Improving Machine Translation Results by Corpus Filtering for a Low-Resource Language

Anthony Rubin

Abstract

In this paper, we discuss the implementation of a method to reduce noise in a parallel corpus. A compositionality method based on cosine similarity as well as basic features to filter the corpus are used with a classifier to predict good sentence translation pairs within a parallel corpus. The purpose of this is to clean a noisy parallel corpus to the point that it can be used to accurately train a machine translation model. The accuracy of our resulting machine translation models was quantified using a BLEU score, and we improved results progressively by our proposed method.

1 Introduction

Machine translation, be it neural machine translation (NMT, *e.g.* Bahdanau et al. (2015)) or statistical machine translation (SMT, Brown et al. (1993)), is built upon large amounts of parallel data. A parallel corpus is a pair of texts written in different languages which are translations of each other (Gale and Church, 1993). Since multilingual publication has become more widespread, there is an increasing amount of such parallel data available. Gathering this data set can still be both costly and time consuming. One option is to crawl the web in an attempt to gather large amounts of parallel data at little cost (Resnik and Smith, 2003). The catch is that the data returned is often riddled with noise, such as URL's, bad translations, foreign languages, links to images, and more. Among all of this noise, however, is some sound parallel data. The challenge of corpus filtering is finding a balance between removing enough noisy data to allow for accurate machine translation and not removing too much data (Frénay and Verleysen, 2014).

In this paper we use a noisy filtering method using a classifier to predict a good sentence pair. Specifically we deal with a method of cleaning a noisy English-Nepali parallel corpus. Nepali is a low resource language. In short, a low resource language is one where there exists few ways to annotate the data. In the case of corpus filtering, this means there are a fewer number of tools for tokinization, part of speech tagging, parsing, etc. making it a more difficult task when trying to clean the data for a machine translation system. Our method of parallel corpus cleaning was to gather certain metrics or information about each sentence, which would indicate whether a sentence pair should be included as part of our training data or rejected. The advantage of this approach is its generality. We found that each sentence pair had features from which we could predict if it would make a good translation. These features are likely found among sentence pairs in any parallel corpus despite the language. Therefore, this same approach can be used to filter parallel data for any other language. To achieve this we propose a compositionality method based on cosine similarity (Hartung et al., 2017) as well as basic features to filter the corpus. The remainder of this paper is as follows: we describe the metrics in detail (Section §2), show results of experiments (Section §3), and draw a conclusion (Sections §4).

2 Metrics as Features

The input data for the classifier was a list of vectors of integers, with each vector containing the metric information of a given parallel sentence pair. There were eight such metrics for every sentence as described below. Some of the metrics were binary (true or false). Others were some value between 0 and 1, which would be calculated from averaging that metric value in the Nepali sentence with that metric value in the English sentence.

2.1 Basic metrics

We are using following basic metrics:

$\mathcal{M}_1$: digit and punctuation mark ratio This metric is the ratio between the total number of digit and punctuation marks in the sentence out of the total number of characters. It was important to filter out sentences with a high ratio of digits and punctuation marks, as these would most likely not make for a good translation.

$\mathcal{M}_2$: correct character count This metric represents the total number of correct alphabetic characters out of the total number of characters in each respective sentence. More specifically, for English sentences a correct alphabetic character is one that is found in the English alphabet. The same is done for the Nepali sentence with the Nepali alphabet. Naturally, a high ratio will mean a better possible translation.

$\mathcal{M}_3$: incorrect language If an English sentence contains Nepali or a Nepali sentence contains English, that metric has a score of 1. If not, the metric has a score of 0.

$\mathcal{M}_4$: language character count If an English sentence contains less than a certain percentage of English characters, or a Nepali sentence contains less than a certain percentage of Nepali characters, give that metric a score of 1. If not, the metric has a score of 0. This is similar to metric two ($\mathcal{M}_2$), except that this metric acts as a penalty for any sentence pair below a certain percentage threshold. We used 75% as a threshold.

