Cross-lingual Word Embeddings

Jungo Kasai[1]

[1]Yale College, New Haven

Introduction

One of the key lessons from the recent work in Natural Language Processing (NLP) is that vectorization of words, i.e. word embeddings, play an important role in various tasks. For instance, Word2Vec, developed by Mikolov et al. preserve a linguistically sensible linear structure. Such results encourage us to see word embedding as a good representation of semantics. Moreover, pre-trained word embeddings are often used as an initialization point for many tasks such as Machine Translation and Part-of-Speech Tagging. Though monolingual word embedding has been contributing to semantics and many other sub-fields in NLP as above, multilingual embedding has yet to be explored extensively. First of all, since monolingual embedding turns out to preserve semantic relations between words, it is encouraging to superpose multiple language word embeddings in the same vector space and extend our notion of semantics to multiple languages. Furthermore, multilingual embedding can potentially enrich our monolingual learning in languages with less data than English as we might be able to transfer a model in English to a model in another language. Motivated by these insights, this research project explores techniques for multilingual word embeddings in the literature. In particular, we focus ourselves on a type of techniques that bridge individually trained word embeddings across different languages. This type of technique has a practical advantage in comparison to the approaches that require word embeddings for multiple languages at the same time; it does not require any aligned dataset such as the Europarl corpora, and therefore it can utilize a larger amount of data available.

Proposed Method

Existent cross-lingual word embeddings methods can be roughly categorized into two approaches: connecting pre-trained monolingual word embeddings and joint learning across languages. The former includes Mikolov et al's attempt to superpose two vectors spaces generated from the Skipgram algorithm corresponding to two different languages through a linear transformation. Such an approach requires word pairs to formulate a quadratic optimization objective function. The objective is optimized using the stochastic gradient descent algorithm.

The latter includes the Bi-skip algorithm proposed by Luong et al. In the vanilla Skipgram scheme, we train a model to predict words around each word in one language, but the Bi-skip model jointly train a model that spans across two distinct languages. It makes the use of aligned sentence data such as the Europarl corpora and first align words in sentences using the Berkeley Aligner. Then, instead of just training a model predicting words around each word in one language, we train a model that predicts words around each word in one language, an aligned word, and words around the aligned word in the other language. Although such a method achieves systematic joint learning, it requires paired data sets, and such datasets might not be available for a language of interest, and this limitation undermines the practical applications of cross-lingual word embeddings since one potential application of cross-lingual word embeddings is transfer learning from a popular language to a language that suffers data scarcity. Therefore, we proceed with the former type of approaches.

However, Mikolov et al's method causes a problem in optimization. Since they use the stochastic gradient descent algorithm, the optimization process does not terminate after a finite number of updates, and it would require parameter tuning, which becomes costly as we align more pairs across more languages. We address this issue by incorporating the conjugate gradient descent method, which for each of d dimensions terminates after at most d time steps where d is the number of word embeddings. Moreover, this method directly allows for L-2 norm regularization.

λ	Precision@1	Precision@5	Precision@10
0.0 (First Problem)	0.338	0.483	0.539
0.5	0.336	0.477	0.533
1.0	0.331	0.473	0.528

Table 1. Test results after alignment training on 5000 translation pairs from English to Italian. Precision@5 denotes the 5-best accuracy and so forth.

λ	Precision@1	Precision@5	Precision@10
0.0 (First Problem)	0.161	0.280	0.349
0.5	0.189	0.325	0.377
1.0	0.175	0.305	0.357

Table 2. Test results after alignment training on 1000 translation pairs from English to Italian.

