CSI4107 - Assignment 2 README

Theo Van der Burgt 300019142
Ariane Poserio 300011641
Anshu Sharma 300011600

Work Division

Part 1 = Theo and Anshu
Part 2 = Ariane
Part 3 = Theo and Anshu
README = Theo and Ariane

Run File Instructions

On your windows machine, download the code from github and using the command prompt, change directory to the assignment folder.

Once inside the CSI4107_Assignment_2 folder, run the following command to load the assignment

Python A2_E1.py

Explanation of Algorithm and Samples

For this assignment the sent2vec Library was used to run a BERT model on the data of the vector based search used within assignment 1.

The first approach we took was to use the BERT model to find the most similar results to the query. To reduce the time required we used a binary classifier to only run bert on tweets that contain at least one word from the query.

Functionality

##################
STEP 1
##################
The modifications are made on top of the existing A2_E1.py file from assignment 1. This process is taking the corpus of tweets returned from the ranked retrieval of the inverted index approach used in assignment 1. A BERT model is then run on the already retrieved set of

tweets and then re-ranks the returned values based on the similarity according to the BERT model to try and increase the precision and mean average precision.

After running the bert reordering we got the following results:

```
end of Query 48
anshu@Anshus-MacBook-Pro CSI4107_Assignment_2 % trec_eval Relevance.txt Results_E1.txt
runid                   all     myTag
num_q                   all     49
num_ret                 all     48951
num_rel                 all     2640
num_rel_ret             all     1949
map                     all     0.0341
gm_map                  all     0.0161
Rprec                   all     0.0133
bpref                   all     0.1044
recip_rank              all     0.0191
iprec_at_recall_0.00    all     0.0682
iprec_at_recall_0.10    all     0.0655
iprec_at_recall_0.20    all     0.0644
iprec_at_recall_0.30    all     0.0618
iprec_at_recall_0.40    all     0.0588
iprec_at_recall_0.50    all     0.0558
iprec_at_recall_0.60    all     0.0481
iprec_at_recall_0.70    all     0.0440
iprec_at_recall_0.80    all     0.0367
iprec_at_recall_0.90    all     0.0160
iprec_at_recall_1.00    all     0.0011
P_5                     all     0.0041
P_10                    all     0.0082
P_15                    all     0.0095
P_20                    all     0.0102
P_30                    all     0.0109
P_100                   all     0.0194
P_200                   all     0.0253
P_500                   all     0.0338
P_1000                  all     0.0398
anshu@Anshus-MacBook-Pro CSI4107_Assignment_2 %
```

From this BERT approach we noticed that that precision at 1000 increased but overall the over measures suffered slightly as a result. Originally, our MAP score from A1 was at 0.1777, but after using the BERT approach, it decreased to 0.0341. This could be because we were using cosine similarity instead of the distance between vectors. Since the library we are using looks at distance instead of similarity, it could have changed our results.

##################
STEP 2
##################
The modifications are made under the “A2 HERE” section in the middle of the code for the existing A1.pu file from assignment 1. In this section, we are using Gensim's library; an open-source library for topic modelling and natural language processing using statistical machine learning. From this library, we are using Word2Vec to model and learn similar word

associations from a large corpus of text. The code also uses GloVe embeddings (Global Vectors for Word Representation), as an alternate method to create word embeddings.

In theory, the code should have taken the top most similar word for each token and then computed the cosine similarity using the word embeddings. However, running the model on each token took a lot of time to run; approximately 3+ hours per test. Had the code finished processing without errors, we would have liked to test this method by running trec_eval on the word embeddings in order to hopefully improve the MAP score.

##################
STEP 3
##################
The modifications are made on top of the existing A2_E3.py file from assignment 1. This process is taking the corpus of tweets and using boolean matching to reduce the size of the corpus to a list of tweets that contain at least one word in the query.
Once the size of the tweet list reduced, a BERT model is then run on the smaller set of tweets and then ranks the returned values based on the similarity according to the BERT model.

By running this model we got results but due to the results not properly formatting in a normalized form the results were not usable. This would not have stopped up had the timing of running it been different. The execution of this BERT model on all 49 queries took more than six hours causing us to need to use the results from another test we ran.