CS 505 – Spring 2022 – Assignment 2 (100 pts) – Vector Space Models

Problems due 11:59PM EST, March 29, 2022.

In this assignment, you will learn about **vector space model**, and use python libraries such as **spacy**, **sklearn**, **gensim**, and **huggingface**, which are popular in NLP. You have 2 weeks to finish this particular assignment.

Submit in Gradescope by 11:59PM EST, March **29**, 2022
–Download the submission (answer sheet) template at this **link** and use it to write down your write-up answers.
–Please indicate names of those you collaborate with.
–Every late day will reduce your score by 20
–After 2 days (i.e., if you submit on the 3rd day after due date), it will be marked 0.

Submit your submission (answer sheet) template complete with link to your code (Jupyter Notebook)

When necessary, you must show how you derive your answer.

Problem 1. *Vector Space Models (65 pts)*

The file: will_play_text.csv contains lines from William Shakespeare's plays. The second column of the file contains the name of the play, while the fifth and the sixth contain the name of the character who spoke and what they spoke, respectively. ***Tokenize*** *and* ***lower case*** *each line in will_play_text.csv using spacy. The file vocab.txt lists the words in the vocabulary, while the file play_names.txt lists the names of the plays. For the following questions, include generated visualizations in your answer sheet.*

1. *(5 pts) Create a term-document matrix where each row represents a word in the vocabulary and each column represents a play. Each entry in this matrix represents the number of times a particular word (defined by the row) occurs in a particular play (defined by the column). Use CountVectorizer in sklearn to create the matrix, using the file vocab.txt as input for the vocabulary parameter. From your term-document matrix, use PCA in sklearn to create a 2-dimensional representation of each play. Visualize these representations to see which plays are most similar to each other. Include the visualization in your answer sheet. You can follow the tutorial here to create the visualization.*

2. *(2 pts) What plays are similar to each other? Do they match the grouping of Shakespeare's plays into comedies, histories, and tragedies here?*

3. *(3 pts) Create another term-document matrix where each row represents a word in the vocabulary and each column represents a play, but with TFIDF counts (using TFIDFVectorizer in sklearn and vocab.txt for vocabulary). Use PCA again on these TFIDF term-document matrix and visualize the plays. Include the visualization in your answer sheet.*

4. *(2 pts) Does using TFIDF give you better grouping of plays? Why do you think so?*

5. *(4 pts) Create a word-word matrix where each row (and each column) represents a word in the vocabulary (vocab.txt). Each entry in this matrix represents the number of times a particular word (defined by the row) co-occurs with another word (defined by the column) in a sentence (i.e., line in will_play_text.csv). Using the row word vectors, create a representation of a play as the average of all the word vectors in the play. Use these vector representations of plays to compute average pairwise cosine-similarity between plays that are comedies (do not include self-similarities). You can use the grouping of plays in here.*

6. *(4 pts) Using vector representations of plays computed in (1.5), compute average pairwise cosine-similarity between plays that are histories, and between plays that are tragedies (do not include self-similarities).*

7. *(6 pts) Using vector representations of plays computed in (1.5), compute the average cosine-similarity between plays that are comedies and histories, the average cosine-similarity between plays that are comedies and tragedies, and the average cosine-similarity between plays that are histories and tragedies.*

8. *(4 pts) Use gensim to learn 100-dimensional word2vec representation of the words in the play (you can use default parameters but with min_count=1 so you can learn vector representations of all the words in your data i.e., no need to use vocab.txt in this question). Use the learned word2vec representation to construct vector representations of plays as the average of all the word vectors in the play. Use these vector representations of plays to compute average pairwise cosine-similarity between plays that are comedies (do not include self-similarities)*

9. *(4 pts) Using vector representations of plays computed in (1.8), compute average pairwise cosine-similarity between plays that are histories, and between plays that are tragedies (do not include self-similarities).*

10. *(6 pts) Using vector representations of plays computed in (1.8), compute the average cosine-similarity between plays that are comedies and histories, the average cosine-similarity between plays that are comedies and tragedies, and the average cosine-similarity between plays that are histories and tragedies.*