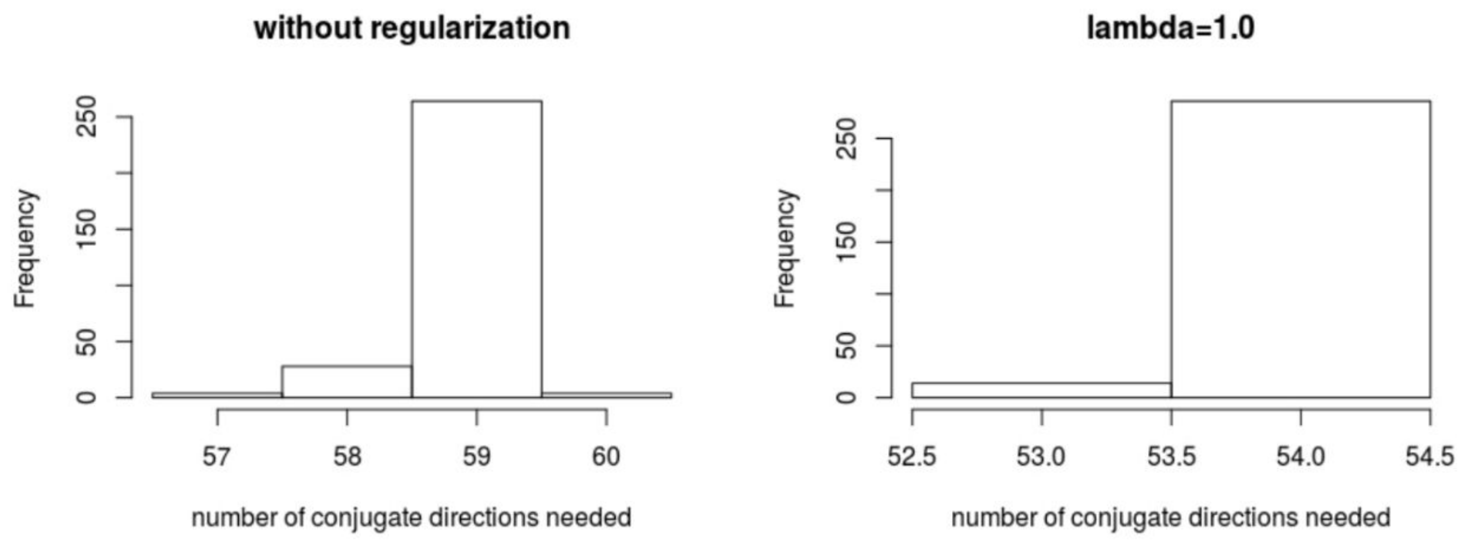

Figure 1. The number of conjugate directions need until it converges up to the threshold of 1e10.

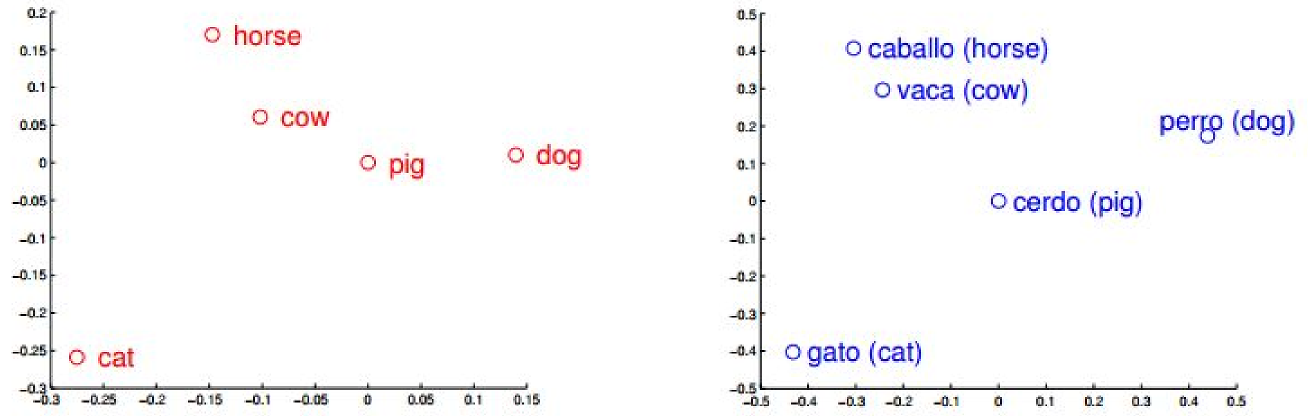

Figure 1. An example PCA example by Mikolov et al between English and Spanish.

Experiments

The CBOW Word2Vec models are trained with 2.8 billion English words from the British National Corpus, Wikipedia, and ukWaC, and 1.6 billion Italian words from itWaC to generate 300 dimensional 200K word (d=300). For both of the languages, the window sizes for the Word2Vec algorithm are all five, meaning we define the context as the five words from each side of a word. The learned vectors are normalized. Following Dinu et al, 6500 English-Italian translation pairs are extracted from a dictionary built from Europarl corpora. Then, the 6500 pairs randomly split into 5000 pairs of the training set and 1500 pairs of the test set for linear alignment. Table 1 shows the result. It should be noted that the regularization hurt the performance. We also train the linear transformation, only using 1000 pairs out of the training set. Seen in Table 2 are the results. We observe that L-2 norm regularization significantly improves performance. Although we would need more systematic analyses to draw a conclusion, we can argue that over-fitting is unlikely to be happening in the case where we have 5000 word pairs. Lastly, seen in Figure 1 is the distribution of the number of time steps required for the conjugate descent algorithm to converge, demonstrating that it efficiently optimizes the loss.

Conclusion and Future Work

We have succeeded in implementing an efficient deterministic optimization algorithm that terminates at a finite number of time steps. This method enables us to learn alignment between vectors spaces across languages from a large number of translation word pairs. Nonetheless, the results show that the learn alignments are not accurate enough to be applicable to translation task. One obvious linguistic problem with our methodology is that we fail to capture multiple meanings of words. Moreover, we do not provide a connection between the Word2Vec objective function and the alignment objective function; the Word2Vec employs an inner-product based loss, but our alignment objective function is quadratic. Replacing the quadratic loss by an inner-product based could improve alignment accuracy.