CSI5386: Natural Language Processing

Assignment 1
Corpus analysis and word embeddings

Due: Fri, Feb 7, 2020, 10pm

Note: This assignment should be done in groups of two students. Only one student from each group needs to submit via the Virtual campus, but to specify the names of the partners.

Part 1: Corpus processing: tokenization, and word counting [50 points]

Implement a word tokenizer for Twitter messages that splits the text of the messages 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. Some of them are adapted to social media texts, for example to tools for POS tagging mentioned in Part 2.

Use this corpus of 48401 Twitter messages as input to your tokenizer. The format is one Twitter message per line. Provide in your report the following information about the corpus:

a) Submit a file microblog2011_tokenized.txt with the tokenizer's output for the whole corpus. Include in your report the output for the first 20 sentences 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 100 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 Twitter specific symbols. How many words did you find? List the top 100 most frequent words in your report, with their frequencies. What is the type/token ratio when you use only word tokens (called lexical diversity)?
f) From the list of words, exclude stopwords. List the top 100 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 100 pairs and their frequencies in your report.

Part 2: Evaluation word embeddings [50 points]

Word embeddings are dense representations of the meaning of words, build via neural language models. Chose at least 8 pre-trained word embeddings, including CBOW, Skip-grams, GloVe, and a few others. If they have different parameters, you can experiment with them, but choose one for your report.

Use the code from https://github.com/kudkudak/word-embeddings-benchmarks to access 12 benchmark datasets and possibly the evaluation script. Select word similarity and word analogy questions (no categorization

datasets). Using other code or your own is ok to, as long as you use the required benchmark datasets. If you use the code mentioned above, be careful with the version of python you use. Version 3.5.5 seems to work, while other might conflict with one of the other libraries used). Also, one bracket seems to be missing in the last print in one of the examples.

Please use the following 12 datasets for evaluation:
Similarity tasks:
MTurk
MEN
WS353
Rubenstein and Goodenough
Rare Words
Multilingual SimLex999
SimLex999
TR9856

Analogy tasks:
MSR WordRep
Google_analogy
MSR
SEMEVAL 2012 Task 2

Include in your report details about what word embeddings you chose (with what parameters and what mechanism is behind them), and a table with their results of their evaluations on the above-mentioned benchmark datasets, in the order mentioned. Also include the average score over all the datasets. We will have a mini-competition, with chocolate prizes, for the best average score over the benchmark datasets for your best word embedding results.

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 (microblog2011_tokenized.txt), 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 embedding files. Include a readme file explaining how to run each program that you implemented and how to use any external resources.