BUCC 2017 Shared Task: a First Attempt Toward a Deep Learning Framework for Identifying Parallel Sentences in Comparable Corpora

Francis Grégoire
RALI - DIRO
Université de Montréal
gregoifr@iro.umontreal.ca

Philippe Langlais
RALI - DIRO
Université de Montréal
felipe@iro.umontreal.ca

Abstract

This paper describes our participation in BUCC 2017 shared task: identifying parallel sentences in comparable corpora. Our goal is to leverage continuous vector representations and distributional semantics with a minimal use of external preprocessing and post-processing tools. We report experiments that were conducted after transmitting our results.

1 Introduction

Traditional approaches for parallel sentence identification from comparable corpora rely on machine learning models with the use of features measured by statistical machine translation (SMT) systems. Munteanu and Marcu (2005) present how to extract parallel sentences from newspaper articles using general and alignment features to train a binary maximum entropy classifier. Abdul-Rauf and Schwenk (2009) use an SMT-based system on comparable corpora to translate the source language side to detect corresponding parallel sentences on the target language side. While continuous vector representations of words and sentences estimated by neural language models and neural networks (Bengio et al., 2003; Collobert and Weston, 2008) have been successfully applied to a variety of natural language processing tasks, ranging from handwriting generation (Graves, 2013) to machine translation (Sutskever et al., 2014), few efforts have been devoted to parallel sentence identification. Ferrero et al. (2017) successfully use word embeddings for cross-language plagiarism detection, which can be considered a similar task to ours.

The primary objective of our proposed approach is to assess whether we are able to identify parallel sentences using a scalable and flexible method by relying on recent advances in neural language modeling and deep learning architectures to eliminate the need for any domain specific feature engineering. We want to evaluate the feasibility of a model learnt from distributional semantics alone in a "pure" setting by using as few external tools as possible. Our approach can be considered as a first attempt to accomplish the proposed task using a deep learning framework. Our aim is not to attain state-of-the-art performance, but to open interesting directions to enable researchers to advance research with this important task.

In fact, in the following sections we report the approach of our two-day effort to participate on this year's shared task. Due to the short limit of time, we used models pretrained on a standard parallel corpus. The details of our approach will be described elsewhere. In this paper we report what we learned so far and few experiments that were conducted after submitting our results.

2 Approach

2.1 Model

Our model architecture is a bidirectional recurrent neural network with gated recurrent units (Bi-GRU) (Cho et al., 2014) built for both the source language and target language sentences. The Bi-GRU encodes each sentence in both directions to generate two continuous vector representation of the sentence, $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$. The forward network processes the input sentence and updates its recurrent state from the first token until the last one. The backward network processes the input sentence in reverse direction. The concatenation of the final recurrent state in both directions is the sentence representation $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i ; \overleftarrow{\mathbf{h}}_i]$.

Once both source and target sentence representations have been encoded, $\mathbf{h}_i^S$ and $\mathbf{h}_i^T$, we measure the semantic similarity between the two sentences to estimate the probability that they are par-

Proceedings of the 10th Workshop on Building and Using Comparable Corpora, pages 46–50,
Vancouver, Canada, August 3, 2017.

allel:

$$p(y_i = 1|\mathbf{h}_i^S, \mathbf{h}_i^T) = \sigma(\mathbf{h}_i^{S^\top} \mathbf{M} \mathbf{h}_i^T + b) \quad (1)$$

where $\mathbf{M}$ and b are model parameters, σ is the sigmoid function, $y_i = 1$ if the sentence pair is parallel and $y_i = 0$ otherwise. The model outputs a positive instance if a sentence pair gets a probability score higher than a decision threshold λ:

$$\hat{y}_i = \begin{cases} 1 & \text{if } p(y_i = 1|\mathbf{h}_i^S, \mathbf{h}_i^T) \geq \lambda \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

