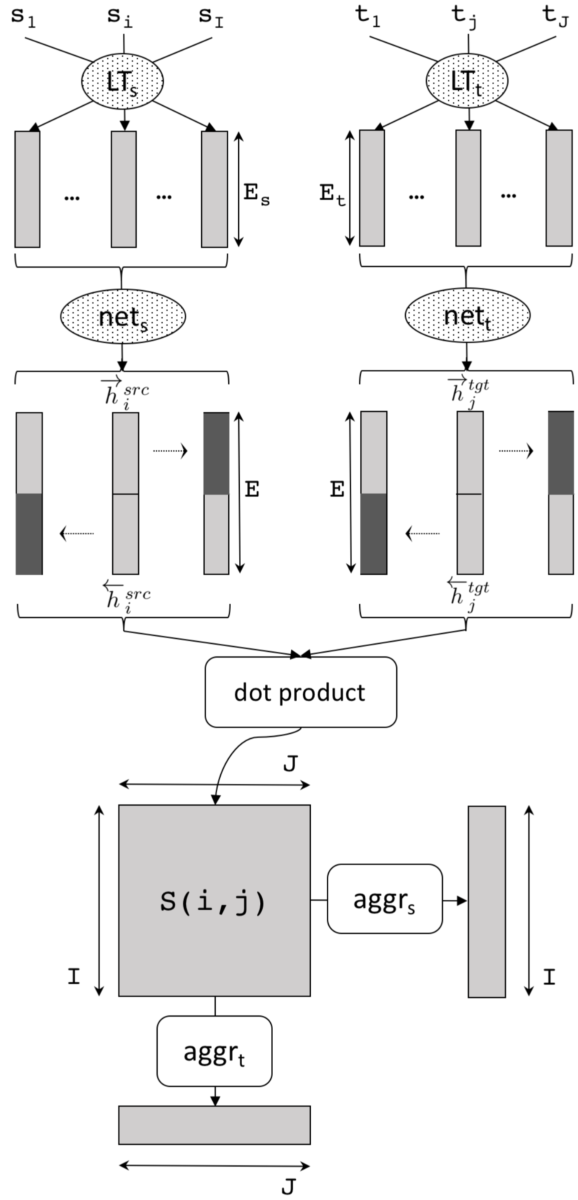
s_1
s_i
s_I
t_1
t_j
t_J
LT_s
LT_t
E_s
E_t
net_s
net_t
$\overrightarrow{h}_i^{src}$
$\overleftarrow{h}_i^{src}$
$\overrightarrow{h}_j^{tgt}$
$\overleftarrow{h}_j^{tgt}$
E
E
dot product
J
S(i,j)
$aggr_s$
$aggr_t$
I
I
J

Figure 1: Illustration of the model. The network is composed of source and target word embedding lookup tables (LT_s and LT_t) and two identical subnetworks (net_s and net_t) that compute in context representations of source (s_i) and target words (t_j).

The model is composed of 2 Bi-directional LSTM subnetworks, net_s and net_t, which respectively encode source and target sentences. Since both net_s and net_t take the same form we describe only the source architecture. The source-sentence Bi-LSTM network outputs forward and backward hidden states, $\overrightarrow{h}_i^{src}$ and $\overleftarrow{h}_i^{src}$, which are then concatenated into a single vector encoding the i^{th} word of the source sentence, $h_i^{src}=[\overrightarrow{h}_i^{src};\overleftarrow{h}_i^{src}]$. In addition, the last forward/backward hidden states (outlined using dark grey in Figure 1) are also concatenated into a single vector to represent whole sentences $h_{src}=[\overrightarrow{h}_I^{src};\overleftarrow{h}_1^{src}]$. At this point a measure of similarity between sentences can be obtained by cosine similarity:

$$sim(h_{src},h_{tgt})=\frac{h_{src}\cdot h_{tgt}}{||h_{src}||*||h_{tgt}||} \tag{1}$$

where two vectors (embeddings) with the same orientation have a cosine similarity of 1, while two vectors with opposed orientation have a similarity of -1, independent of their magnitude.

