Bilingual Word Embedding for English and Tamil

ABHIJEET G, DA-IICT, India

The two dataset contains pre-trained vectors of dimension 300 and they are monolingually pre-trained and ranked according to their frequencies. They are trained in a common embedding space. The results produced by the the FastText library is tried to be emulated in this project. The Facebook machine learning tools of PyTorch , MUSE Python library and FastText datasets is used in the notebook. The 300 vectors of each corpora are then translated into a dictionary , where it is trained using discriminator function and adversarial training.

Additional Key Words and Phrases: Bilingual embedding, FastText, Word Embedding

ACM Reference Format:
Abhijeet G. 2018. Bilingual Word Embedding for English and Tamil. 1, 1, Article 1 (April 2018), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION TO THE PROBLEM

Generally, machine translation is performed by mapping each word from both the corpora, using basic dictionary translation. But, the words of the natural language has a lot of syntactic and semantic meaning, which needs a concept like word vectors to get effective translation. When the dataset is huge, the concept of word vectors will be more accurate in getting the translation from both the languages. For doing bilingual word embedding, we first assign a source language and a target language that we can import using the floatTensor data structure of PyTorch library.

2 PROBLEM STATEMENT

To try and Perform Word Embedding from 2 monolingual pre-trained word vector datasets (English and Tamil), and to try and map them similarly in a vector space. [**?**]

3 ABOUT THE DATASET

We use unsupervised word vectors that were trained using fastText 2 . These correspond to monolingual embeddings of dimension 300 trained on Wikipedia corpora; therefore, the mapping W has size 300 ÃŮ 300. Words are lower-cased, and those that appear less than 5 times are discarded for training.

Tamil Language Dataset. The dataset contains 2 million Tamil words ranked according to their frequencies, and pre-trained to give the real numbers for each word.

FastText tool.

Author's address: Abhijeet G, DA-IICT, , Gandhinagar, Gujarat, 382421, India, 201711042@daiict.ac.in.

XXXX-XXXX/2018/4-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

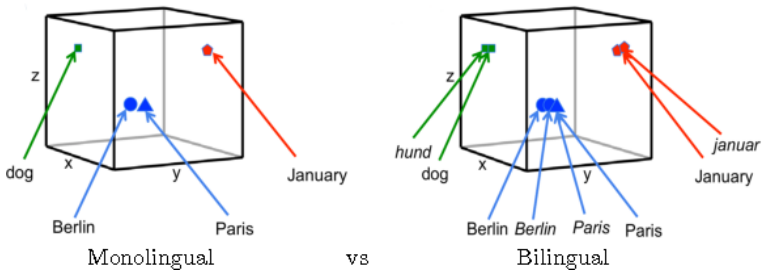

Fig. 1. Ref- Semantic Scholar

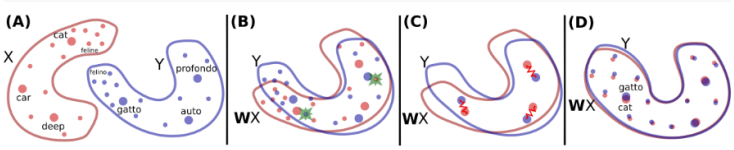

Fig. 2. Ref- FastText arxiv paper

4 APPROACH TAKEN

The discriminator aims at maximizing its ability to identify the origin of an embedding, and W aims at preventing the discriminator from doing so by making W X and Y as similar as possible.

To train our model, we follow the standard training procedure of deep ad- versarial networks i.e for every input sample, the discriminator and the mapping matrix W are trained successively with stochastic gradient updates to respectively mini- mize L and L.

The matrix W obtained with adversarial training gives good performance, but the results are still not on par with the supervised approach.

In fact, the adversarial approach tries toalign all words irrespective of their frequencies. However, rare words have embeddings that are less updated and are more likely to appear in different contexts in each corpus, which makes them harder to align.

Under the assumption that the mapping is linear, it is then better to infer the global mapping using only the most frequent words as anchors. Besides, the accuracy on the most frequent word pairs is high after adversarial training.

To refine our mapping, we build a synthetic parallel vocabulary using the W just learned with ad- versarial training. Specifically, we consider the most frequent words and retain only mutual nearest neighbors to ensure a high-quality dictionary. [**?**]

4.1 Discriminator parameters

For our discriminator, we use a multilayer perceptron with two hidden layers of size 2048, and Leaky-ReLU activation functions. The input to the discriminator is corrupted with dropout noise with a rate of 0.1. We include a smoothing coefficient s = 0.2 in discriminator predictions. Stochastic gradient descent is used with a batch size of 32, a learning rate of 0.1 and a decay of 0.95 both for the discriminator and W . The learning rate is divided by 2 every time our unsupervised validation criterion decreases.

4.2

Discriminative Training
Adversarial Training

5 METRIC USED TO EVALUATE THE SHARED EMBEDDING SPACE

Definition 5.1 (k-NN search).

6 RESULT AND EVALUATION

7 CONCLUSION AND FUTURE WORK

The mapping of the two corporas is done and the words are mapped to their close IDs , which can be related. The pre-trained dataset consists of 1 million words out of which only 350-400 words were used in training the mapping from source language to target language. The future work can be to map all the 1 million word vectors , and then draw more inferences from it.

8 REFERENCES

(1) WORD TRANSLATION WITHOUT PARALLEL DATA Alexis Conneau,Guillaume Lample
(2) @inproceedingsgrave2018learning, title=Learning Word Vectors for 157 Languages, author=Grave, Edouard and Bojanowski, Piotr and Gupta, Prakhar and Joulin, Armand and Mikolov, Tomas, booktitle=Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), year=2018
(3) @articlebojanowski2016enriching, title=Enriching Word Vectors with Subword Information, author=Bojanowski, Piotr and Grave, Edouard and Joulin, Armand and Mikolov, Tomas, journal=arXiv preprint arXiv:1607.04606, year=2016

A WORD EMBEDDING VS LSA

ACKNOWLEDGMENTS