Volume 6, Issue 2, February 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Data Mining and Text Analytics of Twitter Data

[1]Karan Diware*, [2]Vikram Rajpurohit, [3]Nikit Kale, [4]Swati Ringe
Research Scholar (B.E.), [4]Assistant Professor
[1, 2, 3, 4,] Computer Engineering Department, Fr Conceicao Rodrigues College of Engineering, Mumbai University
Mumbai, Maharashtra, India

Abstract—Microblogging today has become a very popular communication tool among Internet users.Microblogging web-sites are rich sources of data for opinion mining and sentiment analysis. Because microblogging has appeared relatively recently, there are a few research works that were devoted to this topic.In this paper,we will explore how text analysis techniques can be used to dig into some of the data in a series of blog posts, focusing on different trends, tweets languages, tweets volumes for different trends on twitter. Furthermore, sentiment anaylysis or opinion mining can be performed more efficiently using machine learning algorithms compared to the traditional lexicon-based approach. Later, we perform data visualization to get a good visual representation of the data and highlight the interesting insights. Generally, this type of opinion mining is useful for consumers who are trying to research a product or service, or marketers researching public opinion of their company.

Keywords— Machine Learning, Sentiment Analysis, Data Visualization, Data preprocessing, NLP

I. INTRODUCTION

In the past decade, new forms of communication, such as micro-blogging and text messaging have emerged and become ubiquitous. Microblogging web-sites are rich sources of data for text analytics,opinion mining and sentiment analysis.While there is no limit to the range of information conveyed by tweets and texts, often these short messages (max 140 char) are used to share opinions and sentiments that people have about what is going on in the world around them.For example, During the 2014 Fifa World Cup, millions of fans and viewers from all over the globe used Social Media to share their thoughts and emotions about the games, teams and players and thus created massive amounts of data by doing so.Throughout the tournament, Facebook saw a record-breaking 3.4 billion interactions and Twitter saw a whopping 800 million Tweets about the World Cup. So, we decided to collect, analyze and visualize some of this data to look for interesting insights and correlations. We have worked on the following tasks:

1) Data scraping from Twitter:
This is the first step in a series of mining data on Twitter.Using the most popular language Python to crawl Twitter to gather data.

2) Data pre-processing:
Using Python we will study the structure of a tweet and we'll start digging into the processing steps we need for some text analysis .The tools that can be used are Weka , Rapidminer , RapidAnalytics , Python-NLTK , Knime , PSPP , Orange and R-libraries depending on the nature of the data collected.

3) Sentiment Analysis of the Tweets:
The purpose of this step is to build an algorithm that can accurately perform sentiment analysis and opinion mining on the given data. Our hypothesis is that we can obtain high accuracy on classifying sentiment in Twitter messages using machine-learning techniques .At first,we will discuss about the traditional Lexicon-based approach and later on move to machine learning algorithms.The tools that can be used for performing this task are Rapidminer,Weka, GATE(General Architecture for Text mining), R-Text Mining module and Python-NLTK.

4) Data Visualizations:
A picture is worth a thousand tweets: more often than not, designing a good visual representation of our data, can help us make sense of them and highlight interesting insights. After collecting and analyzing Twitter data, we will continue with some notions on data visualization with – Python, Tableu, R-programming. Other tools available are D3.js, QlikView , Weka , SAS - Visual Analytics for generic purposes and Flot , Gephi , WolframAlfa , Cytoscape , Instant Atlas for specific purposes.

II. LITERATURE REVIEW

During the past decade,the amount of data grew at an enormous rate even though the data stores are already vast and is still growing. The primary challenge is how to make the database a competitive business advantage by converting

seemingly meaningless data into useful information. Finding solution to this problem is critical because companies are increasingly relying on effective analysis of the information simply to remain competitive. A mixture of new techniques and technology is emerging to help sort through the data and find useful competitive data.

Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010). (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification, rather than the three-way polarity classification – negative,positive and neutral, as we do.