Similar to (Legrand et al., 2016) our model extracts context information from source and target sentences and then computes simple dot-products to estimate word alignments. The objective function is computed at the level of words. To enable unsupervised training, we use an aggregation operation that summarizes the alignment scores for a given target word. A soft-margin objective increases scores for true target words while decreasing scores for target words that are not present. The aggregation function combines the scores of all source (or target) words for a particular target (or source) word and promotes source words which are likely to be aligned with a given target word according to the knowledge the model has learned so far. Alignment scores $S(i,j)$ are given by the dot-product $S(i,j)=h_i^{src}\cdot h_j^{tgt}$, while aggregation functions are defined as:

$$\begin{aligned} aggr_s(i,S)&=\frac{1}{r}\,log\left(\sum_{j=1}^{J}e^{r*S(i,j)}\right)\\ aggr_t(j,S)&=\frac{1}{r}\,log\left(\sum_{i=1}^{I}e^{r*S(i,j)}\right)\end{aligned} \tag{2}$$

The loss function is defined as:

$$\begin{aligned}\mathcal{L}(src,tgt)=&\\ &\sum_{i=1}^{I}log\left(1+e^{aggr_s(i,S)*\mathcal{Y}_i^{src}}\right)+\\ &+\sum_{j=1}^{J}log\left(1+e^{aggr_t(j,S)*\mathcal{Y}_j^{tgt}}\right)\end{aligned} \tag{3}$$

where $\mathcal{Y}_i^{src}$ and $\mathcal{Y}_j^{tgt}$ are vectors with reference labels containing -1 when the word is present in the translated sentence, and $+1$ for divergent (unpaired) words.

Further details on the network can be found in (Pham et al., 2018).

3.1 Training with Negative Examples

Training is performed by minimising Equation 3, for which examples with annotations for source $\mathcal{Y}_i^{src}$ and target $\mathcal{Y}_j^{tgt}$ words are needed.

As positive examples we use **paired** sentences of the parallel corpus. In this case, all words in both sentences are labelled as parallel, $\mathcal{Y}_i^{src} = -1$ and $\mathcal{Y}_j^{tgt} = -1$.

As negative examples we use random **unpaired** sentences. In this case, all words are labelled as divergent, $\mathcal{Y}_i^{src} = +1$ and $\mathcal{Y}_j^{tgt} = +1$.

In order to be able to predict less obvious divergences we **replace** random sequences of words on either side of the sentence pair by a sequence of words with the same part-of-speeches. The rationale behind this method is to keep the new sentences as grammatical as possible. Otherwise, to predict divergence the network can learn to detect non-grammatical sentences. Words that are not replaced are considered parallel (-1) while those replaced are assigned the divergent label ($+1$). Words aligned to some replaced words are also assigned the divergent label ($+1$).

Finally, motivated by sentence segmentation errors observed in many corpora, we also build negative examples by **inserting** a second sentence at the beginning (or end) of the source (or target) sentence pair. Words in the original sentence pair are assigned the parallel label (-1) while the new words inserted are considered divergent ($+1$).

In order to avoid that negative examples are easily predicted just by looking at the difference in length of training sentences we constraint all negative examples to have a difference in length not exceeding 2.0. Very short sentences, of up to 4 words, are accepted if the length ratio does not exceeds 3.0.

4 Experiments

4.1 Neural Similarity Classifier

All data is preprocessed with `OpenNMT`[3], performing minimal tokenisation, basically splitting-off punctuation. After tokenisation, the $50,000$ most frequent words of each language are used as vocabulary. Each out-of-vocabulary word is mapped to a special UNK token. Word embeddings (LT_s and LT_t) are initialised using `fastText`[4], further aligned by means of `MUSE`[5] following the unsupervised method detailed in (Lample et al., 2018). Size of embeddings is $E_s = E_t = 256$ cells. Both Bi-LSTM use 256-dimensional hidden representations ($E = 512$). We use $r = 1.0$. Optimisation of the parameters is done using the stochastic gradient descent method along with gradient clipping (rescaling gradients whose norm exceeds a threshold) to avoid the exploding gradients problem (Pascanu et al., 2013). For each epoch we randomly select 1 million sentence pairs that we place in batches of 32 examples. Word alignments and English part-of-speeches used to build negative examples were performed by `fast_align`[6] and `FreeLing`[7] respectively. We run 10 epochs and start decaying at each epoch by 0.8 when score on validation set increases. Similarity is always computed following equation 1.

