Wednesday, October 22, 2014

Project #1

Twitter Keyword Search

Write a keyword search engine that accepts a text file containing a number of tweets as input (each line is one tweet), and provides a simple user interface for querying the list of tweets against keywords. The system replies to keyword queries with the top 10 tweets that are most relevant to user query keyword(s).

As an example, assuming that the input file contains the following three tweets:
tw1 = <that moment while reading a student paper when the font changes>
tw2 = <followed by that moment when you Google the passages before and after the font changes>
tw3 = <just happened to me in this moment>

Given the query "the moment font changes", your search system must

i) find the list of tweets that contains at least one word in the given query, (in our example the list would be {*tw1, tw2, tw3*}),

ii) compute the similarity of the tweets in the list with the query keywords according to the number of times they pass in the tweets (in our example the scores would be {*tw1: 4, tw2: 5, tw3: 1*}), and

iii) select the top 10 most similar tweets
(in our example since we have less than 10 results, we will select all three of them)

iv) display the tweet texts to the querying user in the relevance order,
(in our example that would be:

1) followed by that moment when you Google the passages before and after the font changes
2) that moment while reading a student paper when the font changes
3) just happened to me in this moment

)

- You are free to use any data structure you like for storing the tweets, the intermediate data structures such as inverted indexes, etc… But you must define the complexity of each functionality in your search engine, e.g. if you are using an inverted index, building the inverted index, finding the inverted lists associated with the query keywords, taking the union of the inverted lists, computing a ranking, selecting the top 10 most relevant tweets, returning the tweet texts, etc… Efficient data structure selection and usage will get higher points.

Minimum Requirements [55%]

- Command line user interface that lets inserting raw tweet files, and querying.
- Ability to retrieve relevant tweets to the query keywords.
- Ability to display relevant tweets in correct importance order.

Possible Extensions [15%]

- Develop a GUI [15%]

Comparative Features [30%]

Your algorithms and data structures will be compared against others in the class and ranked according to:

- Speed of key operations