11. *(4 pts) For this question, use the original, non-tokenized, non-lowercased file. Use LABSE in huggingface to compute vector representation of each line in will_play_text.csv. Unlike previous methods that compute word vectors, LABSE can compute whole line/sentence vectors. Compute the vector representation of each play as the average of its lines' vector representations. Use these vector representations of plays to compute average pairwise cosine-similarity between plays that are comedies (do not include self-similarities)*

12. *(4 pts) Using vector representations of plays computed in (1.11), compute average pairwise cosine-similarity between plays that are histories, and between plays that are tragedies (do not include self-similarities).*

13. *(6 pts) Using vector representations of plays computed in (1.11), compute the average cosine-similarity between plays that are comedies and histories, the average cosine-similarity between plays that are comedies and tragedies, and the average cosine-similarity between plays that are histories and tragedies.*

14. *(2 pts) Which vector representation (word-word counts vs. gensim vs. LABSE) gives the best measure of similarities between plays of similar genre vs. different genres? Why do you think so?*

15. *(4 pts) Use LABSE's vector representations of lines to construct the vector representation of each character as the average of all lines' representations that the character spoke. Visualize the characters using PCA. Include the visualization in your answer sheet.*

16. *(5 pts) Mention 2 interesting insights with respect to the grouping of plays and/or characters e.g., what are characters that are most similar/dissimilar to each other? Do the vector representations of female characters differ distinguishably from male ones? Can you find plays that are central i.e., closest to centroid to each genre (i.e., comedies, histories, tragedies)? For each insight, include explanation and/or the visualization that supports your insight.*

Problem 2. *(35 pts) In this question you will use the corpus of Shakespeare's plays (will_play_text.csv) as well as two new corpora: the works of Jane Austen (jane_austen.txt) and Sir Arthur Conan Doyle (arthur_conan_doyle.txt). For this question you will use **gensim** to create separate 100-dimensional word2vec word vector representations for each corpus. Then we will try to analyze the differences between these embeddings and try to understand what we can say about the bodies of work they were trained on.*

1. *(3 pts) Train word2vec representations for all three of the corpora (use min_count=1 to get vectors for all words in each processed corpus). Report the length of the vocabulary of each corpora. Be sure to clean and preprocess the text beforehand. Use **spacy** to sentence split, tokenize, lemmatize, lowercase, and remove stopwords.*

2. *(6 pts) Once you have the word2vec models, report the closest 3 words (using cosine-similarity) to each of these words in the list for each of the three corpora. The word list is: ['courage','hope', 'love', 'woman', 'man', 'he', 'she', 'good', 'bad','evil','beauty', 'fate'].*

3. *(3 pts) Add another three words that you think would be interesting to observe over the different corpora. Report the closest 3 words (using cosine-similarity) to each of your chosen words for each of the three authors.*

4. *(2 pts) Mention any differences between the close words across different documents. What do you think the difference indicates? For example, does the difference represent any gender bias?*

5. *(15 pts) Look into this repository. This file provides a list of analogies. Analogies are essentially meaning symmetries in language. They can be direct analogies like Paris France Berlin Germany. Here we know that what Paris is to France, The same thing Berlin is to Germany. Some earlier work has showed that this relationship sometimes is also captured in word representations. Imagine the relationship to be*

$$w(paris) - w(france) = w(berlin) - w(germany) \tag{1}$$

Then the below relationship also holds

$$w(paris) - w(france) + w(germany) = w(berlin) \tag{2}$$

This makes sense on paper but in practice the representation relationships are not always this clear. Instead we calculate the left side of the equation and check the closest word (in terms of cosine-similarity) in the vocabulary to the calculated embedding. If the word is the one on the right we say that the analogy holds. So what we want you to do is read all the analogies in the given link and for each author, remove any analogies that have OOV words. Don't try to find analogies that hold for all three authors. For each of the three authors, remove OOV analogies and from the ones that are left (i.e., don't include any OOV words), we want you to check how many of the analogies hold (i.e., calculate the accuracy). Report the analogy accuracy of each author.

6. *(6 pts) What does the accuracy scores tell you? Which corpus then gives you better commonsense representations? Do you think this is a good way to measure word representations e.g., can you think of potential issues in designing analogies?*