Big Data Science

–

Twitter keyword extraction

Learning outcomes

- Developing and running a Java program on the HPC Prince cluster.
- Analyzing a collection of tweets to extract common themes and phrases.
- Implementing a rule-based or graph-based method for keyword extraction.

Getting started

For this homework, you will be using Java. You are allowed to use any external tools or libraries, but make sure to cite all of your sources.

Prerequisites

You will need access to the Prince cluster running on HPC.

Please contact Prof. Bari (*abari@nyu.edu*) if you are having trouble accessing your account.

Dataset

We will use the Sentiment140 dataset, which contains 1.6 million tweets. Each tweet is labeled as either positive (“4”) or negative (“0”). The dataset includes the following columns:

- `target`
- `ids`
- `date`
- `flag`
- `user`
- `text`

For this assignment, we are mostly interested in the tweet texts.

Download the data .csv file here:

https://drive.google.com/file/d/17ttbfbyRFE7KL_vV5LqMVoHMVrkiwQxU/view?usp=sharing

Tasks

HPC setup

After you download the file, your first task is to transfer the file to your `/scratch` directory (see here for a detailed overview of the file system on Prince). The easiest way is to use `scp`. For detailed instructions on how to transfer files from and to the Prince cluster, please refer to the official documentation: How to copy files from / to the HPC clusters.

Please note:

- When you first log into Prince, you end up in your home directory, e.g. `/home/netID`. Your scratch directory is located at `/scratch/netID`.
- When you log into Prince, you are initially accessing a *login* node. In order to perform any kind of computationally extensive operation, you need to request a *compute* node.

For example, you can request an interactive shell by using the following command:
`$ srun --mem=5GB --time=00:15:00 --cpus-per-task 1 --pty $SHELL`

For more details, please refer to any of the following tutorials on HPC:

- https://wikis.nyu.edu/display/NYUHPC/Slurm+Tutorial
- https://hpcc.usc.edu/gettingstarted/
- https://slurm.schedmd.com/quickstart.html

After you transferred the data to your `/scratch` directory on Prince, request an interactive shell session as described above and run the following commands:

```
$ ls -l
$ wc -l sentiment140.csv
$ head -n 5 sentiment140.csv
```

Attach a screenshot of your output.

Data exploration

Next, we would like you to proceed as follows:

1. Extract the raw tweet text by only using the “text” column.
2. Find and report the top 20 most frequent hashtags.
3. Find and report the top 20 most frequent @-mentions.
4. Depending on which of the two keyword extraction tasks you choose below, you might want to remove tokens from the raw tweet text that likely won’t be helpful for keyword extraction, such as emojis, URLs, hashtags, and so on.

We recommend using Stanford CoreNLP for text processing, but you are welcome to choose different Java tools or implement your own.

Attach the lists of most common hashtags and @-mentions. Briefly comment on your results, and report if you notice anything interesting.

N-gram analysis

Before we dive into more sophisticated methods of keyword extraction, we will start with a basic n-gram analysis. Please read the Wikipedia article on n-grams for a quick introduction.

In short, our goal is to find common phrases of length 1, 2, 3, and 4 that frequently appear together in our tweet collection. Feel free to reuse any code you wrote for previous homeworks in this class, just as long as it is your own work.

Report your 20 most frequent 1-grams, 2-grams, 3-grams, and 4-grams.

Keyword extraction

For this section, you can choose which of the following two tasks to work on ([1] OR [2]). You are of course welcome to work on both, but you are only required to complete one of them.

[1] Rule-based methods

For this question, we strongly encourage the use of Stanford CoreNLP. Your task is to leverage the various annotators to further explore common themes and patterns in the data.

For an interactive demo of CoreNLP, see: http://corenlp.run/

1. Run the part-of-speech (POS) tagger.

Redo your n-gram analysis, but only include nouns in your results. How do the results compare?

Next, think of at least three POS patterns that you think are suitable for keyword extraction. For example, a common pattern could be “adjective noun”, like “predictive analytics”. Collect the most frequent phrases that appear with each of your patterns, and report the top 10 for each.

2. Run the syntactic dependency parser.

Where applicable, collect sequences of (“nsubj”, “root”, “dobj”), i.e. the subject of a sentence, followed by the verb of the sentence, followed by the direct object of the sentence.

What are the most common occurrences? Feel free to try other combinations as well, for example “all adjectives that modify the subject” etc.

3. Run the named-entity tagger.

What are the most common named entities in the dataset? Briefly comment on your results.

4. Run your best keyword extraction method on the positive [“4”] and the negative [“0”] set of tweets independently.

Briefly analyze your results.

[2] TextRank

For this part, we will use TextRank, a graph-based ranking model that can be used for both text summarization and keyword extraction.

1. Read the original paper called “TextRank: Bringing Order into Texts” by Mihalcea and Tarau (2004). Briefly summarize the TextRank algorithm, and how it is applied to keyword extraction.

2. Find an existing Java library that implements TextRank. (Optionally, you are encouraged to implement the algorithm yourself for extra credit.)

3. Run the TextRank algorithm on the entire dataset. Use the best configuration reported in the paper. Report and briefly analyze your results.

4. Run the TextRank algorithm on the positive [“4”] as well as the negative [“0”] set of tweets independently. Briefly analyze your results.

5. Optionally, experiment with other hyperparameter settings and report your results.

Submission

Deliverables

Your submission should consist of the following:

- A plain text or .pdf file with all of your written answers and model outputs if applicable.
- Your code and a detailed README file on how to run it on Prince. If you're using the interactive shell, a complete set of screenshots is acceptable as well.

Grading

Deliverables	**Points**
HPC setup	5
Data exploration	5
N-gram analysis	10
Keyword extraction ([1] or [2])	30
(total)	**50**