Twitter Sentimental Analysis using Scala

Nicolaas Ruberg, Marco Acerbis, Salman Razzaq

Languages and Algorithms for Artificial Intelligence (91251)

Exam Session Summer 2020

Prof. Gianluigi Zavattaro

Abstract

This work presents a simple example to assess a Scala implementation running on an Amazon cluster environment. It also introduces some basic techniques of sentimental analysis over tweets.

One of these techniques consists of the use of a list of words that express positive, negative, or neutral concepts. And based on these words, we compute the sentiment of a tweet.

In our experiment, we looked for tweets that would contain the word “COVID” and provided the sentiment analysis of this set of tweets for the English, Italian, and Portuguese Languages.

Our implementation was developed in Scala, using Apache Spark, and deployed in an Amazon EMR cluster.

Keywords: Scala, Sentimental, Twitter, languages

Twitter Sentimental Analysis using Scala

In this work, we aim to summarize online opinions on Twitter, exploring the MapReduce features of Scala in a cluster environment. To explore Twitter, we look for a concept (e.g., COVID, Trump, Democracy), we mine topics where users have expressed their opinions and determine whether the views are positive or negative. For our summarization, we advocate the quantitative aspect of views based on the identification of word sentiment in the tweet.

Sentiment Analysis

Sentiment analysis refers to analyzing an opinion or feeling about something using data like text or images. It is a valuable tool to help companies in their decision-making process. For instance, if public sentiment towards a product is not so good, a company may try to modify the product or stop the production altogether to avoid any losses (Gauthan, 2019). With more and more people joining social media platforms, websites like Facebook and Twitter are good sources of public sentiment.

The basic idea behind the sentiment analysis consists of identifying a score (positive, negative, or neutral) for a post or publications. For the Twitter platform, the tweets are posts, and likewise, a sentimental score is assigned.

There are several Positive and Negative words Databases available such as SentiWordNet (Sebastian, 2006), WordStat (WordStat, 2020) and many others, (Davidova, 2017) presents a more detailed review. For our experiment we relied on a subset of the English words compiled by (Minqing, 2004), their work provides files with an opinion lexicon for Negative, Positive and Stop words, the latter being neutral words.

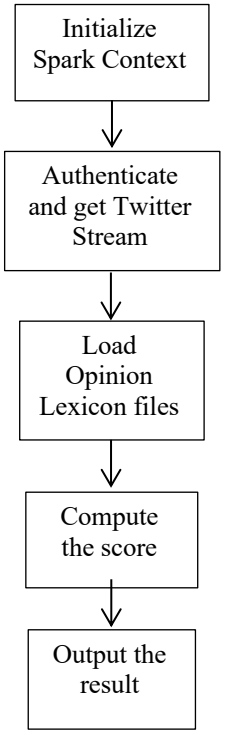
We generated the Italian and Portuguese languages' opinion lexicon translating the English lexicon. Most of the available API's for language translation is free up to the limit of

5000 characters (e.g., google translator, MyMemory, etc.), which was limited to our purpose. Just to mention, the positive words file has 2004 words with 19089 characters, which cannot be translated by those free APIs. To overcome this situation, we used the embedded translation function from Microsoft Word, which uses the online Microsoft Translator.

Experiment Description

The example provided by Stefano Baghino on Github (Stefano, 2017) inspired our implementation. And by using Spark and the Twitter extensions, we read live streams from Twitter and using the MapReduce programming model from Scala to select and filter the desired information.

In the following diagram, we provide the logic view of our implementation:



- Initialize Spark Context – Setup the initial parameters for Spark, whether the execution is local or in a cluster; also, we set the context name. The Spark Context

needs to be aware of where it is running, locally, or on the cluster. A local execution and a cluster execution need a different initialization; we detect and treat this situation. At the beginning of the program, we set the word files path to a local or S3 path.

