**PROBLEM 1:**

I have considered the following two pretrained word embeddings:-

1) Word2Vec

2) Glove (Global vectors for word representation)

In order to implement the above two embeddings, an open source library called ‘GENSIM’ is used. This library comes along with ‘ANACONDA’ and can be installed using the following command:

**pip install gensim**

Using gensim ‘word2vec’ can be easily implemented.

**1) Word2Vec**

For Word2Vec, I used the pre-trained vectors trained on Google News dataset. This model contains 300 dimensional vectors for 3 million words and phrases. The corpus can be downloaded here:

<https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing>

As the vectors set is too large to handle, while building the model I have set the limit to 1 million. Still it takes around 15 minutes to compute the Accuracy for all the eight modules in the ‘Mikolov’s Analogy dataset’.

The computed accuracy is as follows:

|  |  |
| --- | --- |
| **Module** | **Accuracy** |
| capital-world | 21.551724137931033 |
| currency | 35.33487297921478 |
| city-in-state | 71.90920145926226 |
| family | 85.17786561264822 |
| gram1-adjective-to-adverb | 28.830645161290325 |
| gram2-opposite | 42.857142857142855 |
| gram3-comparative | 90.91591591591591 |
| gram6-nationality-adjective | 89.99374609130707 |

**2) Glove**

The pre-trained word embedding to build a Glove model can be downloaded from the below url:

<http://nlp.stanford.edu/data/glove.42B.300d.zip>

It contains 42b tokens, 1.9 million tokens and 300 d vectors. The corpus is around 1.75 GB and hence takes around 15 minutes to create the model and compute the accuracy on test data.

The accuracies are as follows:-

|  |  |
| --- | --- |
| **Module** | **Accuracy** |
| capital-world | 0 |
| currency | 0 |
| city-in-state | 0 |
| family | 90.9090909090909 |
| gram1-adjective-to-adverb | 30.241935483870968 |
| gram2-opposite | 35.59113300492611 |
| gram3-comparative | 85.58558558558559 |
| gram6-nationality-adjective | 0 |

From the above table of accuracies, we can make it out that the training corpus has no data regarding the capital world, currency, city in state and nationality-adjective. Hence the accuracy for all these modules turned out to zero.

**PROBLEM 2:**

In this case, I have considered five different words to check if they appear mostly with their antonyms. The words are:

**----increase**

[('decrease', 0.8370318412780762),

('increases', 0.7709376811981201),

('increased', 0.7578041553497314),

('reduction', 0.6908220648765564),

('increasing', 0.6871615648269653),

('decreases', 0.6816173791885376),

('rise', 0.6352647542953491),

('decreasing', 0.6218624114990234),

('decline', 0.6128641366958618),

('uptick', 0.5923734903335571)]

**----up**

[('down', 0.6396992206573486),

('out', 0.5464873313903809),

('off', 0.5370627045631409),

('ups', 0.4826122522354126),

('upthe', 0.47866734862327576),

('in.', 0.4756893813610077),

('up.The', 0.4518883228302002),

('around', 0.4468981623649597),

('aside', 0.440209299325943),

('away', 0.43584108352661133)]

**----agree**

[('disagree', 0.7711759209632874),

('concur', 0.7131548523902893),

('agrees', 0.5929451584815979),

('disagreed', 0.5711543560028076),

('Agree', 0.5635050535202026),

('disagreeing', 0.5525435209274292),

('respectfully\_disagree', 0.5464814901351929),

('agreed', 0.5445383787155151),

('insist', 0.5273953676223755),

('accept', 0.5188775658607483)]

**----enter**

[('entering', 0.7399863004684448),

('entered', 0.6956064701080322),

('reenter', 0.6487391591072083),

('enters', 0.5622495412826538),

('entry', 0.551796019077301),

('Entering', 0.48456794023513794),

('participate', 0.48203885555267334),

('leave', 0.4764121174812317),

('join', 0.4726879596710205),

('register', 0.4570498466491699)]

**----beautiful**

[**('gorgeous', 0.8353004455566406),**

**('lovely', 0.8106936812400818),**

**('stunningly\_beautiful', 0.7329413890838623),**

**('breathtakingly\_beautiful', 0.7231341600418091),**

**('wonderful', 0.6854087114334106),**

('fabulous', 0.6700063943862915),

('loveliest', 0.6612576246261597),

('prettiest', 0.6595001816749573),

('beatiful', 0.6593325138092041),

('magnificent', 0.6591403484344482)]

This clearly indicates that the first three words: **Increase, up and agree –** occur most frequently with their respective synonyms – decrease, down and disagree.

Whereas the other two words – **enter and beautiful** has its synonyms as the most similar words. Because ‘beautiful’ is used mostly to describe a pleasant context where its synonyms are used.

**PROBLEM 3:**

I have created the following two analogy sets:

**1) Cause and effect analogy**

cause\_effect\_list = [['sunrise','dawn','sunset','dusk'],

['tired','sleep','hungry','eat'],

['stress','anxiety','war','destruction']]

**2) Performer and action analogy**

performer\_action\_list = [['painter','paint','dancer','dance'],

['teacher','educate','student','learn'],

['artist','paint','scientist','research']]

This new analogy set is applied on the word2vec model and the accuracies are obtained as zero for both of them.

The most similar words for the new analogy test set are:

1)

['sunrise','dawn','sunset','dusk'] ('twilight', 0.5267351269721985)

['tired','sleep','hungry','eat'], ('sleeping', 0.4959504306316376)

['stress','anxiety','war','destruction'] ('wars', 0.57405686378479)

Accuracy: 0.0

2)

['painter','paint','dancer','dance'] ('dancers', 0.49229949712753296)

['teacher','educate','student','learn'], ('educating', 0.5912973284721375)

['artist','paint','scientist','research'] ('researcher', 0.41154393553733826)

Accuracy: 0.0

Accuracy for both the analogies is zero as the words never occurred together in a context throughout the train data.