We train our model by minimizing the cross entropy of our labeled sentence pairs $(\mathbf{x}_i^S, \mathbf{x}_i^T, y_i)$ that we feed in our BiGRU, where $\mathbf{x}_i^S = (\mathbf{w}_{i,1}^S, \ldots, \mathbf{w}_{i,|\mathbf{x}_i^S|}^S)$ is a source sentence and $\mathbf{x}_i^T = (\mathbf{w}_{i,1}^T, \ldots, \mathbf{w}_{i,|\mathbf{x}_i^T|}^T)$ is a target sentence. $\mathbf{w}_{i,t}^S$ and $\mathbf{w}_{i,t}^T$ are the continuous word representation (word embeddings) of the words in the source and target sentences, respectively. We use a parallel corpus made of N parallel sentence pairs $(\mathbf{x}_i^S, \mathbf{x}_i^T)$, for $i \in \{1, \ldots, N\}$. For every pair of parallel sentences we add negative examples by randomly selecting 5 negative sentence pairs $(\mathbf{x}_i^S, \mathbf{x}_j^T)$, for $j \neq i$.

2.2 Candidate filtering

Very often, identifying parallel sentences in comparable corpora is an extremely unbalanced classification task because the number of sentence pairs to be examined is potentially the Cartesian product between sentence pairs in the corpora. This is not an issue for small comparable corpora, e.g. two Wikipedia articles. However, in our case we are given two monolingual corpora of approximately 370,000 and 270,000 sentences, for a potential of $9.99e10$ pairs of sentences to evaluate. To reduce the size and the noise of the candidate sentence pairs, traditional approaches apply candidate filters such as sentence length ratio, bilingual dictionary word overlap, word alignment conditions from SMT and information retrieval systems (Resnik and Smith, 2003; Munteanu and Marcu, 2005; Abdul-Rauf and Schwenk, 2009).

Following our idea to evaluate the feasibility of an approach using only distributional representations, for each sentence we learned its continuous vector representation and created our set of candidate sentence pairs by using the n-best cosine similarity score between each source sentence and every target sentences. Since we are working with vector representations, doing the Cartesian product is tractable. To estimate the vector representation of each sentence, $\mathbf{s}_i^S$ and $\mathbf{s}_i^T$, we employ a distributional bag-of-words approach where word embeddings have been mapped to a shared vector space, i.e. cross-lingual word embeddings (Gouws et al., 2015). The sentence representation is the normalized sum over the word embeddings present in it:

$$\mathbf{s}_i^S = \frac{\sum_t \mathbf{w}_{i,t}^S}{|\sum_t \mathbf{w}_{i,t}^S|_2} \quad (3)$$

3 Experiments

In this section we present experiments that were conducted after the submission of our results. First, we describe the resources used to perform the shared task, the training settings and the evaluation metrics.

3.1 Dataset

We only participated to the fr-en language pair, making use only of our models pretrained on the Europarl v7 English to French parallel corpus from WMT'15[1]. To create our training set, 500K parallel sentence pairs are randomly selected. The vocabulary sizes range between 103K to 119K for English and 126K to 140K for French depending on the digit preprocessing method (see Section 4.2). We tokenize the dataset with the scripts from Moses[2] and all words are lowercased. Empty sentence pairs are removed.

For the shared task, we replaced all digits with 0 (e.g. 1982→0000).

3.2 Training settings

We use TensorFlow[3] (Abadi et al., 2016) to train our models. The dimension of the BiGRU recurrent state is 200 in each direction with word embeddings of dimension 300. We train our models using a mini-batch size of 128 and Adam optimizer (Kingma and Ba, 2014) with a learning rate of $2e$-4 for a total of 10 epochs. We augment our training examples with new negative examples by sampling 5 negative sentence pairs for each parallel sentence pair. We apply gradient clipping to a value of 5.

[1] http://www.statmt.org/wmt15/translation-task.html
[2] https://github.com/moses-smt/mosesdecoder
[3] https://github.com/tensorflow/tensorflow

Cross-lingual word embeddings used for candidate filtering are trained for 10 epochs with the BilBOWA toolkit[4] (Gouws et al., 2015) by using the 2M sentence pairs of Europarl both as monolingual and parallel training data. We use the default parameters with word embedding dimension of 300 and a subsampling rate of value $1e$-4.

