Shashwat Jain
2017103

Report

Question1

The code based on the Word2vec algorithm was used to compute the vector values for each unique initial 10000 and 1000 words from the abc corpus of nltk.

The code was run twice, for 1000 initial words (5000 epochs) and 10000 initial words (50 epochs).

The plots after every multiple of 1000 epochs are in *plot1.zip* (for initial 1000 words of abc corpus) and the plots for initial 10000 are in *plot2.text* for every multiple of 10. The result of both of them is in text file *result1.text* and *result2.txt* respectively.

Algorithm Explanation

We train our data using a 3 layer neural network and apply the logistic regression approach on each layer. We map each of the unique words to the id, and after that, we make the pair of neighboring words of each word within the distance *window*. We use these pairs as training data after converting each word into its id using a one-hot encoding approach.

dim[X]=[number of pairs,unique words in corpus]
dim[W1]=[unique words in corpus,embedding size]
dim[B1]=[embedding size]

Hidden_layer= [X].[W1]+B1

dim[hidden_layer]=[number of pairs,embedding size]

dim[W2]=[embedding size,vocabulary size]
dim[B2]=[vocabulary size]

output_layer=[hidden_layer][W2]+B2

dim[output_layer]=[number of pairs,vocabulary size]

Now, we convert this output_layer to probabilities using the ***softmax function*** *with* ***negative sampling***

Finally the loss is calculated using the function:
loss(output_layer)=*Y*log(softmax(output_layer))*

The model runs multiple times and updates the values of W1, W2, B1, B2 such that means the loss is minimum for the next epoch using the backpropagation.

After training

res =W1 + B1
dim(res)=[**vocabulary size , embedding size**]

Finally, we get a row corresponding to each unique word of embedding size

That means loss is minimum. It's obvious that words with a greater difference of their corresponding vector will produce more varied corresponding results in output_layer, hence res gives us the sense o*f separation between the two words*.

Question 2

Relevance feedback:

Retrivel Scores:

Iteration 1= 0.592663091194912
Iteration 2=0.620253261195146
Iteration 3= 0.6202532611951462

Relevance feedback with the query expansion:

Retrieval Scores:

Iteration 1= 0.9962933966033297

Iteration 2= 0.9962933966033297
Iteration 3= 0.9962933966033297

References

1. https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

2. https://towardsdatascience.com/google-news-and-leo-tolstoy-visualizing-word2vec-word-embeddings-with-t-sne-11558d8bd4d

3. https://en.wikipedia.org/wiki/Word2vec

4. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

5. https://towardsdatascience.com/learn-word2vec-by-implementing-it-in-tensorflow-45641adaf2ac