Efthymios Kouloumpis, TheresaWilson, Johns Hopkins University, United States of America ,Johanna Moore, School of Informatics University of Edinburgh, Edinburgh,United Kingdom, in a paper on 'Twitter Sentiment Analysis:The Good the Bad and the OMG!' in July 2011 have investigated the utility of linguistic features fordetecting the sentiment of Twitter messages. We evaluate the usefulness of existing lexical resources as well as features that capture information about the informal and creative language used in microblogging. We can take a supervised approach as well as an unsupervised approach to the problem, **but leverage existing hashtags in the Twitter data for building training data.

Also,S.Chandrakala in 'Opinion Mining and Sentiment focuses on Network Databases, Data Mining, Distributed Classification: A Survey' proposed a work on recent papers on sentiment analysis and its related tasks with future challenges. S.Padmaja in 'Opinion Mining and Sentiment Analysis – An Assesment of Peoples Belief: A Survey' proposed a work on commonly used Machine Learning Models for text classification and an overview of the most popular machine learning algorithm used in sentiment analysis.Raisa Varghese in her paper 'A Survey on Sentiment Analysis and Opinion Mining'proposed the different levels of sentiment analysis and the major challenges involved in sentiment analysis. Sindhu. C in her paper 'A Survey on Opinion Mining and Sentiment Polarity Classification' proposed a systematic flow and Machine learning approaches to optimize the performance.

A survey report from Pang and Lee on 'Opinion mining and sentiment analysis' gives a comprehensive study in the area with respect to sentiment analysis of blogs, reviews etc. Algorithms used in the survey include Maxium Entropy, SVM and Naive Bayes.

III. BRIEF DESCRIPTIONS

A. Data Scraping

The first step in this process is Data scraping from twitter.There are not any existing data sets of Twitter sentiment messages. Therefore,we collected our own set of data.

To start, Python is a great tool for grabbing data from the Web. Generally speaking, we'll get our data by either accessing an API (Application Programming Interface) or by 'scraping' the data off a webpage. The easiest scenario is when a site makes available an API. Twitter is such a site.Twitter's API provides a straightforward way to query for users and returns results in a JSON format which makes it easy to parse in a Python script.Collecting Twitter data is a great exercise in data science and can provide interesting insights in how people behave on the social media platform. Below is an overview of the steps to build a Twitter analysis from scratch.

First get the overview of Twitter API does and then follow the steps:

1. Get R or Python
2. Install Twitter packages
3. Get Developer API Key from Twitter
4. Write Code to Collect Tweets
5. Parse the Raw Tweet Data [JSON files]
6. Analyze the Tweet Data

Some of the tools that can be used in this process are : Beautiful Soup - Python library for scraping web pages , Twitter API-Python wrapper for performing API requests , MongoDB - open source document storage database, PyMongo - Python wrapper for interfacing with a MongoDB instance , Cronjob – time-based job scheduler

B. Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data goes through a series of steps during preprocessing:

Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

- Data Integration: Data with different representations are put together and conflicts within the data are resolved.
- Data Transformation: Data is normalized, aggregated and generalized.
- Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.
- Data Discretization: Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

The Anatomy of a Tweet:

- text: the text of the tweet
- created_at: the date of creation of the tweet
- favorite_count, retweet_count: the number of favourites and retweets
- favorited, retweeted: boolean stating whether the authenticated user have favourited or retweeted this tweet
- lang: acronym for the language (e.g. “en” for english)
- id: the tweet identifier
- place, coordinates, geo: geo-location information if available
- user: the author’s full profile
- entities: list of entities like URLs, @-mentions, hashtags and symbols
- in_reply_to_user_id: user identifier if the tweet is a reply to a specific user
- in_reply_to_status_id: status identifier id the tweet is a reply to a specific status

As we can see there’s a lot of information we can play with. All the *_id fields also have a *_id_str counterpart, where the same information is stored as a string rather than a big int (to avoid overflow problems). We can imagine how these data already allow for some interesting analysis: we can check who is most favourited/retweeted, who’s discussing with who, what are the most popular hashtags and so on. Most of the goodness we’re looking for, i.e. the content of a tweet, is anyway embedded in the text, and that’s where we’re starting our analysis.