4.2 Simple Filtering

The Corpora of the shared task contains 1 billion word (English token count) German-English corpus crawled from the web as part of the Paracrawl project. Observing that many sentence pairs could be easily filtered out by simple rules imposed on length and language, we use a very simple filter which removes 80% of the sentence pairs. Our basic filterig consists of:

- Language Identification on source and target sentences,
- removing pairs whose source-target or target-sources length ratio is higher than 6,
- removing pairs whose source or targets length is higher than 100.

After this simple filtering, our corpus is reduced to 22 million sentence pairs.

5 Results

Participants in the shared task have to submit a file with quality scores, one per line, corresponding to

[3] http://opennmt.net

[4] https://github.com/facebookresearch/fastText

[5] https://github.com/facebookresearch/MUSE

[6] https://github.com/clab/fast_align

[7] https://github.com/TALP-UPC/FreeLing.git

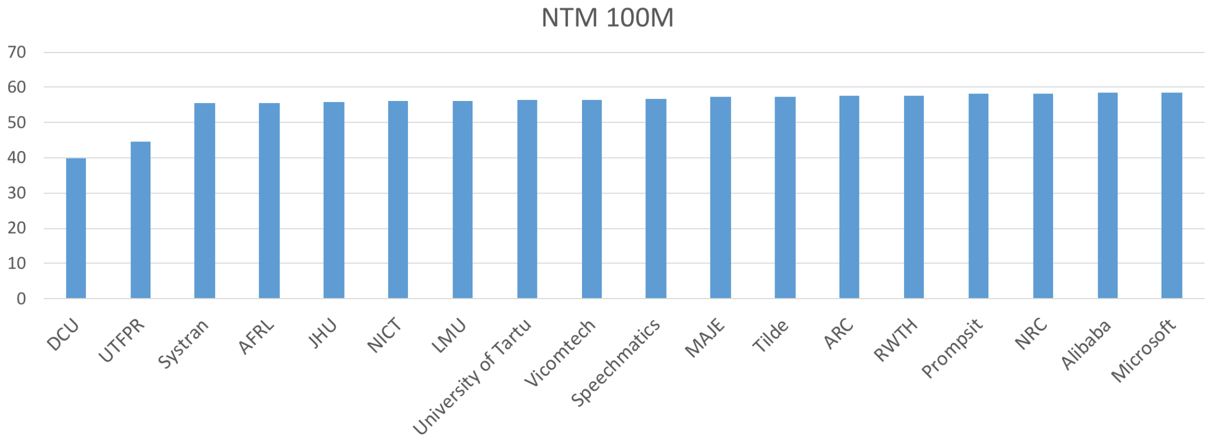

Figure 2: BLEU score of the best submission of each participant measured for the neural MT system trained with 100M tokens. Score is averaged over the six blind test sets.

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[8] using the BLEU (Papineni et al., 2002) score.

Figure 2 displays the score of the best submission of each individual participant corresponding to the 100 million tokens corpus using the neural MT system. BLEU score is averaged over the six blind test sets.

As it can be seen, very similar results were obtained by most of the participants. Accuracy results fall within a margin of 3 points BLEU for the first 16 classified.

6 Conclusions

We have presented our submission to the WMT18 shared task on parallel corpus filtering. We participated for the first time using a neural sentence similarity classifier that predicts relatedness between sentence pairs in a multilingual context. The primary objective of our approach was to assess whether we were able to identify parallel sentences using a flexible method that relies on deep neural networks. Thus, eliminating the need for any domain specific feature engineering and using as few external tools as possible. We succeeded in our objective as we built a very simple network that was able to filter out divergent sentence pairs. Only assisted by a very simple filtering technique using rules based on length and language identification.

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

References

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*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 66–73. Association for Computational Linguistics.

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.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1310–III–1318. JMLR.org.

Minh Quang Pham, Josep Crego, Jean Senellart, and François Yvon. 2018. Fixing translation divergences in parallel corpora for neural mt. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*.