Homework 2
Name: Wang Jue
NetID: jwr0983

GitHub Link: https://github.com/MSIA/jwr0983_msia_text_analytics_2020/tree/homework2

Problem 1

For this problem I used 3 different models to evaluate the performance of different parameters for word2vec in gensim library. The first one uses CBOW with window size = 5, embedding size = 100 and minimum count = 3, the second using CBOW with window size = 10 and embedding size = 150 and minimum count = 3, while the third one uses skip-gram with window size = 10, embedding size = 100 and minimum count = 3.

Computation run-time:

- CBOW with window size 5 and embedding size 100: 1.95 seconds
- CBOW with window size 10 and embedding size 150: 1.98 seconds
- Skip-gram with window size 5 and embedding size 100: 2.97 seconds

From the above results we see that skip-gram is generally slower than CBOW on this dataset. The reason is that skip-gram approach involves more calculations. For the same window size skip gram has to undergo significantly more backward propagations than CBOW, making it computationally more inefficient.

Similar words:

We used the following 10 words to test whether the three models give comparable results in terms of similar words. The results are mostly similar, and as an example, for the word "god", the first model gives the top 5 similar words "existence, fact, belief, evidence, and manifestation", the second model gives "existence, fact, belief, evidence, manifestation", while the third model gives "satan, exist, believing, existance and exists". As the corpus is relatively small, we see that the embedding size of 100 is sufficient to give sensible results.

Problem 2

	Distributed representations of words and phrases and their compositionality	**Bert: Pre-training of deep bidirectional transformers for language understanding**
Date of Publication	16 Oct 2013	11 Oct 2018
Number of Google Scholar Citation	22964	10265

Ease of installation	Original implementation in C; Available in Python genism library	Available in pytorch transformers library and other major frameworks
Algorithm Summary	Proposes two model architectures for learning distributed representations of words that try to minimize computational complexity including Continuous Bag-of-Words Model and Continuous Skip-gram Model. Continuous Bag-of-Words Model predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word. The encoded word vectors show **multiple degrees of similarity**, it was shown for example that vector(”King”) - vector(”Man”) + vector(”Woman”) results in a vector that is closest to the vector representation of the word Queen	BERT stands for Bidirectional Encoder Representations from Transformers. BERT is designed to pre- train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. The model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based on its context.
Corpus Size in Original Implementation	In the original paper the authors have restricted the vocabulary size to 1 million most frequent words	For the pre-training corpus we use the BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). Use document-level corpus rather than shuffled sentence-level corpus
Training Complexity	CBOW: $Q = N \times D + D \times \log2(V)$ Skip-gram: $Q = C \times (D + D \times \log2(V))$ V is size of vocabulary, N previous words are encoded, D is dimension of encoded vector, C is maximum distance of words	Time complexity of bidirectional transformer with number of layer L, hidden size H, and number of self-attention heads as A. (For base model Total Parameters = 110M, for Large model total parameters = 340M)