NLP Unit Evaluation 1

Team N-16 :

- Sameer Ravindra Kulkarni - 1PI13CS132
- P Sai Vishwas - 1PI13CS102
- Anirudh Agarwal - 1PI13CS199
- Rohan Agarwal - 1PI13CS124

Problem Statement

To determine the similarity between 2 words from the given corpus using the principles of Language model and Continuous Bag of Words (CBOW). We then evaluate the performance by comparing this with the scores produced by Word2Vec .

Input: A dataset consisting of large number of tweets is used to obtain similarity between words.

Preprocessing :

1. The tweets were first cleaned .
   a. All hashtags are removed
   b. All screen names are removed
   c. Similar action was taken on URLs , RT and emoticons .
2. All tweets were converted to lowercase
3. We have not used WordNetLemmatizer for lemmatization or PortStemmer for stemming.

[Stemming and Lemmatization leads to some unnecessary removals like it converts “was” to “wa” etc.] . Thus, as a result after trying with both of them, we decided to create corpora without it.

Unigram Model :

1. A unigram language model was built
2. We decided to remove 15% of the least frequent words and replace them with the token ‘##token##’
3. Probability of a word is calculated as count(word)/count(vocabulary), independent of the other words.
4. The triplets are created and scores for word pairs selected are calculated.

Test Case Generation :

1. 10 random word pairs were selected

Our Implementation :

1. We extracted all triples from shortened corpus after threshold cutting.
2. We then calculated D and Z for each word pair.
3. With these values we then calculated the similarity score between each pair of words chosen in our test cases.

Word2Vec :

1. While training model for Word2Vec, we take min_count (minimum frequency count below which all words are ignored during training) as 176 which was the threshold frequency for the corpus for our model.
2. Also window parameter is set to 3 and Word2Vec was trained by passing list of lists consisting of complete sentences.
3. For the same word pairs we computed the similarity using Word2Vec

Word 1	Word 2	Our Implementation	Word2Vec
dubai	city	0.0049510	0.247316443111
rahul	sonia	0.07470288	0.726216626567
modi	modi	1.0	1.0
modi	pm	0.049966077	0.254416567231
taliban	terrorism	0.1125639	0.343685154454
govt	government	0.13193885	0.4082797
namo	modi	0.04439482	0.348291588955
muslim	arab	0.05927180	0.155806067401
congress	bjp	0.0813589	0.21507357963
india	uae	0.201879788	0.239223803887

Conclusion :

> We can see that the results given by our implementation are visibly different from the values obtained using Word2Vec .

> This is because in our implementation we consider only the immediate words on either side of the selected word whereas Word2Vec uses deep neural networks and does not restrict itself to the immediate word on either side of the selected word.

> However, when tested for a sample corpora :
The Alexander was great
The Sikandar was great
Comparing (Alexander, Sikandar) from our model gave 1.0
While Word2Vec for same sample corpus gave 0.00834

> Also we observe that when we ask for the similarity of a word with itself we get a score = 1.0 for both implementations.