word2vec skipgram implementation

Arun Govind M
arungm@iisc.ac.in

Abstract

In this assignment we implement and experiment with word embeddings from word2vec skipgram model. The embeddings are then evaluated on SimLex-999 word similarity task.

1 Introduction

In this assignment we implement and experiment with word embeddings from word2vec skipgram (Mikolov et al., 2013) model trained on the Reuters corpus. The best performing model is then evaluated on SimLex-999 (Hill et al., 2015) word similarity task.

2 Data

The Reuters corpus is used to learn the word embeddings. The Reuters Corpus contains 10,788 news documents totaling 1.3 million words. The documents have been classified into 90 topics, and grouped into two sets, called "training" and "test"; thus, the text with fileid 'test/14826' is a document drawn from the test set. This test-train split is used to train the model.

3 Methodology

3.1 Preprocessing

The training set contains a total of 1,253,696 word long with 35,247 unique words. Removing the stopwords and normalizing the case(all to lower case) we are left with 626,962 words long corpus with 24,640 unique words.

3.2 Negative Sampling

The skipgram model learns embeddings by learning to maximise the probability of predicting the context words given the target word. The loss function for the model is defined as

$$loss = -\sum \log \frac{1}{1+e^{-v_w^T v_c}} - \sum \log \frac{1}{1+e^{v_w^T v'_c}}$$

Where c' represent the negative contexts. These are sampled from the following unigram distribution of the words.

$$P(w) = \frac{count(w)^{0.75}}{\sum count(w_i)^{0.75}}$$

The training data is created by collecting *(target,context)* pairs from a sliding window over the corpus.

4 Implementation

The word2vec skipgram model is implemented as a neural network with 2x(vocabulary)x(embedding size) parameters updated with tensorflow's GradientDescentOptimizer with learning rate 0.1

5 Experiments

The hyperparameters for the model are dimensions of the embeddings, window size, batch size, number of negative samples.

6 Results

6.1 SimLex-999 similarity task

The models are evaluated by comparing the cosine similarity of the word vectors (50-d vectors) with the SimLex-999 rating for corresponding word pairs. The Pearson correlation coefficient between the two aforementioned values is the evaluation metric cited below.

batch size	window size	negative samples	correlation
16	3	1	**0.1863835**
16	3	2	0.0551504
32	3	1	0.0941699
32	5	1	0.1303181

Table 1: Correlation with SimLex-999 rating

6.2 Qualitative results

Below are the qualitative results for the best performing model visualized using PCA.

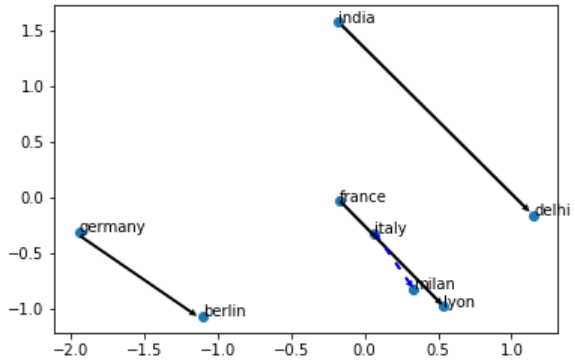

Figure 1: PCA visualization of Countries-cities

The 2-d PCA visualisation of the word embeddings shown above reflects semantic relationship between countries and cities.

References

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41.4:665-695.

Tomas Mikolov, Ilya Sutskever, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.