1) Tokenisation

We start our analysis by breaking the text down into words. Tokenisation is one of the most basic, yet most important, steps in text analysis. The purpose of tokenisation is to split a stream of text into smaller units called tokens, usually words or phrases. While this is a well understood problem with several out-of-the-box solutions from popular libraries, Twitter data pose some challenges because of the nature of the language.

Let’s see an example, using the popular NLTK library to tokenise a fictitious tweet:

```
from nltk.tokenize import word_tokenize
tweet = ' RT @karan_diware: Hello World !!! :D http://mysite.com #Machine_Learning #NLP
print(word_tokenize(tweet))
# ['RT', '@', 'karan_diware', ':', 'hello', 'world', '!', ':', 'D', 'http', ':', '//mysite.com', '#', 'NLP'
```

There are some peculiarities that are not captured by a general-purpose English tokeniser like the one from NLTK: @-mentions, emoticons, URLs and #hash-tags are not recognised as single tokens.The tokeniser is probably far from perfect, but it gives you the general idea. The tokenisation is based on regular expressions (regexp), which is a common choice for this type of problem.

After tokenisation we can continue our preprocessing of the data :

2) Filtering

URL

People use twitter not only for expressing their opinions but also for sharing information with others. Given the short maximum length of tweets, one way of sharing is using links.Tweets include various links or URLs and these do not contribute to the sentiment of the tweet. The URLs in the data used in this project are of the form http://onion.com/page/334r5ty These do not contribute to the sentiment of the tweet. Hence these were parsed and replaced by a common word, URL .

Usernames:

Tweets often refer to other users and such references begin with the @ symbol. These again do not contribute to the sentiment and hence are replaced by the generic word USERNAME .

Duplicates or repeated characters.

People use a lot of casual language on twitter. For examples, 'happy' is used in the form of 'haaaaaappy'. Though this implies the same word 'happy', the classifiers consider these as two different words.

Table 1: Data Filtering

Tweets containing	Replaced by
http://onion.com/p/334r5ty	URL
@karan	USERNAME
whaaattttt	what
shiitttt	shit

To improve this and make words more similar to generic words, such sets of repeated letters are replaced by two occurrences. Thus,whaaatttt would be replaced by what.

3) Twitter Slang removal

Tweets contain a lot of casual language .Given that the maximum length of a tweet is 140 characters, people tend to use abbreviations or some short forms for words. These short words are replaced by the actual words that they represent to improve performance of the learning algorithms

Table 2: Twitter Slang removal

Twitter Slang	Actual word
bou	about
Bff	best friend forever
2moro	Tomorrow
f9	fine
nva	never
Lol	laugh out loud

The advantage of doing slang removal is evident from the above table. If these are not mapped to the common original word, then training on them would not produce good accuracy and may also cause overfitting, as these might not be found in the test data.

4) Stop Words Removal

In information retrieval, there exists many words that are added as conjunctions in sentences. For example, words like a,of,there,the, and, before, while, and so on do not contribute to the sentiment of the tweet. Also these words do not help in classifying the tweets as they appear in all classes of tweets. These words are removed from the data so as to avoid using them as features.The stop words corpus was obtained from NLTK. Some modifications were required to this as the corpus also had some negative words such as nor, not, neither which are important in identifying negative sentiments and should not be removed.

5) Stemming

In information retrieval, stemming is the process of reducing a word to its root form. For example, walking, walker, walked all these words are derived from the root word walk. Hence, the stemmed form of all the above words is walk.

NLTK provides various packages for stemming such as the Porter Stemmer, Lancaster Stemmer and so on. The Porter-Stemmer was used in this project which uses various rules for su x stripping. In addition to stemming the train and test data, the positive and negative word corpus was also stemmed. Stemming reduces the feature space as many de-rived words are reduced to the same root form. Multiple features now point to the same word and hence it increases the probability of the word.

Table 3: Stemming

Original words	Root word
Overwhelmed	Overwhelm
Overwhelming	Overwhelm
Overwhelm	Overwhelm

As we will see in the results section, stemming gives a good increase in accuracy. By stemming different derived words are mapped to their root words and this allows more matching between the tweets in the test and training set.