3.3 Evaluation metrics

For the evaluation of our models we present the precision, recall and F_1 scores as mentioned on the shared task website[5].

3.4 Details

Candidate filtering To obtain our candidate sentence pairs we apply our n-best cosine similarity filter as described in Section 2.2. For the shared task we applied the filter on the shared task test set, but for the experiment reported here we applied it to the shared task training set. A low n value will result in fewer candidate sentence pairs to evaluate and can be detrimental to the recall score. On the other hand, a high value can lead to an undesirable number of candidate sentence pairs and potentially a lower precision score due to a higher number of false positive examples. We evaluate the loss of parallel sentences in the training set with respect to the value of n.

Digits preprocessing While observing our system's outputs during the shared task we noticed a substantial number of false positive examples due to digits being replaced to 0. Consequently, we analyze our approach by measuring the impact of the following preprocessing choices for training and evaluating our model: (i) keep digits; (ii) replace digits to 0; (iii) remove digits. For this experiment we create validation sets by using the 9,086 pairs of parallel sentences from the shared task training set and adding 50M randomly selected negative sentence pairs. Hence, 0.018% of the sentence pairs are considered parallel.

Model evaluation Whereas in the previous experiment we report results on experimental noisy validation sets matching the optimal decision threshold λ, in this experiment we evaluate our approach in a real inference setting on the shared task training set using a 40-best cosine similarity filter and a fixed λ value of 0.99.

[4]https://github.com/gouwsmeister/bilbowa

[5]https://comparable.limsi.fr/bucc2017/bucc2017-task.html

n	found	found (%)	Δ (%)	pairs	Δ (%)
1	6,891	75.84		369,810	
10	7,824	86.11	13.54	3,698,100	900.00
20	8,020	88.27	2.51	7,396,200	100.00
30	8,114	89.30	1.17	11,094,300	50.00
40	8,190	90.14	0.94	14,792,400	33.33
50	8,243	90.72	0.65	18,490,500	25.00
60	8,279	91.12	0.44	22,188,600	20.00
70	8,311	91.47	0.39	25,886,700	16.67
80	8,340	91.79	0.35	29,584,800	14.29
90	8,370	92.12	0.36	33,282,900	12.5
100	8,388	92.32	0.22	36,981,000	11.11
1000	8,752	96.32	4.34	369,810,000	900.00

Table 1: Parallel sentences found from the n-best cosine similarity filter. The Δ columns are the percentage increase in number of parallel sentences found and candidate sentence pairs.

4 Results

4.1 Candidate filtering

In Table 1 we present the information regarding the number of parallel sentences and number of candidate sentence pairs obtained by augmenting the value of n for our candidate filtering method described in 2.2. We see that our cosine similarity filter is able to capture most of the parallel sentence pairs, even for low n values. For $n = 1$, we are surprised to see that such a simple approach using pretrained cross-lingual word embeddings on the Europarl dataset is able to capture 75.84% of the parallel sentence pairs found in the shared task training set. By looking at the Δ columns, we anticipate that there is a precision-recall trade-off by increasing n. For example, if we increase from a 30-best to a 40-best filter, we increase the recall score at most by 0.94%. On the other hand, we augment the number of candidate sentence pairs to evaluate by 33.33%, increasing the risk of false positive examples and a lower precision score. For the shared task we naively used $n = 100$.

4.2 Digits preprocessing

In this experiment we trained two new models on Europarl; by keeping or removing digits. In Table 2 we report the precision, recall and F_1 scores for our three different approaches evaluated on validation sets made of the 9,086 parallel sentences and 50M randomly selected sentences from the shared task training set. The precision-recall curves with respect to different decision threshold values λ are reported in Figure 1. We observe that

Model	Precision (%)	Recall (%)	F_1 (%)
Digits	83.25	65.86	73.54
Digits to 0	71.41	56.38	63.01
No digits	79.65	63.86	70.89

Table 2: Performance of our models trained on Europarl with three different digits preprocessing method and evaluated on our validation sets made from the shared task training set.

