Analyzing Twitter data using Hadoop and MongoDB MapReduce

For this project we had to analyze a dataset of about 5 million twitter tweets (~9GB) collected using Twitter's datastream API in order count and plot the number of tweets mentioning each of the following (swedish) pronouns:

`“han”, “hon”, “den”, “det”, “denna”, “denne”, “hen”`

taking into account only **unique** tweets, by ignoring the retweets and also tried to find a case-insensitive solution.

Analysing Twitter data using Hadoop streaming and Python

For this task we are using `Python` and `Hadoop Map-Reduce`.

Related files (`Hadoop` directory)

- `mapper.py` :

  1. Define a set seen_ids to store the IDs of original tweets (not retweets).Parse each JSON file and check if a tweet is original.If it is, add its ID to seen_ids and process the tweet.
  2. For retweets, add the main tweet's ID to seen_ids if it doesn't already exist. Process the retweet as needed.
     3.For each tweet ID, maintain a map (id_map) that tracks word occurrences.`id_map` structure:
     `Key`: `Tweet ID`
     `Value` : `A map with word counts for that tweet`

- `reducer.py` : Nothing special, a simple reducer procedure that reduces the counts of the pronouns from input provided via STDIN.

- `plot.py` : Script that reads our `MapReduce` procedure's output and creates plot with the pronouns and their count.

Result

Word occurences

Analysing Twitter data using MongoDB and Python

For this task I loaded all the `json` files into a collection named `tweets` in a new database named `twitter`.

Then I leveraged `pyMongo` in order to get the work done.

Related files (`MongoDB` directory)

- `pymongo.py` : `python` file that loads the data into the collection
- `mapper.py` : Mapper script that connects to a MongoDB database to process the tweets. It tracks the counts of the words that we want to count in both original tweets and retweets. For each tweet, it checks if it's a retweet and if the retweet has already been processed. It counts the occurrences of predefined words in the tweet's text and stores these counts in a dictionary keyed by the tweet ID. Finally, it prints the word counts for each tweet.
- `reducer.py` : Reducer script processes input from standard input (STDIN) to count occurrences of the pronouns and updates their totals in a dictionary. It then outputs the final counts for each of these words.