Sentiment Analysis on Twitter : Effect of a Social Network

Abstract

Our purpose is to build a powerful platform for real-time data analysis of tweets on twitter trends. We also want to analyse all the tweets of 2017 based on a downloaded sample of data (average of 6To). All this data analysis will be accessible via a web interface that will be developed. We want to build a powerful system of sentiments analysis by making a database structure of tweets which is relevant about impacts and effects. The system should provide a faster way to execute Machine Learning methodologies behind data extracted from Twitter. Analysis news actuality by getting an analysis on actual trends with real stream data by building an efficient web interface to get results easily and build a system without false accounts and keep a control on data continuously.

# Introduction

The main subject is Sentiment Analysis on Twitter, a microblogging platform where people can easily share their thought on anything and their habits too. We have a lot of publications on sentiment analysis but not so much research about impacts and their effect on society. The maximum characters are 140 which can be a good thing for the process of analysis because it will make it faster in a way to perform on small messages but in the other hand we should pay attention on accuracy of results. Event it’s an enormously continuous stream of data, Twitter is a good extra sentiment though an online community. Therefore, how to optimize all those streaming data and build a web interface for users who want to get data. Our project will use many methodologies from Machine Learning like unsupervised methods to make a classification of sentiments, and supervised method to predicate psychological profile. Finally, one big step will be and efficient system about control of massive data incoming, a check on false account and spam messages that will destroy our results for example.

Jeffrey Zeldman “The best way to engage honestly with the marketplace via Twitter is to never use the words “engage,” “honesty,” or “marketplace.”

In this study, we introduce readers to the problems of Data Processing and Cloud Computation, which have been rapidly developing over the last decade. The rest of this document is organized as follows. In Sect. 2, we provide a specific view of Tweet format. Development, problems, definitions and main trends of this area are described in Sect 3. We analyze in Sect 6, and discuss in Sect 7 about several problems we have been through.

Finally, we talk about others idea and features that were not implemented due to a lack of time in Sect 8.

# Related Work

To begin, ye can refer to this article [cite palpatine] because we can relate that it is a point of start, it gives us a theoretical review on the development of Sentiment Analysis.

The interested reader can also refer to previous surveys in the area, like [cite] and [cite], that helped us on machine learning techniques. As an active research field that has emerged for a long time now, sentiment analysis is now been greatly implement but it is also with a cost (for example, IBM Watson Tone Analyzer). Sentiment analysis is a discipline that extracts people’s feelings, opinions, thoughts and behaviors from user’s text data using Natural Language Processing (NLP) methods.

For methodologies on preprocessing, and feature generation, we based our work on [deep learning]. Then we can also cite [chinois publication] for a reason, they removed numbers from theirs tweet thinking that in general, numbers are of no use when measuring sentiment and are removed from tweets to refine the tweet content but we wanted to keep them thinking the contrary.

Based on 36 million tweets collected from Twitter, Wang et al. proposed a real-time sentiment analysis system for classification of political tweets during 2012 US presidential elections. Their model achieved 59% accuracy in predicting the sentiments of political tweets (Wang et al., 2012).

We had over 800 million tweets in English and all that data to analyze where possible just because of the MapReduce Model [cite les grecs] in majority.

# Methodology

## Preprocessing (Data clean)

## Feature Generation

The data for this task consists of tweets across various domains, classified into four emotions: joy, sadness, anger and fear. The training data additionally carries a real-valued score between 0 and 1 per tweet, indicating the degree of the emotion (that the tweet is classified as) the present in the tweet.

# Cloud Computing

Cloud computing was used for the project, the trend today is machine learning, which is a form of artificial intelligence that uses algorithms to learn from data. These systems build models from incoming transactional data, then find patterns in that data to make predictions. These predictions can be a simple as providing a recommendation to a shopper on an e-commerce website or as complex as determining if a brand of automobile should be retired. As with their learning-system forebears, the overhead of machine-learning systems is typically huge. But today we have the option to place these systems in the cloud. Amazon Web Services, for example, supports machine learning using AWS's algorithms to read native AWS data (such as RDS, Redshift, and S3). Google has supported predictive analysts for some time with its Google Prediction API, and Microsoft provides an Azure machine-learning service. The ability to predict the future for both tactical and strategic purposes has eluded us because of prohibitive resource requirements. But today, thanks to the cloud for machine learning as a service, you can apply this technology far and wide on all that data enterprises have been collecting.

## AWS (Amazon Web Service)

## MapReduce Model and Spark Framework

## Conception of the Database for the Web Interface

The World Wide Web (Web, for short), is a distributed information system based on hypertext. Web interfaces to databases have become very important. After outlining several reasons for interfacing databases with the Web, we provide an overview of Web Technology. We then outline techniques for building Web interfaces to databases.

# Implementation

We have used the following technologies for many reasons. Hadoop assumes that conventional approaches (consisting of developing ever more powerful centralized systems) have technical and financial limitations. The development of distributed systems consisting of machines or nodes, relatively affordable (commodity hardware) and scaling out is an alternative from a technical and financial point of view A distributed system comprising tens, hundreds or thousands of nodes will regularly be confronted with hardware and / or software failures. Google has developed the Google File System (GFS), ancestor of the Hadoop Disrelated File System (HDFS) and The MapReduce Approach. MapReduce is a programming model designed specifically to read, milk and write very large volumes of data. A Hadoop program usually implements both map tasks and reduce tasks.