- Authenticate and get Twitter streams – Tweets from Twitter are available via API for developers. The Twitter company requires registration to access its API, and after analysis, they grant access. Twitter gives a key, and a password to the developer users, and also every application accessing the social network needs a specific token and a password.
- Load Opinion and Lexicon file – depending on the spark context, we load the word files, this is determined based on where the application is running locally or in a cluster, accessing an S3 file system.
- Filter and compute the score – this step is where most of the transformations (map and reduce) takes place. In our code, for the English words, we gently describe those transformations. For the other languages, we applied the changes concisely in a single step. To illustrate, this is the code for the Italian score estimation:

```
// Let's check the Italian posts
val textAndNonNeutralScoreHastag_it: DStream[(Int, String)] =
  Italian tweets.
    map(_.getText).
    map(tweetText => (tweetText, wordsOf(tweetText))).
    mapValues(getMeaninfulSenteces(_,uselessWords_it)).
    filter { case (_, sentence) => sentence.length > 0 }.
    mapValues(sentence => computeScore(sentence, positiveWords_it, negativeWords_it)).
    filter { case (_, score) => score != 0 }.
    map { case (tweetText, score)  => (score,retrieveHashtag(hashtagOf(tweetText)))}.
    filter( { case (_, hashtag) => hashtag != " " })
```

- Output the result – the final step does the transformation to reduce the RDD data stream on time windows, in our implementation, this window has 3 minutes. During this transformation, we summarized the score for this set of tweets.

Experiment Setup

We deployed our Sentiment Analysis tool in the Amazon EMR cluster with a Master and two core nodes. All servers have the same configuration: one CPU with four virtual cores, 16 G bytes of Memory, and 64G bytes of storage.

AWS provides every node of the cluster with Amazon Linux version 2 for Intel CPUs. The application software installed on every node also includes the Spark 2.4.5 with Hadoop 2.8.5 YARN and Zeppelin 0.8.2. This configuration is set up on the creation of the cluster and is one of the default application selections provided by Amazon.

Access to the Master node in the cluster is granted by a public key, on the EMR console is possible to generate it. With this key, an ssh connection is enabled to the Master server. Also, on the cluster configuration, it is necessary to open access on TCP/IP port 22.

On the S3 storage, to upload our application and libraries, AWS provides a bucket, a persistent folder for an application. The results from our experiments also stored in S3.

Experiment Results

Our experiment was executed during a twelve-hour time window, starting on July 8th at 09:23 PM and ending on July 9th at 09:28 AM UTC. During this time window, we summarized the tweets containing the word COVID for the languages English, Italian, and Portuguese.

English Tweets

In the time window observed, there were 427.164 tweets. The overall sentiment, i.e., the sum of all sentiments on the period was -94.905, and the average sentiment for each tweet is nearly -0.22. The following graph presents the sentiment observed:

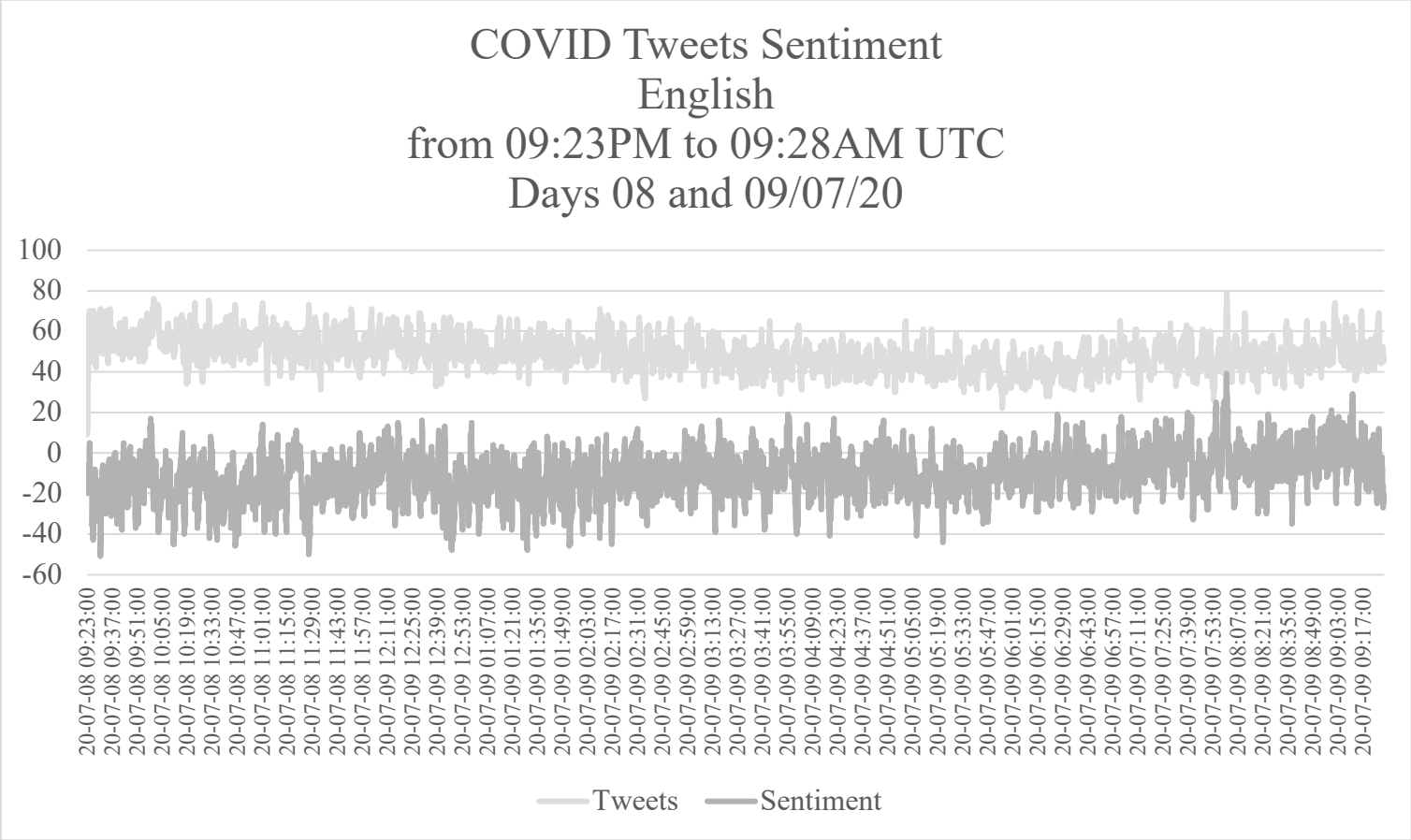


Figure 1- English Tweets – Y-axis expresses the number of tweets as well as the score of it. X-axis measurement time.

In the graph, we notice the fluctuation of sentiments over time in the tweets (darker gray line - sentiment line), with a maximum sentiment of 39 and a minimum of -51. The average number of tweets in the period analyzed was 49.

Italian tweets

In the time window observed, there were 8.136 tweets. The overall sentiment, i.e., the sum of all sentiments on the period was -1.661, and the average sentiment is nearly -0.2. The following graph presents the sentiment during the time frame:

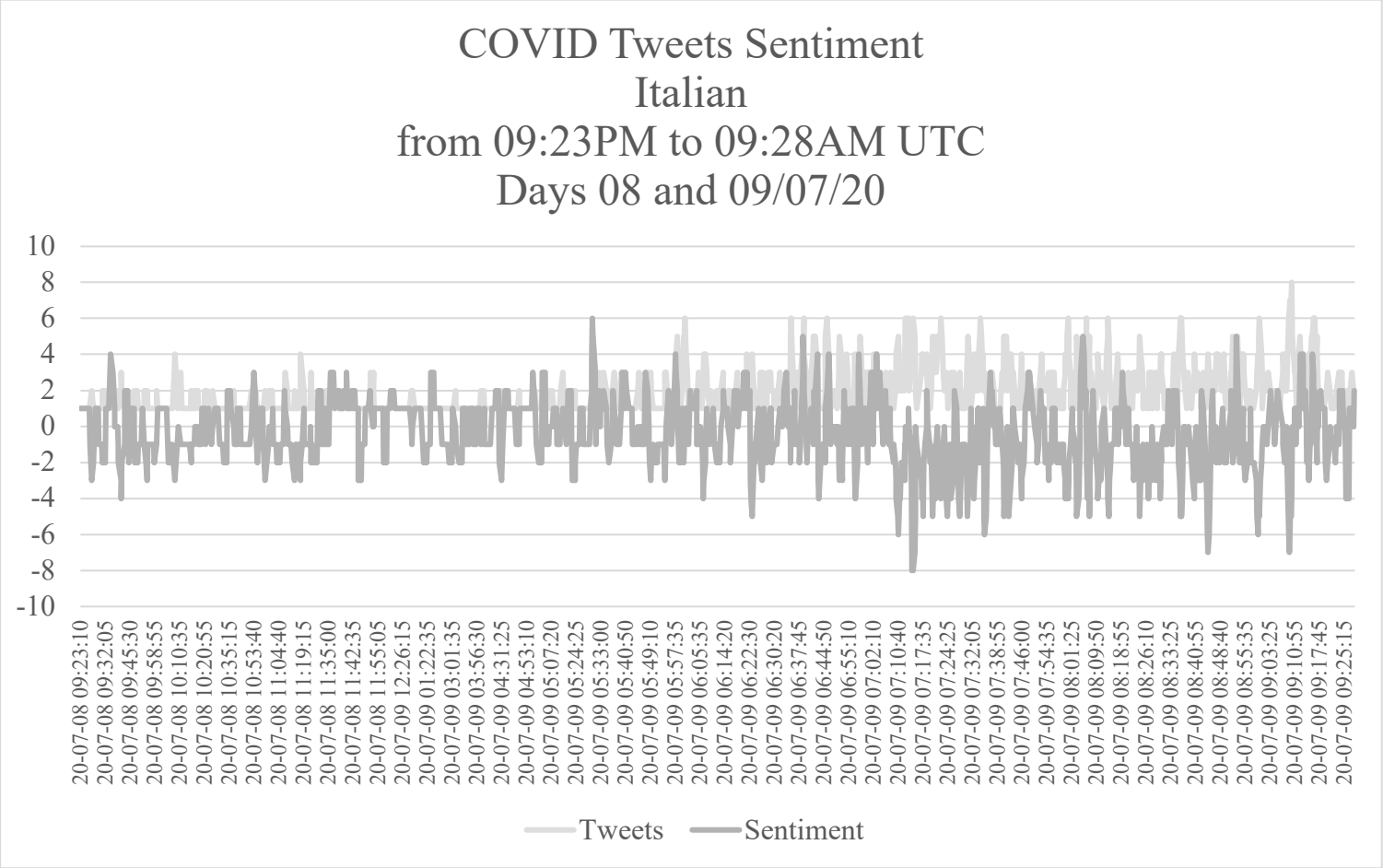


Figure 2- Italian Tweets -Y-axis expresses the number of tweets as well as the score of it. X-axis measurement time.

As in the English analysis, we can also observe the fluctuation of sentiments over time in the tweets (darker gray line - Sentiment line), with a maximum sentiment of 6 and a minimum of -8. The average number of tweets in the period analyzed was almost 2.

