Meaning and Computation – Exercise 2

In this exercise you will use the word2vec model for determining the distributional similarity measure between words. Use Python and the *gensim* package are highly recommended for the experiments.

Setting Up the System:

- The gensim package homepage and installation guidelines can be found here:

  https://radimrehurek.com/gensim/index.html

  https://radimrehurek.com/gensim/install.html

Training:

- Train four word2vec models, with window sizes of 1 and 10 (window size is the number of words to the left and to the right of the target word that we consider as contexts), and with embedding dimensions of 10 and 500.
  a. Use a min_count value of 5, which only creates vectors for words that appeared 5 times or more in the training corpus (less than that would be unreliable).
  b. Code Example:

```
sentences, cur_sent = list(), list()

with open(corpus_filename, errors='ignore') as f:
    for line in f:
        line = line.strip()
        if line == '</s>':
            sentences.append(cur_sent)
            cur_sent = list()
        elif line != '<s>' and not line.startswith('<text') and not line.startswith('</text'):
            cur_sent.append(line.split('\t')[0])

model = gensim.models.Word2Vec(sentences,

                                min_count=5,window=10,size=500)
```

Correlation with Simlex-999:

- Compute the similarity between each pair of words that appears in the Simlex-999 corpus. Use the gensim *similarity* method to compute the similarity between pairs of words.
- Example:

```
print(model.similarity(word1,word2))
```

- If the model was unable to create a vector for one of the words, assume its similarity with any other word is 0.
- For each model, compute the correlation between the list of similarities induced by the model, and the list of gold-standard similarities supplied with the Simlex corpus. Report correlations for each of the POS tags as well.
- Correlation should be computed using the Spearman correlation coefficient. See here for some explanation: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient
- You are **not** required to implement the Spearman coefficient. You can use any implementation you find/program, but you are advised to use an existing python function to do that (available through the scipy package): http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.spearmanr.html

Files:

- We will use a corpus in a similar format to ex1, taken from Wikipedia:

  https://www.dropbox.com/s/t7taandme7b0qeg/wackypedia_en1.words10.20Mwords?dl=0

- We will use the simlex-999 dataset for evaluation. The dataset contains a list of words and the human-annotated similarity scores assigned to them. The models we will build will be evaluated against human judgements on the Simlex-999. The dataset can be found here:

  https://www.cl.cam.ac.uk/~fh295/SimLex-999.zip

Questions:

1. Compute the Spearman correlation between the similarities produced by word2vec and the human similarity scores using each of the four models you trained. Report your results.
2. Compute the correlation for each of the POS tags separately (Adjective, Nouns and Verbs). Does the model produce similar results for different POS? Hypothesize why this is so.
3. How does changing the window size and model size alter the results?

In answering questions 1-3, you are advised to consider the similarity measures produced by the different models (i.e., the similarity scores for each pair of words), and not only the overall Spearman correlation.

4. Gensim implements a method of predicting the answer to an analogy, as in "king is to ?, as male is to female", by finding the closest word vector to the vector *v("king")+v("woman")-v("man")*. Example:

   model.most_similar(positive=['woman', 'king'], negative=['man'])

   Think of five examples of analogies, and examine the predictions of word2vec relative to your predictions. Do word2vec's predictions make match yours? Do they make sense? Can you characterize which type of relations are captured using this method and which are not?

   Note: characterizing which type of relations are and are not captured using the method is clearly a difficult and somewhat open-ended question. Any insight or well-formulated hypothesis would count as a valid answer to this question.

Submission:

In addition to answers for the "Additional questions" above, please also provide the source code you have written, as well as the computed similarity measures for each of the four models, and their Spearman correlation coefficients with the Simlex measures. Please do so both for the entire Simlex dataset, and for each POS tag individually.

The computed similarity scores for a model, for each pair should be printed each to a separate file. Each file should have the format of the Simlex-999.txt dataset. That is, each line should be of the following format:

<Word1> <word2> <POS tag> <similarity score>

The files should be named size500_window1, size500_window1, size10_window10, size500_window10

Grade:

Grades will be given on a scale of 0 to 7. The exercise will account for 7% of the final grade.