Hadoop is particularly effective for dealing with problems that have one or more of the following characteristics: Volume of data to store or process very important. Need to perform processing on all data (batch rather than transactional, therefore). Heterogeneous data in terms of origin, structure, and format (JSON) Execute the tasks of a Hadoop job in parallel, without a pre-established order. A Hadoop cluster is made up of tens, hundreds, or thousands of nodes. It is the addition of the storage and processing capacities of each of these nodes which makes it possible to offer a storage space and a computing power yet to handle data volumes of several To or Po. To improve the performance of a read / write cluster, Hadoop’s file management system, HDFS, writes and reads files in blocks of 64 MB or 128 MB. Working on such large blocks maximizes data transfer rates by limiting search time on hard drives (seek time). // Input file graph and block MapReduce is a programming model designed specifically to read, process and write very large volumes of data. A Hadoop program usually implements both map tasks and reduce tasks. A Hadoop program is usually divided into three parts: The driver, which runs on a client machine, is responsible for configuring the job and submitting it for execution. The map is responsible for reading and processing data stored on disk. The reducer is responsible for consolidating the results from the map and write them on disk.

# Results

# image

# Discussion

We now turn our attention to the following interesting question: whether the subjective data that exist on the web carry useful information. Information can be thought of as data that reduce our uncertainty about some subject. According to this view, the diversity and pluralism of information on different topics can have a rather negative role. It is well understood, that true knowledge is being described by facts, rather than subjective opinions. However, this diversity in opinions, when analyzed, may deliver new information and contribute to the overall knowledge of a subject matter. This is especially true when the object of our study is the attitude of people. In this case, opinion native data can be useful to uncover the distribution of sentiments across time, or different groups of people.

The term data mining refers loosely to process of semiautomaticcally analyzing large databases to find useful patterns. Like knowledge discovery in artificial intelligence (also called machine learning) or statistical analysis, we use data mining to discover rules and patterns from data. However, data mining differs from machine learning and statistics in that it deals with large volumes of data, stored primarily on disk. That is, data mining deals with “knowledge discovery in databases”. Some types of knowledge discovered from a database can be represented by a set of rules. The following is an example of a rule, stated informally: “Donald Trump with his totals of retweets incomes are greater than the average with the most sadly effects on users”. Of course, such riles are not universally true, and have degrees of “support” and “confidence”, as we shall see. Other types of knowledge are represented by equations relating different variables to each other, or by other mechanisms for predicting outcomes when the values of some variables are known. There are a variety of possible types of patterns that may be useful, and different techniques are used to find different types of patterns. Usually there is a manual component to data mining, consisting of preprocessing data to a form acceptable to the algorithms and postprocessing of discovered patterns. For this reason, data mining is really a semiautomatic process in real life. The mode widely used applications are those that requires some sort of prediction. In our case, we want to predict emotions and sentiments, then a psychological profile. Prediction is one of the most important types of data mining. We outline what is classification, study techniques for building one type of classifiers, called decision-tree classifiers, and then study other predication techniques. Abstractly, the classification problem is this: Given that user belong to the archive, and given his tweet. We use a given instances (called training instances) of items along with the classes to which they belong, the problem is to predict the class (in our study it is a sentiment or an emotion) to which a new item belongs.

# Application area: Psychological prediction

POMS (Profile of Mood States) is a psychological rating scale used for calculating the mood state score, the result depends on the values of 65 adjectives. For our project we reorganized the adjectives in two ways, first in 3 categories: Positive, Negative and Neutral, second way in 5 categories: Joy, Surprised, Fear, Angry, Sadness. More about the way to calculate POMS adjectives, click here.

# Discussion

Hence, the difference between Apache Storm and Spark Streaming shows that Apache Storm is a solution for real-time stream processing. But Storm is very complex for developers to develop applications. There are very limited resources available in the market for it. Storm can solve only one type of problem i.e Stream processing. But the industry needs a generalized solution which can solve all the types of problems. For example Batch processing, stream processing interactive processing as well as iterative processing. Here Apache Spark comes into limelight which is a general purpose computation engine. It can handle any type of problem. Apart from this Apache Spark is much too easy for developers and can integrate very well with Hadoop.

Apache Spark is an in-memory distributed data analysis platform– primarily targeted at speeding up batch analysis jobs, iterative machine learning jobs, interactive query and graph processing. One of Spark’s primary distinctions is its use of RDDs or Resilient Distributed Datasets. RDDs are great for pipelining parallel operators for computation and are, by definition, immutable, which allows Spark a unique form of fault tolerance based on lineage information. If you are interested in, for example, executing a Hadoop MapReduce job much faster, Spark is a great option (although memory requirements must be considered). Apache Storm is focused on stream processing or what some call complex event processing. Storm implements a fault tolerant method for performing a computation or pipelining multiple computations on an event as it flows into a system. One might use Storm to transform unstructured data as it flows into a system into a desired format. Storm and Spark are focused on fairly different use cases. The more “apples-to-apples” comparison would be between Storm Trident and Spark Streaming. Since Spark’s RDDs are inherently immutable, Spark Streaming implements a method for “batching” incoming updates in user-defined time intervals that get transformed into their own RDDs. Spark’s parallel operators can then perform computations on these RDDs. This is different from Storm which deals with each event individually. One key difference between these two technologies is that Spark performs Data-Parallel computationswhile Storm performs. Task-Parallel computations. Either design makes tradeoffs that are worth knowing. I would suggest checking out these links.

# Conclusion and Future Work

In this report we have presented a sentiment analysis tool on a Web interface, in one hand we used data from an archive end in the other we used real time stream analysis. Due to the absence of labelled data we couldn’t discuss the accuracies of the two enhancements. In the future, we plan to use the as feedback mechanism to classify new tweets.

# References

We are grateful to the following people for resources, discussions and suggestions: Jason Scott (Archivist) and Diana Yuan (Co-Founder & Vice President, Talent & Operations from Indico).