Portuguese tweets

In the time window observed, there were 2.963 tweets. The overall sentiment, i.e., the sum of all sentiments on the period was 977, and the average sentiment is nearly 3.03. The following graph presents the sentiment during the time frame:

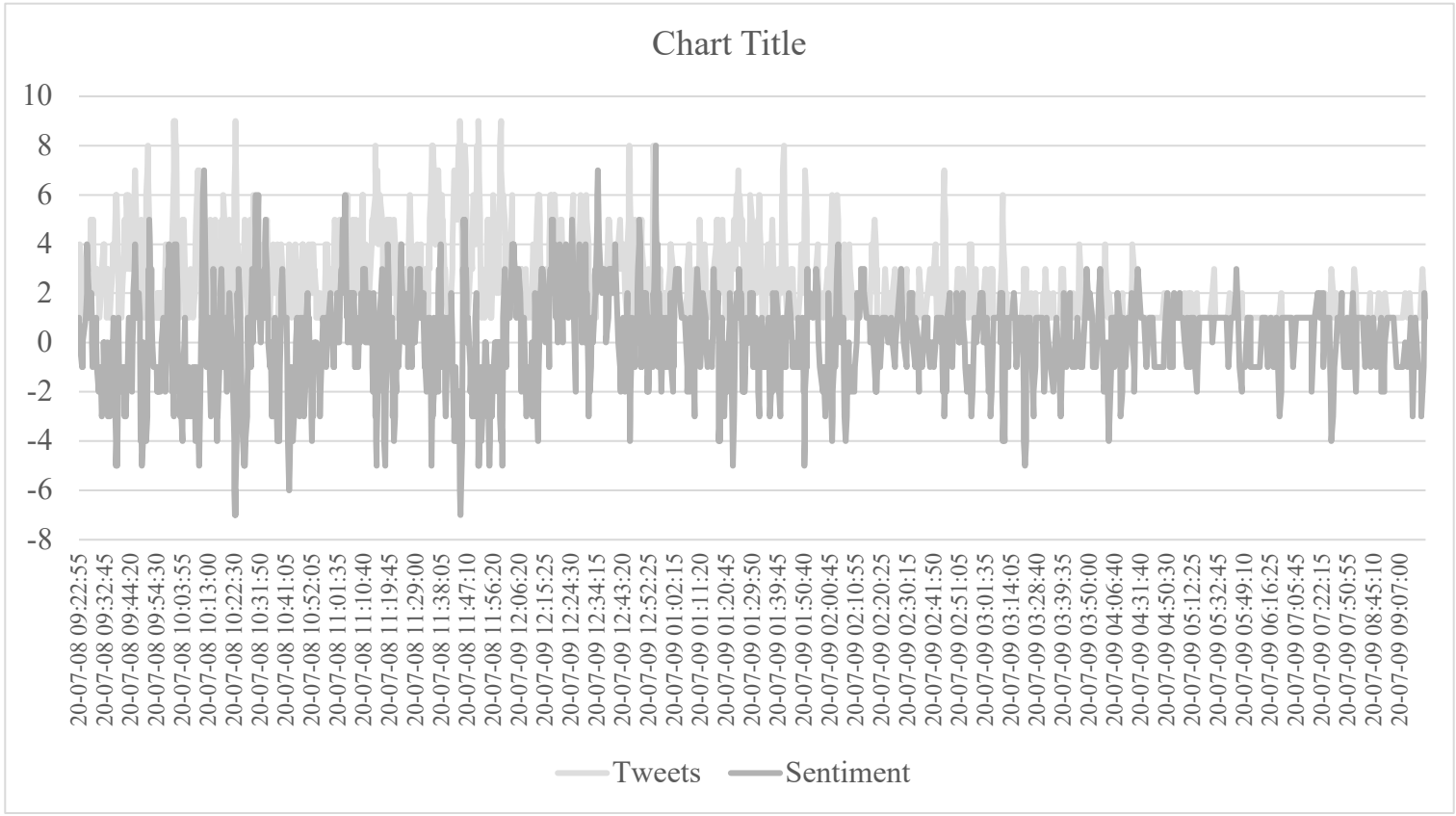


Figure 3 - Portuguese Tweets - Y-axis expresses the number of tweets as well as the score of it. X-axis measurement time.

As in the previous analysis, we can also observe the fluctuation of sentiments over time in the tweets (darker gray line - Sentiment line), but this time in a more positive window with a maximum sentiment of 12 and a minimum of -6. The average number of tweets in the period analyzed was 3.9.

