Distributed Sentiment Analysis for identifying hate speech

Rahul Ranjan

Computer Science & Engineering
IIIT Senapati, Manipur Imphal, India
rahul@iiitmanipur.ac.in

Shubham Raj

Computer Science & Engineering
IIIT Senapati, Manipur Imphal, India
s.raj@iiitmanipur.ac.in

Abstract—This document describes a work and development of a distributed sentiment analysis system being developed for industrial usage. This project seeks to leverage advances in open source development in natural language processing and understanding, distributed databases, and distributed processing frameworks, as much as possible to advance the development process and to utilize the most recent technology The importance of detecting and moderating hate speech is evident from the strong connec- tion between hate speech and actual hate crimes (Watch, 2014). Sites such as Twitter and Facebook have been seeking to actively combat hate speech (Lomas, 2015). Most recently, Facebook announced that they would seek to combat racism and xenopho- bia aimed at refugees (Moulson, 2016). Hate speech in the form of racist and sexist remarks are a common occurrence on social media.

Index Terms—Distributed System, Sentiment Analysis, Hadoop, Big data, Cloud Computing, Logistic Regression.

I. Introduction

The goal of this project is to develop relatively low cost, scalable, and efficient software tools for industrial usage of sentiment analysis. In this paper, we begin with a review of candidate tools and determine which ones to use. Then we describe our initial implementation followed by a description of our selection of an appropriate social media source. We then analyze the results and discuss revision of the system based on initial performance and problems. Finally, we present future directions and conclusions. Hate speech is an unfortunately common occur- rence on the Internet (Eadicicco, 2014; Kettrey and Laster, 2014) and in some cases culminates in severe threats to individuals. Social media sites therefore face the problem of identifying and cen- soring problematic posts (Moulson, 2016) while weighing the right to freedom of speech. The importance of detecting and moderating hate speech is evident from the strong connec- tion between hate speech and actual hate crimes (Watch, 2014). Early identification of users pro- moting hate speech could enable outreach pro- grams that attempt to prevent an escalation from speech to action.

We have determined we need four components for our development environment. 1) A distributed database system for fast retrieval and scalability of large data sources. Distributed databases also provide high availability, where if a node goes down, the system will still process transactions using the remaining working nodes it has available. 2) A programming language and distributed computing framework which facilitates scalability of concurrent sentiment analysis on different messages. 3) A natural language processing system to process the social media messages and allow us to develop rules to customize the system to perform sentiment analysis using our institutional knowledge. 4) A social media source which will allow us to sample a large audience and determine their sentiments related to our products.

II. Methodology

Following Steps have been Followed to achieve the required result -

- The data has been downloaded from kaggle and used for training and testing. The data is properly labelled so that it can be used for sentiment analysis.
- There after our focus was on creation of lexicon-builder using the hadoop for storage and analysis of large amount of data in order to make the whole system real time applicable. After hadoop part is programmed in Scala, a java oriented language which has good application in NLP and standford library.
- In the NLP section for the sentiment anlysis purpose which is done in python-environment because of its compatibility with NLTK.
- In the sentiment analysis part we have successfuly achieved the designed result after processing the data properly before applying sentiment classifier (here Logistic Regression).
- We have obtained the desired result with a good accuracy of the system but the data is currently limited to 33000 tweets approx. We will dicuss in later part of this paper more about its implementation.

Let's Discuss the methodology briefly -

III. Data

Twitter is a micro-blogging tool where users opt-in to receive and send extremely brief content – or tweets – with others. The “tweet,” or messages used in Twitter, are limited to 140 characters. This creates a wonderful practice of being concise with the message you would like to convey.The tweets also contains images, URLs, videos, gif,etc. Tweets also contains hashtags, which are words that capture the subject of the tweet and they are prefixed by ’#’ character.The hashtags may carry sentiments or emmotions (#sarcasm), actions (#bullying), climate (#poor) and many more.There are other symbols as well like ’’ which symbolises usernames or handles, retweet (‘rt’) is a tweet by a user X that has been shared by user Y to all of Y’s followers,a heart shape logo represents a ’like’ on twitter.
The dataset is downloaded from kaggle [?] for the analysis purpose which is accurately labelled with 0 stands for no racist or hatred words/hashtags whereas 1 stands for racist/hatred tweets/hashtags.The training (80%) and testing(20%) data consists of hashtags of hate-speech, trolling and cyberbullying.