C. Sentiment Analysis

Sentiment Analysis is the computational study of people's opinions, attitudes and emotions toward a trend or an entity. The entity can represent individuals, events or topics. These topics are most likely to be covered by authors. According to some researchers - Opinion Mining extracts and analyzes people's opinion about an entity while Sentiment Analysis identifies the sentiment expressed in a text then analyzes it. Therefore, the target of Sentiment Analysis is to find opinions, identify the sentiments they express, and then classify their polarity.

Sentiment Analysis can be considered a classification process. There are three main classification levels in Sentiment Analysis: document-level, sentence-level, and aspect-level Sentiment Analysis.

Document-level Sentiment Analysis aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit (talking about one topic).

Sentence-level Sentiment Analysis aims to classify sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective. If the sentence is subjective, Sentence-level Sentiment Analysis will determine whether the sentence expresses positive or negative opinions.

Classifying text at the document level or at the sentence level does not provide the necessary detail needed opinions on all aspects of the entity which is needed in many applications, to obtain these details; we need to go to the aspect level.

Aspect-level Sentiment Analysis aims to classify the sentiment with respect to the specific aspects of entities.

This analysis can be useful for new comer researchers in this field as it covers the most famous Sentiment Analysis techniques and applications in one research paper. This paper uniquely gives a refined categorization to the various Sentiment Analysis techniques which is not found in other paper. In this paper, the authors give a closer look on these fields.

IV. FEATURE SELECTION

Feature selection is often integrated as the first step in machine learning algorithms like SVM, Neural Networks, k-Nearest Neighbors, etc. The main goal of the feature selection is to decrease the dimensionality of the feature space and thus computational cost. As a second objective, feature selection will reduce the overfitting of the learning scheme to the

training data. During this process, it is also important to find a good tradeoff between the richness of features and the computational constraints involved when solving the categorization task.

Some of the most frequently used statistical methods in Feature Selection and their related articles :

- Point-wise Mutual Information (PMI)
- Optimal orthogonal centroid
- Chi-square ($\chi 2$)
- Count Difference
- Latent Semantic Indexing (LSI)
- Document Frequency Difference (DFD)

Sentiment classification techniques

Sentiment Classification techniques can be roughly divided into machine learning approach, lexicon-based approach and the hybrid approach. The *Machine Learning Approach* (*ML*) applies the popular ML algorithms and uses linguistic features. The *Lexicon-based Approach* relies on a sentiment lexicon, a collection of known and precompiled sentiment terms.The *hybrid Approach* combines both approaches and is very common with sentiment lexicons playing a key role in the majority of methods.

Lexicon-based approach

The lexicon-based approach depends on finding the opinion lexicon which is used to analyze the text. There are two methods in this approach - dictionary-based approach and corpus-based approach. The dictionary-based approach which depends on finding opinion seed words, and then searches the dictionary of their synonyms and antonyms. The corpus-based approach begins with a seed list of opinion words, and then finds other opinion words in a large corpus to help in finding opinion words with context specific orientations. This could be done by using statistical or semantic methods.

Machine Learning approach

Machine Learning based approach depends on the popular machine learning algorithms for Sentiment Analysis or Opinion Mining. It uses the conventional Machine Learning algorithms to obtain optimum results.This approach can be classified further under supervised learning, unsupervised learning and weakly, semi and unsupervised learning.

A. Supervised Learning algorithms:

The traning data given by researchers which is manually classified tend to increase the popularity of the the supervised methods .The most common *approaches* here use the unigrams model as features when describing training and test data. In opinion mining certain sentiments are expressed in two or more words, and the accurate detection of negation is important because it reverses the polarity. Pedersen (2001) showed that word n-grams are effective features for word sense disambiguation, while Dave et al. (2003) indicated that they are able to capture negation. In an alternative approach to negation (Pang and Lee 2002), each word following a negation until the first punctuation receives a tag indicating negation to the learning algorithm.The various supervised learning models that can be used are explained below :

Naive Bayes classifier:

The Naive Bayes classifier is the simplest and most commonly used classifier. Naive Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the Bag of Words feature extraction which ignores the position of the word in the document. It is based on the Bayes Theorem to determine the probability that a given feature set(test data) belongs to a certain label.

