Course: CS 696 Big Data Tools
Instructor: Roger Whitney
Students: Kevin D. O'Mara 816148418
Yu Tseng Chou 817486872

CS 696 Big Data Tools Final Project
LDA Topic Modeling on Tweets

Project Goal

The goal of the project was to explore topic modelling on Twitter tweets using unsupervised machine learning. We were curious what kinds of trends might emerge.

Our program utilizes the built-in Latent Dirichlet Allocation API in Spark to perform topic modeling on tweets that were collected in June 2017.

Data Source

We downloaded tweets from:
https://archive.org/details/archiveteam-twitter-stream-2017-06

The tar file holds tweets for the month of June, 2017 only. The file structure is [year]/[month]/[day]/[hour]/[minute].json.bz2 The minute.json.bz2 files are compressed. There are almost 50 GB of compressed tweets.

We wrote a script to extract tweets for a specific hour of a specific day. We chose to analyze only tweets from June 2nd, 8 pm in order to reduce the file size. This day holds over 700 MB of tweets.

For your convenience, we included a preprocessed text file "tweets.txt" with tweets from June 2nd, 8pm in the program folder. Detailed instructions on how to run the program are provided in the "How to Run" section.

References

For the program, we modified the example provided in the link below. We broke the file up into separate classes and cleaned up parts of the code. We also moved several parameters to command line input.
LDA Example: https://blog.knoldus.com/2016/10/08/spark-lda-clustering/

We used the following external libraries, included in the build.sbt file:
Stanford NLP API: https://nlp.stanford.edu/nlp/javadoc/javanlp/
Google Protobuf Java: https://developers.google.com/protocol-buffers/docs/javatutorial

How to Run

- Extract and Pre-Process Data
    a. (This step is optional. Only follow if you wish to analyze a different set of tweets.)
    b. Download the tweets. (almost 50 GB!)
    c. Modify `extractdata.sh`:
        i. Change `dayHourOfInterest` to the day and hour you are interested in analyzing
    d. Run `extractdata.sh`
    e. The decompressed and preprocessed data will be in a file named "tweets.txt"
- Perform Topic Modelling

```
sbt package

# Spin up cluster using start-master.sh and start-slave.sh
spark-submit target/scala-2.11/ldatweets_2.11-1.0.jar
[inputfilename]

# Optional flags
# -k, --topics => number of topics to discover, default: 5
# -n, --topicWords => number of words to associated with each
topic, default: 20
# -s, --stopWords  => path to custom stopwords file, one stopword
per line, ex:
s3n://komara.sdsu.big-data.LDATweets/customStopWords.txt

# Example with all options:
spark-submit target/scala-2.11/ldatweets_2.11-1.0.jar tweets.txt
-k 10 -n 20 -s customStopWords.txt
```

How it Works

See this blog for an excellent overview:
http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/

Analysis

The program outputs a specified number of topics. Each topic has a set of most-related words associated with it, each with a weight. We expected each topic to contain a set of distinct words.

Unfortunately, the topics collected had overlapping keywords and very little weight for each keyword. For example:

TOPIC 0	
premiosmtvmiaw	0.006702488077661699
2017	0.004399853107953849
friend	0.0030143302984218475
people	0.0030031152057011655
trump	0.0029267250824064734
lady	0.0027175592956569347
want	0.0026989798672401185
time	0.0026975356757376542
veranomtv	0.0025948040893535645
gaga	0.0024541047681937234

TOPIC 1	
premiosmtvmiaw	0.006852547195159178
2017	0.004290806983922516
friend	0.0031106893934284324
people	0.003033257841831964
trump	0.0029314924086936748
lady	0.002722620108482624
veranomtv	0.0027173089571527692
time	0.00265316021419681
want	0.0025831425012756027
mtvcrushreyes	0.0024910826570160222

At a quick glance you can see there almost all words are the same between these two topics.

We varied the number of topics to observe how the topics may change. Unfortunately, the same prominent topics continued to emerge.

We added a custom stop words file for users to add customized stop words to the program. We had to remove tweets that contain unicodes since those unicodes translate to foreign languages that were not meant to be broken down into characters.