$\mathcal{M}_5$: excessively long token length If a sentence in any language contains unusually long tokens, it would be an error from text preprocessing or tokenization. Therefore, it would not be a good candidate for a parallel sentence pair. We use a threshold $\theta > 40$ for token length: *i.e.* if a sentence in English contains several words with a length greater than 40, we give that metric a score of 1. Otherwise the metric has a score of 0.

$\mathcal{M}_7$: other web crawl noise Since this corpus was crawled from the Web, there are many sentences that are only URL's, links to images or something of the like. If a sentence contains useless metadata from the webpage such as http, www, .com, .org, .jpg, .png, or .gif, we give that metric a score of 1 to mark it as a noisy sentence. If not this metric has a value of 0.

2.2 Compositionality metric

This metric was used to calculate the cosine similarity between every pair of words in the sentence for Compositionality. Hartung et al. (2017) proposed a method to learn a compositionality function which combines individual words into a phrase representation. Their method can capture the compositional attribute meaning. Given word vectors $\mathbf{A}$ and $\mathbf{B}$, the cosine similarity is calculated as follows:

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \quad (1)$$

For each sentence pair, all possible un-ordered pairs of words in each sentence were computed. For example, if the sentence was *big brown fox*, the resulting pairs would be (*big*, *brown*) , (*brown*, *fox*), and (*fox*, *big*). The cosine similarities of each of these word pairs would be computed using word embeddings. If one of the words in the pair did not exist in the embedding model that pair would get discarded. The average compositionality value of all the ($\frac{n*(n-1)}{2}$) pairs in a sentence was found. This was done for every English sentence and every Nepali sentence. From there we were able to calculate the compositionality value for each sentence pair in the corpus.

$$\mathcal{M}_8 = \frac{\text{Compositionality value of Nepali}}{\text{Compositionality value of English}} \quad (2)$$

This compositionality metric allowed us to tackle this corpus filtering problem from a different semantic angle than the basic metrics which do not have the same level of robustness.

3 Experiments and Results

The English-Nepali parallel corpus we used was obtained by the Paracrawl project[1] as part of a shared task at the Fourth Conference on Machine Translation (WMT19). The corpus contained over 2 million sentence pairs. Also provided in the shared task was a few thousand sentences of high

[1] https://paracrawl.eu

quality English-Nepali parallel data. We implemented the basic metrics which were described as well as the compositionality metric using word embedings by fastText (Joulin et al., 2016). English embedding was made from a portion of the AFP News Corpus which has around 2M sentences with 63M tokens,[2] and the Nepali embedding from the Nepali Monolingual written corpus which has 0.9M sentences with 15M tokens.[3]

To measure our baseline, we used Moses (Koehn et al., 2007) to build a statistical machine translator and used all sentence pairs as training data by limiting sentence length to 80. Nepali sentences are preprocessed using the model trained from the Nepali POS tagged corpus. The resulting BLEU score (Papineni et al., 2002) was 2.2.

We then implemented a multilayered approach to solve this problem. Three different SVM models were built. All three SVM models used the same positive training examples from an English-Nepali parallel corpus from ELRA.[4] The same number of negative training examples for every SVM were randomly selected sentence pairs from the original noisy parallel corpora. All three of these models filtered the parallel corpora down to about 350K sentence pairs. Since the only difference between the three SVM's was the negative training examples, we found the intersection of the three SVM outputs. This allowed for an even stricter filtering of the corpus. The intersection resulted in a new parallel corpus with 280K sentence pairs. A translation model was built with this corpus which achieved a BLEU score of 2.55. A manual inspection of this new corpus revealed some persisting noisy data.