Equation

$$P(label|features) = \frac{P(label) * P(features|label)}{P(features)}$$

P(*label*) is the prior probability of a label or the likelihood that a random feature set the label. *P*(*features* | *label*) is the prior probability that a given feature set(test data) is being classified as a label. *P*(*features*) is the prior probability that a given feature set is occurred. Given the Naive assumption which states that all features are independent, the equation could be rewritten as follows:

equation (4)

$$P(label|features) = \frac{P(label) * P(f1|label) * \ldots.. * P(fn|label)}{P(features)}$$

An improved Naive Bayes classifier was proposed by Kang and Yoo to solve the problem of the tendency for the positive classification accuracy to appear up to approximately 10% higher than the negative classification accuracy. This creates a problem of decreasing the average accuracy when the accuracies of the two classes are expressed as an average value. They showed that using this algorithm with restaurant reviews narrowed the gap between the positive accuracy and the negative accuracy compared to Naive Bayes and Support Vector Machine.

Maximum Entropy Classifier:

The Maximum entropy Classifier also known as a conditional exponential classifier converts labeled feature sets to vectors using encoding. This encoded vector is then used to calculate weights for each feature that can then be combined

to determine the most likely label for a feature set.The probability of each label is then computed using the following equation:

$$P(fs|label) = \frac{dotprod(weights, encode(fs, label))}{sum(dotprod(weights, encode(fs, l))forlinlabels)}$$

Kaufmann used Maximum Entropy classifier to detect parallel sentences between any language pairs with small amounts of training data. The other tools that were developed to automatically extract parallel data from non-parallel corpora use language specific techniques or require large amounts of training data. Their results showed that Maximum Entropy classifiers can produce useful results for almost any language pair. This can allow the creation of parallel corpora for many new languages.

Bayesian Network:

Bayesian Network is considered a complete model for the variables and their relationships.The main assumption of the NB classifier is the independence of the features. The other extreme assumption is to assume that all the features are fully dependent. In Text mining, the computation complexity of Bayesian Network is very expensive; that is why, it is not frequently used.

Bayesian Network was used by Hernández and Rodríguez to consider a real-world problem in which the attitude of the author is characterized by three different (but related) target variables. They proposed the use of multi-dimensional Bayesian network classifiers. It joined the different target variables in the same classification task in order to exploit the potential relationships between them. They extended the multi-dimensional classification framework to the semi-supervised domain in order to take advantage of the huge amount of unlabeled information available in this context. They showed that their semi-supervised multi-dimensional approach outperforms the most common Sentiment Analysis approaches, and that their classifier is the best solution in a semi-supervised framework because it matches the actual underlying domain structure.

Support Vector Machines:

Support Vector Machines are a group of supervised learning methods that can be applied to classification or regression. A Support Vector Machine is actually a discriminative classifier which is formally defined by a separating hyperplane, i.e., when it is given labeled training data , the algorithm outputs an optimal hyperplane which categorizes new examples.Support Vector Machines is based on the concept of decision planes that define decision boundaries

SVMs were used by Li and Li as a sentiment polarity classifier. Unlike the binary classification problem, they argued that opinion subjectivity and expresser credibility should also be taken into consideration. They proposed a framework that provides a compact numeric summarization of opinions on micro-blogs platforms. They identified and extracted the topics mentioned in the opinions associated with the queries of users, and then classified the opinions using SVM. They worked on twitter posts for their experiment. They found out that the consideration of user credibility and opinion subjectivity is essential for aggregating micro-blog opinions. They proved that their mechanism can effectively discover market intelligence (MI) for supporting decision-makers by establishing a monitoring system to track external opinions on different aspects of a business in real time.

