**CS 590 NLP**

**HW5**

**Word2Vec**

**Due 03/23 11:59 pm**

**Overall Goal:**

In this homework you will work with word2vec embeddings to understand them. To save time, you will be again working with the document from HW1, the dataset from HW3-4, and 2 other documents.

|  |
| --- |
| **Useful python packages:** |
| For word2vec, you may find the gensim package useful: <https://radimrehurek.com/gensim/models/word2vec.html>  Note that gensim has built in pretrained word2vec embeddings so you should choose 1 or experiment with multiple in this homework. (<https://radimrehurek.com/gensim/models/word2vec.html#pretrained-models)>  Gensim also has nice built-in functions for working with word2vec data. |

|  |
| --- |
| **Dataset** |
| The dataset that will be used is the toxic comment dataset (same as HW3-4). This consists of toxic comments, so be cautioned when viewing the dataset.  You can find the training and test sets here: <https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data>  Note for the word2vec tasks you will only need to worry about the toxic label (toxic=1 or 0).  The training set (**train.csv)** will be used for training the LM, while the test set (**test.csv**) and labels **(test\_labels.csv)** will be used for testing and analyzing your models.  Take time to understand how the training set and test set are laid out. |

|  |
| --- |
| **Final Report** |
| Just like in the previous homeworks, you will be documenting observations and results in a final report. This is the best way to show your thought process throughout the homework, so it is best to update it as you go along and then refine at the end, rather than try and write all at the end. |

|  |
| --- |
| **Word2Vec Tasks** |
| Your aim is to work with embeddings separately first to understand them a bit further. You may use any pretrained word2vec embeddings you choose. **Note: This assignment is to practice working with pretrained embeddings, so you should not train any of your own.**  **compare\_texts\_w2v(file\_one, file\_two, k = 10)**  This method takes in two dataset files and compares them at the word level by leveraging word2vec. First, this function finds the k most common non stop words for each file. Then these 2\*k words will be used to calculate:   * Similarity between the 2 text files based on the top k words. It is up to you to determine how you want to calculate this similarity score, but cosine similarity should be involved and the result should be a numerical score. (You should also document what algorithm you use).   You should then print out a nice summary for the above statistics, no need to return anything. You may also use helper functions to help your function look nicer.  The above statistics should all go into your final report document. Additionally, you should make some observations about both the similarity statistics. Is anything surprising or not? (This should be more than just pointing out what the values are.)  With the above function, you should make the following function calls:   1. compare\_texts\_word2vec(NOT\_TOXIC\_subset, TOXIC\_subset, k = 5) 2. compare\_texts\_word2vec(NOT\_TOXIC\_subset, TOXIC\_subset, k = 10) 3. compare\_texts\_word2vec(NOT\_TOXIC\_subset, TOXIC\_subset t, k = 20)   Where NOT\_TOXIC\_subset refers to a subset of the dataset where the toxic labels are 0 and TOXIC\_subset refers to a subset of the dataset where the toxic labels are 1. You will have to create these files yourself (**the word2vec function should not be automatically creating them**).  **doc\_overview\_w2v(text\_file, k=10, n=5)**  This method gives an overview by summarizing the top k words and their nearest neighbors for a given document. Thus, you should first follow a similar approach as above for finding the top k frequent non stop words. Then, you should leverage gensim to find the n most similar words for each top k word. You can then print out the document and a summary of these words.  With the above function you should call:   1. doc\_overview\_w2v(“warofworlds.txt”, k = 5, n = 5) 2. doc\_overview\_w2v(“on\_liberty.txt”, k=5, n=5) 3. doc\_overview\_w2v(“kingarthur.txt”, k=5, n=10)   War of the Worlds: <https://www.gutenberg.org/ebooks/36>  On Liberty: <https://www.gutenberg.org/ebooks/34901>  King Arthur: <https://www.gutenberg.org/ebooks/12753>  The above overviews should all go into your final report document. Additionally, you should make some observations about the words. Is anything surprising or not?  (This should be more than just pointing out what the values are.)  **Note: Like all homeworks, you should not screenshot results. You should put them into a nice table in your report for clarity.** |

|  |
| --- |
| **Additional Tips/Guidance** |
| 1. Your thoughtfulness in approaching and analyzing your models will determine the final grade for this homework. This means that lack in analysis will receive lower scores as indicated in the grading scale. 2. Note that preprocessing the text is up to you. This assignment is not specifically evaluating preprocessing like the first assignment, but here you have a chance to practice any preprocessing for your final project and future assignments. (**You should be doing some preprocessing, or else you’ll be purposely setting your models up to fail.**) 3. Get familiar with the gensim library on your own on small sets of data. Just copying and pasting the code without fully understanding it will end up being detrimental as you may not be able to accurately give proper analysis and observations of results. 4. DO NOT SHARE CODE. I have linked gensim libraries which will be useful for this assignment. You may also come to me to discuss/figure things out. 5. Start the assignment early. You will have 2 weeks to complete this assignment, which is plenty of time if you start early any familiarize yourself with the libraries. **IF YOU WAIT UNTIL THE LAST MINUTE TO START, YOU WILL BE LESS LIKELY TO DO WELL.** I won’t be granting any additional extensions so each late assignment with follow the late grade policy. 6. Ask questions/approach like a researcher. Think like this is your chance to explore NLP models and analyze their effectiveness. |

**Grading**

Assignment will be graded as follows:

|  |  |
| --- | --- |
| **Description** | **Points** |
| Code Runs | 5 |
| word2vec Implementation/Tasks | 20 |
| Report analysis and observations | 15 |
| Documentation (Comments, functions, etc) | 10 |
| **Total:** | **50** |