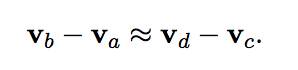
# Assignment 5 Report

Problem 1

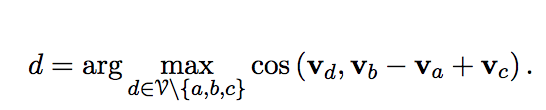
In this problem, we are supposed to perform Mikolov analogy test on the Mikolov analogy dataset. The Mikolov analogy test states that to predict analogy among words a simple algebraic operations can be used. Word analogy can be predicted if we are given a pair of words<a,b> and a third word c then we choose a forth word say d such that the analogy a is to b is same as c is to d. That is, the relationship between c and d should be as close as possible to that of a and b.

Word embedding is the collective name for a set of language modelling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers.

This can be defined using vectors. Let Let va be the vector for a, vb the vector for b, and so on. For the d such that the analogy holds, we expect



Then we need,



The analogy test is executed on GloVe Wikipedia 2014 + Gigaword 5 and GloVe Twitter. These set contains pre-trained vectors trained using GloVe algorithm. The corpus is used as Wikipedia 2014 + Gigaword 5 database containing about 6 billion words and has a vocabulary size of 400K each of which are represented using 50,100,200 and 300 dimensional vectors. I have used 100 dimensional vector. This embedding can be downloaded from: <http://nlp.stanford.edu/data/glove.6B.zip>

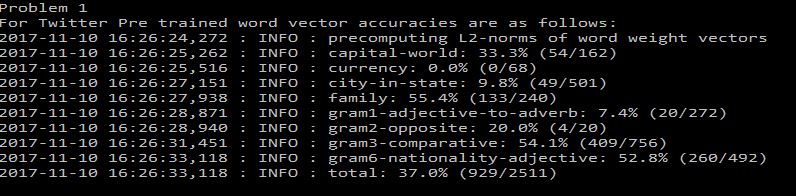
The other corpus is used as Twitter containing about 2 billion tweets and has a vocabulary size of 1.2 million each of which are represented using 25, 50,100 and 200 dimensional vectors. I have used 50 dimension vector. This embedding can be downloaded from:

<http://nlp.stanford.edu/data/glove.twitter.27B.zip>

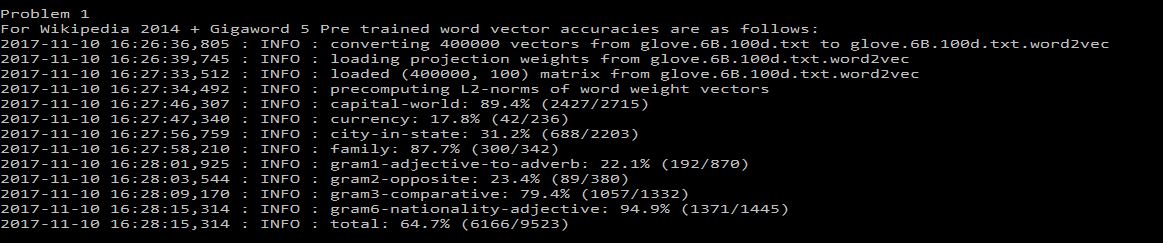
To implement this in python, I have used Gensim package which can be installed using “pip intall --upgrade genism” in command prompt. This package contains “genism.script.glove2word2vec” which is used to convert .txt given as an input and the returns .word2vec file as output. Hence, GloVe format can be easily converted to word2vec format. Then, “gensim.models.KeyedVectors.load\_word2vec\_format” initiates word2vec model by calling the method load\_word2vec\_format. Now, this method will take word2vec format file which we got as the output from above.

**Accuracies**

For GloVe Twitter as pre trained embedding the output for the file word-text.txt (which contains all the 8 analogies that is capital-world, currency, city-in-state, family, gram1-adjective-to-adverb, gram2-opposite, gram3-comparative, and gram6-nationality-adjective) is as follows:



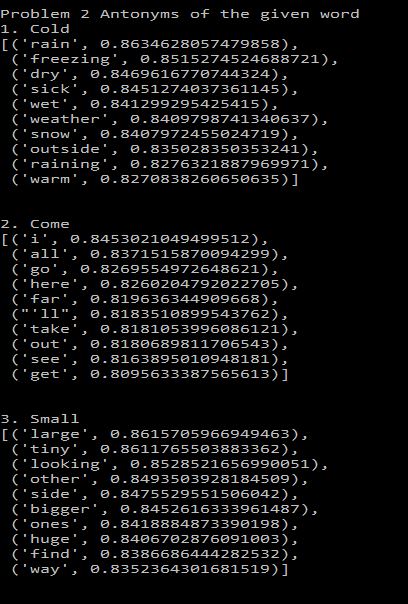
For GloVe Wikipedia 2014 + Gigaword 5 as pre trained embedding the output for the file word-text.txt (which contains all the 8 analogies that is capital-world, currency, city-in-state, family, gram1-adjective-to-adverb, gram2-opposite, gram3-comparative, and gram6-nationality-adjective) is as follows:



**Problem 2**

The known problem with word embeddings is that antonyms often have similar embedding. The main reason for this is that the Word2Vec algorithm places two words close to each other, if they are used in similar contexts. Hence, if we considering an example for the word increase which has decrease as its antonym then both words can appear in similar context like “increase in temperature” and “decrease in temperature”. Therefore, the vectors of these two words are similar.