Neural Network:

Neural Network consists of many neurons where the neuron is its basic unit. Multilayer neural networks are used for non-linear boundaries. These multiple layers are used to induce multiple piece-wise linear boundaries, which are used to approximate enclosed regions belonging to a particular class. The outputs of the neurons in the earlier layers feed into the neurons in the later layers. The training process is more complex because the errors need to be back-propagated over different layers.

Van de Camp and Van den Bosch presented that Neural Networks can be used also for the classification of personal relationships in biographical texts. They marked relations between two persons (one being the topic of a biography, the other being mentioned in this biography) as positive, neutral, or unknown. Their case study was based on historical biographical information describing people in a particular domain, region and time frame. They showed that their classifiers were able to label these relations above a majority class baseline score. They found that a training set containing relations, surrounding multiple persons, produces more desirable results than a set that focuses on one specific entity. They proved that Support Vector Machines and one layer Neural Network (1-NN) algorithm achieve the highest scores.

B. Weakly, semi and unsupervised learning algortihms:

The weak and semi-supervised algorithms are used in many applications. Youlan and Zhou have proposed a strategy that works by providing weak supervision at the level of features rather than instances. They obtained an initial classifier by incorporating prior information extracted from an existing sentiment lexicon into sentiment classifier model learning. They refer to prior information as labeled features and use them directly to constrain model's predictions on unlabeled instances using generalized expectation criteria. In their work, they were able to identify domain-specific polarity words clarifying the idea that the polarity of a word may be different from a domain to the other. They worked on movie reviews and multi-domain sentiment data set from IMDB and amazon.com. They showed that their approach attained better performance than other weakly supervised sentiment classification methods and it is applicable to any text classification task where some relevant prior knowledge is available.

There are also other unsupervised approaches that depend on semantic orientation using PMI or lexical association using PMI, semantic spaces, and distributional similarity to measure the similarity between words and polarity prototypes.

C. Natural Language Processing techniques:

The natural language processing tools can be used to facilitate the Sentiment Analysis process. It gives better natural language understanding and thus can help produce more accurate results of Sentiment Analysis. The approach for Sentiment Analysis presented by Caro and Grella was based on a deep NLP analysis of the sentences, using a dependency parsing as a pre-processing step. Their Sentiment Analysis algorithm relied on the concept of Sentiment Propagation, which assumed that each linguistic element like a noun, a verb, etc. can have an intrinsic value of sentiment that is propagated through the syntactic structure of the parsed sentence. They presented a set of syntactic-based rules that aimed to cover a significant part of the sentiment salience expressed by a text. They proposed a data visualization system in which they needed to filter out some data objects or to contextualize the data so that only the information relevant to a user query is shown to the user. In order to accomplish that, they presented a context-based method to visualize opinions by measuring the distance, in the textual appraisals, between the query and the polarity of the words contained in the texts themselves. They extended their algorithm by computing the context-based polarity scores. Their approach approved high efficiency after applying it on a manual corpus of 100 restaurants reviews.

D. Data Visualization

A picture is worth a thousand tweets: more often than not, designing a good visual representation of our data, can help us make sense of them and highlight interesting insights. After collecting and analyzing Twitter data, there are multiple tools for performing visualization in data science.

Python has already has two exclusive libraries for visualization, commonly known as *matplotlib and seaborn.*

*Matplotlib***:** Python based plotting library offers *matplotlib* with a complete 2D support along with limited 3D graphic support. It is useful in producing publication quality figures in interactive environment across platforms. It can also be used for animations as well.

*Seaborn***:** Seaborn is a library for creating informative and attractive statistical graphics in python. This library is based on matplotlib. Seaborn offers various features such as built in themes, color palettes, functions and tools to visualize univariate, bivariate, linear regression, matrices of data, statistical time series etc which lets us to build complex visualizations.

Now, while there are some options to create plots in Python using libraries like matplotlib or ggplot, one of the coolest libraries for data visualisation is probably D3.js which is, as the name suggests, based on Javascript. D3 plays well with web standards like CSS and SVG, and allows to create some wonderful interactive visualisations.