Comments

Despite being a simple application, the setup and the deployment is complex. Now we comment on some of the challenges that we overcame during the project.

The Intellij IDE set up – the setup of Spark libraries, Scala compiler, and Spark environment on AWS is not an easy task. After a while, we came up with the following configuration:

SW/Library	Version
Scala	2.11.12
Spark-core	2.4.5
Spark-sql	2.4.5
Spark-streaming	2.4.5
Twitter4j-core	4.0.7
Twitter4j-stream	4.0.4
Spark-streaming-twitter	2.3.4

The setup of the AWS cluster in EMR requires some understanding of the AWS platform, and several steps are necessary to create the cluster: set up the AWS account, create the S3 buckets, create the EMR cluster, and managing the cluster in between experiments. During the cluster setup, it is essential to generate the public key for remote access as well as to open the cluster TCP/IP port 22. Another recommendation is to terminate or release the cluster after every experiment; otherwise, it continues consuming AWS credits, this happened during our tests, and we need to ask extra credits to continue the research.

The Scala/Spark setup does an excellent job of hiding the distributed aspects of the programming environment. Nevertheless, when handling files, the program must be aware of where it is running on. In our implementation, the word files and the generated output needed to be mindful if it was a local execution or a remote execution. We distinguish these two running environments based on a shell environment variable, and in the Scala program, we treated this aspect.

To illustrate it, this the code that sets the file path depending on the execution environment:

```
if (! sys.env.get("HOME").toString.contains("bia" ))
  dictPath = "s3://unibo-scala-twitter/"
else {
   dictPath = "./"
}
```

In a nutshell, in our experiment, we have to tackle several issues, such as the sentiment analysis problem, the Twitter API setup, the Scala IDE setup, the Spark library configuration; the AWS setup; and the cluster EMR setup and configuration. Those added up small amounts of complexity that made our experiment changeling.

Conclusions

In this work, we implemented a Twitter sentiment analysis exploring the MapReduce features of Scala in a cluster environment. Given a concept (e.g., COVID, Trump, Democracy, etc.), our implementation allows us to determine whether an opinion is positive or negative for English, Italian and Portuguese Languages in a time window.

To better explore the massive number of tweets, our experiment used the Amazon EMR cluster environment with one Master node and two core nodes.

We presented the results of the sentiment analysis for the concept COVID in a 12-hour window, for the three languages. It is crucial to notice the difference in the figures from the English, Italian and Portuguese tweets, respectively 427.614; 8.136; and 13835, which demonstrates that the English language is dominant in Twitter for the “COVID” topic. It is also interesting to observe the results for every language, with the tweets in the Italian language been more negative concerning the other languages.

Finally, we pointed out the challenges and difficulties that we encountered in the experiment. The complexity of every component in the project is low; the sum of them and the diversity of technologies increased the project complexity.

Bibliography

Sebastian, A. E. (2006). *SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining*. Retrieved from http://nmis.isti.cnr.it/sebastiani/Publications/LREC06.pdf.

WordStat. (2020). Retrieved from https://provalisresearch.com/products/content-analysis-software/wordstat-dictionary/sentiment-dictionaries/

Davidova, O. (2017, 07 13). *Sentiment Analysis Tools Overview, Part 1. Positive and Negative Words Databases*. Retrieved from Medium: https://medium.com/@datamonsters/sentiment-analysis-tools-overview-part-1-positive-and-negative-words-databases-ae35431a470c

Gauthan, R. (2019, 09 18). *Twitter Sentiment Analysis*. Retrieved from Medium.

Stefano, B. (2017, 12 14). *Spark Twitter Stream Example*. Retrieved from Github: https://github.com/stefanobaghino/spark-twitter-stream-example

Minqing, L. B. (2004). Mining and Summarizing Customer Reviews. *Proceedings of the ACM SIGKDD International Conference on Knowledge; Discovery and Data Mining (KDD-2004)* (pp. 22-25). Seattle: ACM.