Benjamin Mao
Alexander Wilkins
Qicheng Hu

Final Report

Overview

For our project we used the twitter API to stream geolocated tweets which were stored in multiple files row by row. We then indexed these tweets using an API called PyLucene where we indexed the tweets according to the date it was created, the user's name, the hashtags, the tweet ID, and the tweet text. Lastly, we used mapreduce to do a measure of popularity using the HDFS.

Design

For the design of our project, we implemented everything in Python. For the twitter streaming and crawling, we used a file called tweetstreams.py which was responsible for creating the stream and listening for tweet objects. These tweet objects were then stored into text files in JSON format.In order to parse the text files, we used a json parser which read the file into a dictionary. We had to account for extended tweets as well since some tweets have text that are longer than the normal tweets and are a separate object.

For the indexing, we used an API called PyLucene which allowed us to create large indices in a relatively short amount of time. We read the text file into dictionaries and then passed them in as Lucene Fields. Since we had to do it locally on a virtual machine, memory was an issue so instead of one large 1 GB text file, we made multiple 100 MB text files. We then walked through the file directory and indexed each file in the folder. We indexed the date it was created, the username, the hashtags, the tweet ID, and the tweet text.

For the searching, we also used PyLucene. We created a loop in the searching function to continuously poll for user input. This input would then be parsed as a query and passed into Lucene's IndexSearcher. This then returned the "K" number of top hits which was specified by us. We outputted the tweet ID, username, text, date, and score. Our scoring metric was based on PyLucene's Similarity metric.

For the extension, we did a measure of popularity on the tweets using the HDFS. This mapreduce job counts the total number of retweet, quote, reply and favorite of each hashtag. However, the data should also be normalized by the size of file. So to account for that, we created a python script to normalize the file called ave.py.

Results

Our results after doing the measure of popularity on tweets showed that certain topics such as Christmas, during this time of the year, is very popular compared to other tweets. They had more retweets, quotes, and replies compared to other tweets we indexed. This makes sense as it is almost Christmas so this topic is bound to be more popular. Aside from that, we stored our results in a folder called index which has the 4 mapreduce results, and a normalized mapreduce result.

Collaboration

For collaboration, we split the project into 3 parts. Alexander Wilkins did the tweet listening and streaming. Benjamin Mao did the indexing which included building the index and searching the index. Qicheng Hu was responsible for the map reduce. However, since Alexander did not have access to the GitHub, he collaborated through Google Drive.