IV. Data Manupulation

The hashtags were extracted from the tweets stored in training and testing dataset file and on proper analysis it gives various informations about the various sentiment words like racists #black, #poor, #backwards, #saveErnakulam, etc. Text Processing is a very important task to obtain less noisy and better result.It involves -

- Removing Stopwords - Stop words are words which are filtered out before or after processing of natural language data (here tweets).The english stopwords can be used from nltk, it provides stopwords (eg-’am’,’is’, ’are’,’but’, ’shall’, ’by’ etc.).These words doesn’t have any significance on the desired results.
- Remove numbers - The tweetid is not required as it doesnot serve any purpose in the desired output , so it’s better to remove these words.
- Remove short-length words - These are those words which have length less than 3 (len<3) are supposed to be remove as these are irrelevant.
- Remove URLs - The URLs present in the tweets are supposed to be removed as it doesnot have any role in the desired results.
- Stemming - Stemming usually refers to a process that chops off the ends of words in order to obtain a root form, and often includes the removal of derivational affixes and inflectional affixes.E.g.- am, is, are => be ;boy’s => boy,etc.
- Lematization - Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.E.g.-’walk’, ’walked’, ’walks’, ’walking’=> walk, etc.
- Removing Punctuations - Punctuation marks are irrelevant for desired output of the system so it is better to remove them.To remove punctuation marks it is better to use regular expression. E.g.- ’@,?,!,etc’

V. Approach & Implementation

After proper preprocessing, we have counted, which hashtag was used most frequently , counted its frequency and designed a plot using matplotlib for pictorial representation of data. This helps in anlysing the topic and references towards which the bullying or hatred speech is talking about, who are involved in this kind of activities like we have obtained #Trump occured most number of times as we are not only extracting hatred hashtags but also all the hashtags present in racist or hastred hashtags containing tweets. Following figure shows the maximum hatred hashtags plotting as per their countings as —

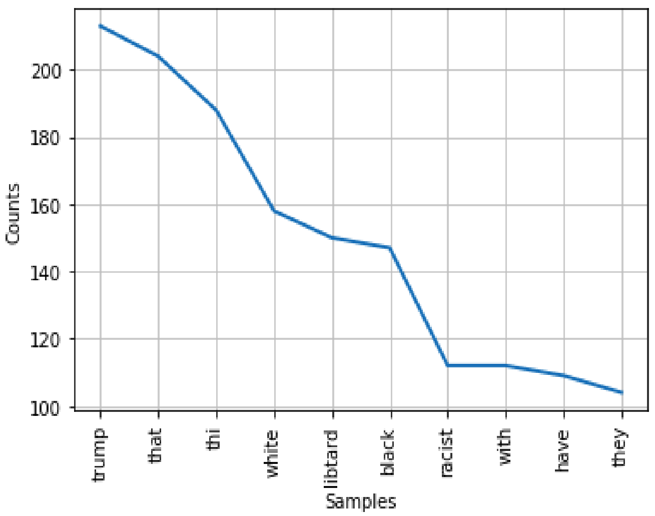

Fig. 1. Top 10 #hashtags used (Count vs hashtags)

It was a tough task to deal with such data, So these data required classifications of tweets so that proper model can be prepared. Before using Vectorizer it is better to add each hashtags / tweets to the data frame ,and then we have used TF-IDF to vectorize the tweets.The TF-IDF Vectorizer converts the tweets into vectors when each tweets are processed and appended to a list and this list was provided to the TF-IDF Vectorizer.Each value in the vector depends on count of words or a term appears in the tweet (TF) and on how rare it is amongst all tweets/documents (IDF).More details for TF-IDF can found here.

	twt1	twt2	twt3	twt4	twt100	twt(n)
T1	10	0	1	0	...	0
T5	0	0	0	0	...	0
T...	...	...	...	...	...	...
T(m)	0	1	8	0	...	3

where:

T_i = term (i = 1 to m) ; twt_j =tweets (j = 1 to n)

Fig.2.[TF-IDF Matrix]

VI. Experiments & Results

The TF-IDF values are calculated using 1-3 n-grams, meaning phrases with 1,2,3 words are used to compute frequencies. Given the extensive open source libraries developed in the Java programming language, we have decided to utilize a language that can access those libraries directly. Also, given the support for distributed processing, code reduction, and ability to use Java libraries, we have decided to investigate the Scala programming language. Akka(Typesafe, 2015) and Spark(Apache Software Foundation, 2015) are distributed processing libraries/frameworks which are written in Scala and provide interfaces to Scala which provide highlevel support for distributed processing.