For GloVe Twitter as pre trained embedding the output for the file word-text.txt and input words as “cold”, “come” and “small” is as follows:



Here we can see that antonym of the words:

1. Cold

[('rain', 0.8634628057479858),

('freezing', 0.8515274524688721),

('dry', 0.8469616770744324),

('sick', 0.8451274037361145),

('wet', 0.841299295425415),

('weather', 0.8409798741340637),

('snow', 0.8407972455024719),

('outside', 0.835028350353241),

('raining', 0.8276321887969971),

('warm', 0.8270838260650635)]

2. Come

[('i', 0.8453021049499512),

('all', 0.8371515870094299),

('go', 0.8269554972648621),

('here', 0.8260204792022705),

('far', 0.819636344909668),

("'ll", 0.8183510899543762),

('take', 0.8181053996086121),

('out', 0.8180689811706543),

('see', 0.8163895010948181),

('get', 0.8095633387565613)]

3. Small

[('large', 0.8615705966949463),

('tiny', 0.8611765503883362),

('looking', 0.8528521656990051),

('other', 0.8493503928184509),

('side', 0.8475529551506042),

('bigger', 0.8452616333961487),

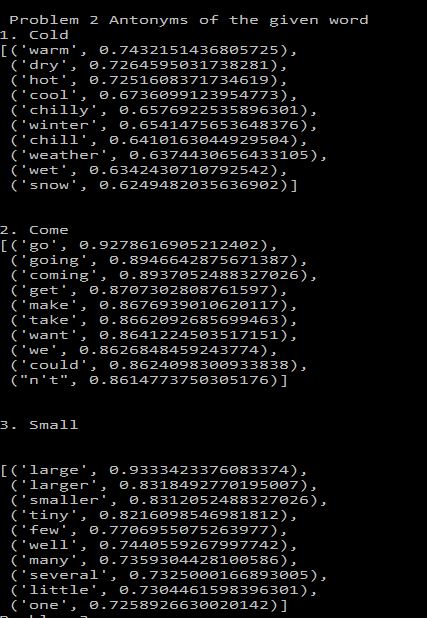
('ones', 0.8418884873390198),

('huge', 0.8406702876091003),

('find', 0.8386686444282532),

('way', 0.8352364301681519)]

Similarly for GloVe Wikipedia 2014 + Gigaword 5 as pre trained embedding the output for the file word-text.txt and input words as “cold”, “come” and “small” is as follows:



Here we can see that antonym of the words:

1. Cold

[('warm', 0.7432151436805725),

('dry', 0.7264595031738281),

('hot', 0.7251608371734619),

('cool', 0.6736099123954773),

('chilly', 0.6576922535896301),

('winter', 0.6541475653648376),

('chill', 0.6410163044929504),

('weather', 0.6374430656433105),

('wet', 0.6342430710792542),

('snow', 0.6249482035636902)]

2. Come

[('go', 0.9278616905212402),

('going', 0.8946642875671387),

('coming', 0.8937052488327026),

('get', 0.8707302808761597),

('make', 0.8676939010620117),

('take', 0.8662092685699463),

('want', 0.8641224503517151),

('we', 0.8626848459243774),

('could', 0.8624098300933838),

("n't", 0.8614773750305176)]

3. Small

[('large', 0.9333423376083374),

('larger', 0.8318492770195007),

('smaller', 0.8312052488327026),

('tiny', 0.8216098546981812),

('few', 0.7706955075263977),

('well', 0.7440559267997742),

('many', 0.7359304428100586),

('several', 0.7325000166893005),

('little', 0.7304461598396301),

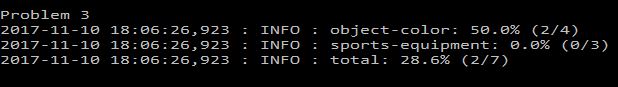
('one', 0.7258926630020142)]

Hence, we can say that word embeddings will have antonyms that often have similar embedding that is Word2Vec algorithm places two words close to each other, if they are used in similar contexts.

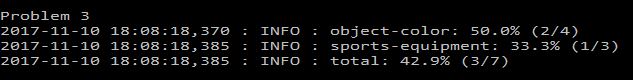
**Problem 3**

In this problem, I have designed two new analogy tests that are not part of Mikolov’s analogy dataset. In the first analogy data set is “object-color” that is I have defined an object and the respective color of that object. The second analogy dataset “sports-equipment”, I have taken a sports and the equipment used to play that sports.

For GloVe Twitter as pre trained embedding the output for the file problem3.txt



For GLoVe Wikipedia 2014 + Gigaword 5 as pre trained embedding the output for the file problem3.txt



It is interesting to note that the word embedding are biased that is the observation was actually done during performing the experiment for Problem 1. In GloVe Twitter we can see that there is no accuracy for sport-equipment where as in GloVe Wikipedia 2014 + Gigaword 5 we get an accuracy for sports-equipment. Hence, we can say that there is variation in the accuracy, which clearly indicates the biasness of two datasets.

**References**

1. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. <http://arxiv.org/pdf/1310.4546.pdf>
2. Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation. [https://nlp.stanford.edu/pubs/glove.pdf](https://nlp.stanford.edu/pubs/glove.pdf%20) <https://nlp.stanford.edu/projects/glove/>
3. Radim Rehurek. Word2Vec in Phython, Gensim. <https://radimrehurek.com/gensim/index.html>
4. Gensim KeyedVectors. <https://radimrehurek.com/gensim/models/keyedvectors.html>
5. Gensim Script Glove 2 Word2Vec. <https://radimrehurek.com/gensim/scripts/glove2word2vec.html>