Text mining and certain plotting using R

There are certain packages that are not installed by default so one has to install them manually

The relevant packages are:

1. tm – the text mining package : The tm package offers a number of transformations that ease the tedium of cleaning data.
2. SnowballC – required for stemming :Typically a large corpus will contain many words that have a common root – for example: offer, offered and offering. Stemming is the process of reducing such related words to their common root, which in this case would be the word offer.
3. ggplot2 – plotting capabilities
4. wordcloud

Other tools available are D3.js, QlikView , Weka , SAS - Visual Analytics for generic purposes and Flot, Gephi , WolframAlfa , Cytoscape , Instant Atlas for specific purposes.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented an overview on the recent updates in Sentiment Analysis algorithms and applications. After analyzing a number of articles, it is clear that the enhancements of Sentiment Analysis algorithms are still an open field for research. Neural Networks, Naive Bayes and Support Vector Machines are the most frequently used Machine Learning algorithms for solving Sentiment Classification problem. They are actually considered a reference model where the other proposed algorithms are compared to.

The data from various social networking sites like twitter, information from micro-blogs and forums is used widely in Sentiment Analysis. These sites are a great source of information about the people's feelings or opinions about a certain matter or a product. To use data from these sites still needs deep analysis of the data using more advnaced algorithms.There is much research ongoing in these fields.

While performing sentiment analysis or opinion mining more efficiently, it is very important to consider the context of the data under supervision and also the user preferences. Therefore, we can't say that a particular algorithm is better than the other. Choice of the algorithm mainly depends on the context of the text or data. Also, Machine learning algorithms tend to perform better than the traditional algoithms as explained in the couple of papers we summarized. However, nowadays, many researchers have been attracted to using Natural Language Processing tools to reinforce the Sentiment Analysis process.This research still need many enhancements.

REFERENCES

[1] https://www.cs.uic.edu/~liub/teach/cs583-fall-05/CS583-**data**-prep.ppt
[2] https://www.scikit-learn.org/stable/modules/**preprocessing**.html
[3] *B. Liu* Sentiment analysis and opinion mining Synth Lect Human Lang Technol (2012)
[4] http://www.sciencedirect.com/science/article/pii/S0167923612001364
[5] B. Pang, L. Lee (http://www.nowpublishers.com/article/Details/INR-011) Opinion mining and sentiment analysis Found Trends Inform Retriev, 2 (2008), pp. 1–135
[6] http://www.sciencedirect.com/science/article/pii/S2090447914000550
[7] www.cs.cornell.edu/home/llee/papers/**sentiment**.pdf
[8] https://gate.ac.uk/sale/talks/gate-course-may10/track-3/module-11-ml-adv/module-11-sentiment.pdf
[9] https://www.lct-master.org/files/Mullen**Sentiment**CourseSlides.pdf
[10] http://www.sciencedirect.com/science/article/pii/S0167923612001339
[11] Jiao Jian, Zhou Yanquan. Sentiment Polarity Analysis based multi-dictionary. In: Presented at the 2011 International Conference on Physics Science and Technology (ICPST'11); 2011
[12] http://dl.acm.org/citation.cfm?doid=2436256.2436274
[13] Zhou L, Li B, Gao W, Wei Z, Wong K. Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In: Presented at the 2001 conference on Empirical Methods in Natural Language Processing (EMNLP'11); 2011.
[14] Livio Robaldo, Luigi Di Caro OpinionMining-ML , Comput Stand Interfaces (2012)
[15] http://www.sciencedirect.com/science/article/pii/S0167923612001388
[16] http://www.sciencedirect.com/science/article/pii/S0957417412009153
[17] http://www.sciencedirect.com/science/article/pii/S016792361200142X
[18] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.244.9953&rep=rep1&type=pdf
[19] http://ijiset.com/vol2/v2s6/IJISET_V2_I6_106.pdf
[20] http://www.ijcat.com/archives/volume4/issue6/ijcatr04061001.pdf
[21] http://www.ijarcce.com/upload/2014/july/IJARCCE2M%20s%20angulakshmi%20An%20Analysis%20on.pdf
[22] *Yung-Ming Li, Tsung-Ying Li Deriving market intelligence from microblogs Decis Support Syst (2013)*