Scala is a hybrid functional/object-oriented programming language. It runs within the Java virtual machine (JVM) and can therefore utilize all Java libraries which are extensive. The improvements of Scala over Java are well-documented (Eder, 2014). One of the biggest advantages is the reduced amount of code that needs to be written using Scala. Also, the support for distributed processing via functional programming is well-established(Milewski, 2014). Hadoop HDFS and HBase (Apache Software Foundation, 2010) were the initial systems that we have worked with to store and retrieve social media text and related metadata and sentiment scores. HDFS/HBase scales well and is mature and stable. Also, the importation of data into HDFS/HBase can be done from a CVS file using a map/reduce utility importTsv. Recently, however, we have been migrating to Cassandra, because it has built-in high availability which we think will be critical as we opt for storage on large clusters of servers which will increase the probability of a single node failure. We reviewed NLP systems and narrowed our search to NLTK(Steven Bird et. al., 2009) and Stanford CoreNLP(Manning et al., 2014). NLTK sentiment analysis uses Naive Bayes classifiers which is a well known and effective machine learning technique. However, two things that discouraged us from using NLTK were 1) a desire to utilize deep learning technologies which have been shown to be a big advancement for machine learning, and 2) distributed processing in the native language of NLTK, Python, is not as well supported as in Java and/or Scala. Instead, we chose the Stanford CoreNLP system, because it's sentiment analysis system uses recurrent neural networks which is a new deep learning technique, it recently added multi-threaded training, and it is actively maintained. Rather than write our own NLP system or purchase one, we preferred the rapid advancement we saw in the Stanford CoreNLP system, and the ability to access the code base as desired.We then determined that we needed distributed processing for our industrial social media sentiment analysis system, because we expected a large number of users and messages, so if we wanted to process them in a reasonable amount of time, we needed to distribute the workload across a large cluster of machines. Both the Spark and Akka frameworks for distributed processing are native to Scala. Spark utilizes Akka to support seamless distributed collections (RDDs) automatically, therefore eliminating the need of the application developer to implement complex message passing or other distributed mechanisms to make use of the power of distributed computing. Since we have used the Spark cluster computing framework on other projects (Burdorf, 2015), we were motivated to use it in association with the sentiment analysis annotator in the Stanford Core NLP system. However our initial experiments showed some interface issues. For example, the Stanford Core NLP Java class isn't serializable, so it doesn't work well within the Spark ecosystem. With the Spark experience as a backdrop, we began investigating the use of Akka toolkit to handle concurrency and distributed processing. Spark is built on top of the Akka framework, so it clearly has the capabilities of providing the concurrency and performance that we need for this project. Akka is an Actors-based distributed processing toolkit that is based on Carl Hewitt's Actor-based distributed computing model (Hewitt et al., 1973). An actor is a fundamental processing unit which supports processing, storage, and communication. Actors can run concurrently on multi-core systems and distributed over a cluster of multi-core systems.

The proposed model can be understood as per following diagram -

A. Logistic Regression

- We used Logistic Regression over SVM because it has fast training time as compared to SVM and its ability to adjust model with new data, also it have very competitive accuracy compared with SVM.
- Prior to the usage of the Stanford CoreNLP system sentiment analysis annotator, one must develop a training set to teach the system what phrases to use to indicate levels of sentiment with social media mined messages.There is then a separate training mechanism

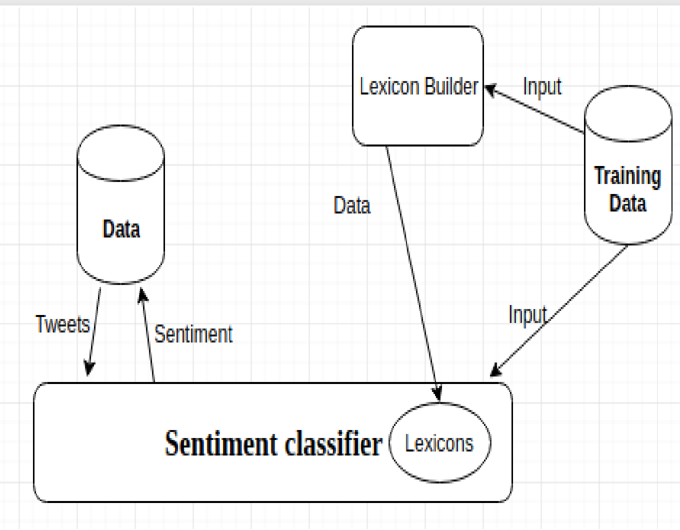

Fig. 2. Architecture of large-scale distributed system for real time twitter sentiment analysis

which reads the training set and generates an internal database using Logistic Regression.

- Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.
  For example,
  To predict whether an email is spam (1) or (0)
  Whether the tumor is malignant (1) or not (0)
  Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

Types of Logistic Regrassion -

1) Binary Logistic Regression
2) Multinomial Logistic Regression
3) Ordinal Logistic Regression

Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.
We have obtained the following result by using Logistic Regression on the dataset.

We obtained the following results as shown in figure shows the desired result which we wanted as they are accurately measured after proper pre-processing -

- Case 1 -When random_ state=None & test_ size=0.2

Iteration	Precision	Recall	F-Measure	Accuracy
1	60.23%	45.98%	51.01%	94.32%
2	62.82%	47.78%	54.17%	94.30%
3	63.12%	.49.43%	55.67%	94.30%
4	64.67%	49.21%	55.78%	94.61%
5	67.03%	53.09%	59.56%	95.22%

- Case 2 -When random_ state=10 & test_ size=0.2

Iteration	Precision	Recall	F-Measure	Accuracy
1	66.67%	51.93%	57.78%	94.54%

VII. Conclusion and Future works

It is clear from the above analysis that how powerful a social media is and it can be used very widely to help the Mankind and other species.Social media can be harnessed to great effect in times when people need support.Some of the steps which are taken in this Project are also adopted by twitter itself to help surrounding community in fighting against situations of cyber bullying,online harashments, sarcasm detection,detecting hatred speeches,etc etc. As twitter has initiated the practise of creating hashtags specific to individual crises to index tweets easily.
The Future tasks that will be Our goal is to produce a system that can process large amounts of social media messages and store sentiment to databases that can help us better understand the hate speech. We will focus on making it more accurate and useful by applying some Neural Network concepts in this project in future.To make this system more powerful & useful, by implementing some technique that can detect non-hashtag words that are relevant for analysis.Using deep learning, the next task will be to identify short conversation messages tweets.

The power of social media will continue to be researched and newer applications will continue to be built to harness its power. As we are far from making the system real time and will be our future task to implement.This system can be further used to identify the irony and sarcasm for Hindi-English code mixed dataset

Acknowledgment

The authors are grateful to Indian Institute of Information Technology Senapati (Manipur) and the Computer Science & Engineering Department for their support and encouragement.

VIII. Bibliography

- Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter Zeerak Waseem University of Copenhagen Copenhagen, Denmark csp265alumni.ku.dk Dirk Hovy University of Copenhagen Copenhagen, Denmark dirk.hovy hum.ku.dk

- Michael I. Jordan, Journal of Machine Learning Research 3 (2003) 993-1022

- Diana Abbas. 2015. What's in a location. https://www.youtube.com/watch?v=GNlDO9Lt8J8, October. Talk at Twitter Flight 2015. Seen on Jan 17th 2016.

- BBC. 2015. Facebook, google and twitter agree german hate speech deal. http://www.bbc.com/news/world-europe-35105003. Accessed on 26/11/2016.

- Kaggle-for downloading datasets to prepare the whole system.

- Yunfei Chen, Lanbo Zhang, Aaron Michelony, and Yi Zhang. 2012b. 4is of social bully filtering: Identity, inference, influence, and intervention. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pages 2677–2679, New York, NY, USA. ACM.

- Stephan Gouws, Donald Metzler, Congxing Cai, and Eduard Hovy. 2011. Contextual bearing on linguistic variation in social media. In Proceedings of the Workshop on Languages in Social Media, LSM '11, pages 20–29, Stroudsburg, PA, USA. Association for Computational Linguistics.

- Overleaf, https://www.overleaf.com (accessed Aug 31, 2018)

- Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. 2015. Detect- ing racism in dutch social media posts, 2015/12/18.

- Oracle+DataScience.com, https://www.datascience.com/blog/k-means-clustering (accessed Aug 31, 2018)

- Hate Speech Watch. 2014. Hate crimes: Consequences of hate speech. http: //www.nohatespeechmovement.org/hate-speech-watch/focus/ consequences-of-hate-speech, June. Seen on on 23rd Jan. 2016.

- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2014. Recursive deep models for semantic compositionality over a sentiment treebank. Steven Bird et. al. 2009. Natural language processing with python. Natural Language Processing with Python.