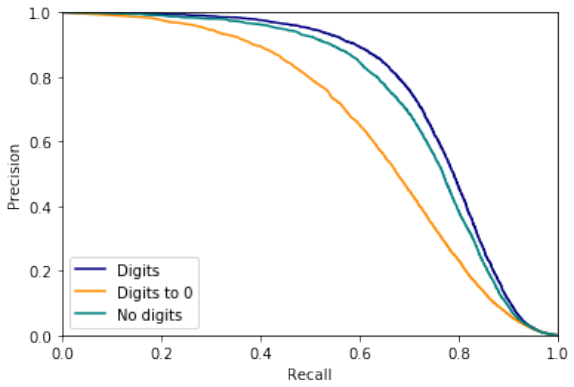

Figure 1: Precision-Recall curve for our models trained on Europarl with three different digits preprocessing method and evaluated on our validation sets made from the shared task training set.

naively replacing the digits to 0, as we did for the shared task, is actually the worst option.

4.3 Model evaluation

Equipped with a filter that seems to work well and a better model trained on a parallel corpus with digits, we expect to obtain a performance in the range of those presented in Section 4.2. Unfortunately for us, it is not the case. Table 3 presents the results we obtained by using our model trained on Europarl with digits, using the 40-best list and $\lambda = 0.99$. One may wonder what happened to our surprisingly low precision score. The problem arises from a combination of how the model is trained on negative examples and how we filtered our candidate sentence pairs. Since our model outputs a positive instance for two sentences sharing an high level of semantic similarity, by filtering the 40 nearest target sentences for each source sentence, we created a pool of candidate sentence pairs that our model outputted as positive most of the time. That being said, those sentence pairs still exist in the Cartesian product of the training set. Thus, the proposed training procedure adding negative examples randomly selected from the training set is definitely not adequate and needs to be replaced by a more effective procedure. For future work, instead of random sampling, we propose to apply the n-best cosine similarity filter on our model's training set in a way to select negative examples from the n-best list to train it. A post-processing step could also be useful.

Decision Threshold	Precision (%)	Recall (%)	F_1 (%)
0.99	12.10	70.95	20.67

Table 3: Performance of our models trained on Europarl with digits using the 40-best cosine similarity filter.

5 Discussion

The idea toward an end-to-end sentence driven approach using deep neural networks for parallel sentence identification is compelling. However, there is much room for improvement. We presented that our initial approach learned on distributional semantics alone has weak points that need to be addressed. With its current architecture and setting, the main issue is the low precision score due to the large amount of false positive examples our system outputs, acting more as a quasi-parallel sentences extractor. The source of this issue comes from the random sampling procedure used to add negative examples to the training set. We have seen that even for a low value n, our simple distributional bag-of-words n-best filter is capable of capturing most parallel sentences found in the comparable corpora, leading to a potentially good recall score. A promising next step would be to use the same n-best filter on our training set and to select negative examples from the n-best list to train our model. We anticipate that selecting negative examples that are similar to the source sentence will allow our approach to capture finer semantic granularities and to have a better precision score. Furthermore, a model trained on negative examples of higher quality should allow us to use a lower optimal decision threshold λ, which increases the recall score.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin,

Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. pages 265–283.

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve smt performance. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '09, pages 16–23. http://dl.acm.org/citation.cfm?id=1609067.1609068.

Yoshua Bengio, Rjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JOURNAL OF MACHINE LEARNING RESEARCH* 3:1137–1155.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '08, pages 160–167. https://doi.org/10.1145/1390156.1390177.

Jérémy Ferrero, Frédéric Agnès, Laurent Besacier, and Didier Schwab. 2017. Using word embedding for cross-language plagiarism detection. *CoRR* abs/1702.03082. http://arxiv.org/abs/1702.03082.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, ICML'15, pages 748–756.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850. http://arxiv.org/abs/1308.0850.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.* 31(4):477–504. https://doi.org/10.1162/089120105775299168.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. *Comput. Linguist.* 29(3):349–380. https://doi.org/10.1162/089120103322711578.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'14, pages 3104–3112. http://dl.acm.org/citation.cfm?id=2969033.2969173.