CSI5386: Natural Language Processing

Assignment 1
Corpus analysis and sentence embeddings

Due: Oct 11, 2022, 10pm

Note: This assignment should be done in groups of students. We use the formed BrightSpace groups. When one person in the group submits, the submission can be seen by all the members of the group.

Part 1: Corpus processing (legal text): tokenization and word counting [50 points]

Implement a word tokenizer that splits texts into tokens and separates punctuation marks and other symbols from the words. Please describe in your report all the decisions you made relative to pre-processing and tokenization. Implement a program that counts the number of occurrences of each token in the corpus.

You can use any tools for tokenization, and write programs only if you need to put the data in the right format or to compute additional information. There are many NLP tools that include tokenizers.

Use the Atticus dataset of legal contacts: https://zenodo.org/record/4595826#.YyXT6HbMI2w
Download the file CUAD_v1.zip, unzip, and see the folder full_contact_txt/
It contains 510 files with full text contracts (a collection of TXT files of the underlying contracts). Each file is named as “[document name].txt”. These contracts are in a plaintext format and are not labeled. You will need to concatenate all the text files to form a corpus.

Provide in your report the following information about the corpus:

a) Submit a file output.txt with the tokenizer’s output for the whole corpus. Include in your report the output for the first 2 .TXT files in the corpus.
b) How many tokens did you find in the corpus? How many types (unique tokens) did you have? What is the **type/token ratio** for the corpus? The type/token ratio is defined as the number of types divided by the number of tokens.
c) For each token, print the token and its frequency in a file called tokens.txt (from the most frequent to the least frequent) and include the first 20 lines in your report.
d) How many tokens appeared only once in the corpus?
e) From the list of tokens, extract only words, by excluding punctuation and other symbols, including forum-specific symbols, if any. Please pay attention to end of sentence dot (full stops). How many words did you find? List the top 20 most frequent words in your report, with their frequencies. What is **the type/token ratio** when you use only words (called lexical diversity)?
f) From the list of words, exclude stopwords. List the top 20 most frequent words and their frequencies in your report. You can use this list of stopwords (or any other that you consider adequate). Also compute **the type/token ratio** when you use only word tokens without stopwords (called lexical density)?
g) Compute all the pairs of two consecutive words (bigrams) (excluding stopwords and punctuation). List the most frequent 20 pairs and their frequencies in your report.

Note: You can split the text into sentences, but it is optional for this assignment.

You can use a lemmatizer if you want to put the words in their root forms. It is optional for this assignment. Please specify in your report if you did.

In your report, please fill in the following table, in addition to the answers to the above points:

# of tokens (b)	
# of types (b)	
type/token ratio (b)	
tokens appeared only once (d)	
# of words (excluding punctuation) (e)	
type/token ratio (excluding punctuation) (e)	
List the top 3 most frequent words and their frequencies (e)	
type/token ratio (excluding punctuation and stopwords) (f)	
List the top 3 most frequent words and their frequencies (excluding stopwords) (f)	
List the top 3 most frequent bigrams and their frequencies (g)	

Part 2: Evaluation of pre-trained sentence embedding models [50 points]

Word embeddings are dense representations of the meaning of words, build via neural language models. Sentence embeddings are dense representations of sentences, that can be composed by averaging the word vectors or sentence representations can be learned directly.
Chose at least 5 pre-trained sentence embeddings. If they have different parameters, you can experiment with them, but choose one for your report. Also make sure to include **doc2vec** (Le and Mikolov 2014, https://cs.stanford.edu/~quocle/paragraph_vector.pdf) and **SBERT** (Reimers and Gurevych, 2019, https://aclanthology.org/D19-1410/) (https://www.sbert.net/) in addition to other pre-trained language model that can represent sentences.

Use the dataset from the Semeval 2016-Task1 Semantic Textual Similarity (STS).
Use the test data STS Core (English Monolingual subtask) - test data with gold labels. Do not use the training data (we will use it in Assignment 2). Read more about the task at https://alt.qcri.org/semeval2016/task1/#

You can write your code or reuse existing code or tools (as long as you acknowledge them and extend them if needed).

Include in your report details about what sentence embeddings you chose (with what parameters and what mechanism is behind them), and add in the table below with the results of their evaluations on the above-mentioned benchmark dataset (if you used more than 5 embeddings, mention in the report but put the best 5 results in the table). Include in your report your system output for your best model for all the files in the test data.

We will have a mini-competition for the best score over the benchmark dataset for your best sentence embedding results. The evaluation score to report is the Pearson correlation between the score obtained by your model and the expected solution. The expected solution scores are numeric values between and 5 (from low similarity to high similarity). There is script that you can use to compute the correlations (included in the zip file for the test data).

Sentence Similarity

Datasets	SE1	SE2	SE3	SE4	SE5	Best Score
STS test data						

Submission instructions:

1. Prepare a report with written answers for the two parts. Summarize the methods that you implemented, any additional resources that you used, present the results that you obtained, and discuss them. Write the names and student numbers of the team members at the beginning of the report and explain how the tasks were divided.

2. Submit your report, results file (output.txt, tokens.txt, results files for part 2), and code electronically through the Virtual Campus or by email. Archive everything in a .zip file. Do not include the initial data files or any external tools or word/sentence embedding pre-trained models. Include a readme file explaining how to run each program that you implemented and how to use any external resources.