A new SVM was made, again with randomly selected negative training examples, but also with selected negative examples persisting in the new parallel corpus. This resulted in a further reduction in corpus size to about 218K and an improved BLEU score of 3.2. Although 3.2 is still a low score, there are a few possible reasons for this. First, was the corpus size. From an initial size of 2M it was reduced to 218K sentence pairs. While 218K is large enough to obtain accurate results in some instances, in general, a SMT system should have upwards of a million sentence pairs. Second, the quality of this data was extremely low. Consider the following sentence pairs selected from the beginning of the unfiltered parallel corpora as shown in Figure 1a compared to the sentence pairs selected from the beginning of the filtered corpora in Figure 1b.

[2]English Gigaword, https://catalog.ldc.upenn.edu/LDC2003T05

[3]http://catalogue.elra.info/en-us/repository/browse/ELRA-W0076/

[4]http://catalog.elra.info/en-us/repository/browse/ELRA-W0077/

	DTEST	ELRA
baseline (1.8M)	2.2	2.83
filtered (RN, 280K)	2.55	2.85
filtered (FN, 218K)	3.2	3.65
filtered (FN, 218K) + DIC	3.6	4.3
random pair (218K)	1.32	1.32

Table 1: MT results using the BLEU scores: RN = random negative examples for SVM and FN = filtered negative. DI is for the dictionary, DTEST from WMT19 for test, and ELRA from the ELRA English-Nepali 1060 sentence parallel corpus.

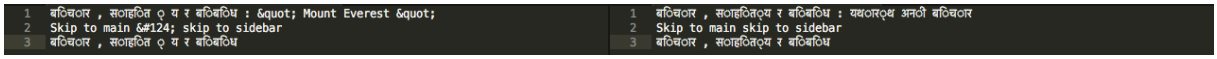
To prove the results of the corpus filtering, a random set of 218K sentence pairs was pulled from the original parallel corpus. The resulting SMT built with this dataset had a BLEU score of 1.32. This proved that our filtered corpus was indeed superior.

As part of our research, we looked to add to the BLEU score by including translations from an English-Nepali dictionary in our SMT training data, which contains almost 10K entries. This helped to raise the BLEU score up to 3.6. Table 1 summarizes the results obtained from each machine translation system depending on the size and quality of the parallel corpus.

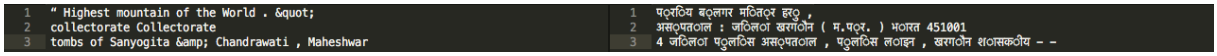
We discovered a direct correlation between the value of the compositionality metric and the quality of a sentence pair. The average compositionality value among all of the positive SVM training examples was 0.47943 while the negative examples had an average of 0.32956.

4 Discussion and Conclusion

In this paper we explored different ways to clean a noisy parallel corpus for a low resource language. We developed a classifier to predict whether or not a sentence pair should be included in a SMT dataset. By reducing the corpus to 10% of its original size, we were able to show an improved BLEU score. We also showed that this filtered corpus was indeed better than a random data set of the same size taken from the original corpus. It is possible that results could have been im-



(a) Original unfiltered parallel corpus



(b) Filtered parallel corpus

Figure 1: Parallel corpus examples: left is English and right is Nepali.

proved if more positive training examples were available, as we proved that we were able to improve the BLEU score by including a dictionary with the filtered corpus. Had the original parallel corpus been larger, say 20 million sentence pairs, a higher accuracy could have been achieved. The low BLEU score after implementing our filtering methods proves not only the low quality of the data but also the difficulty of filtering a parallel corpus for a low resource language.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Benot Frénay and Michel Verleysen. 2014. Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(15):845–869.

William A. Gale and Kenneth W. Church. 1993. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19(1):75–102.

Matthias Hartung, Fabian Kaupmann, Soufian Jebbara, and Philipp Cimiano. 2017. Learning Compositionality Functions on Word Embeddings for Modelling Attribute Meaning in Adjective-Noun Phrases. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 54–64, Valencia, Spain. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hrve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models.

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 the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. 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 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Philip Resnik and Noah A. Smith. 2003. The Web as a Parallel Corpus. *Computational Linguistics*, 29(3):349–380.