RESEARCH ARTICLE OPEN ACCESS

Commodity Juxtaposing from Tweets using Sentiment Analysis

Praveen Jayasankar[1] , Prashanth Jayaraman[2] ,Rachel Hannah[3]
1, 2,,3 Department Computer Science and Engineering
1,2 Meenakshi Sundararajan Engineering College, Chennai-24
3 St.Joseph's College of Engineering, Chennai-119

Abstract:

Sentiment Analysis is the process of finding the sentiments from different classes of words. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state, or the intended emotional communication. In this case, 'tweets'! Given a micro-blogging platform where official, verified tweets are available to us, we need to identify the sentiments of those tweets. A model must be constructed where the sentiments are scored, for each product individually and then they are compared with, diagrammatically, portraying users' feedback from the producers stand point.

There are many websites that offer a comparison between various products or services based on certain features of the article such as its predominant traits, price, and its welcome in the market and so on. However not many provide a juxtaposing of commodities with user review as the focal point. Those few that do work with Naïve Bayes Machine Learning Algorithms, that poses a disadvantage as it mandatorily assumes that the features, in our project, words, are independent of each other. This is a comparatively inefficient method of performing Sentiment Analysis on bulk text, for official purposes, since sentences will not give the meaning they are supposed to convey, if each word is considered a separate entity. Maximum Entropy Classifier overcomes this draw back by limiting the assumptions it makes of the input data feed, which is what we use in the proposed system.

1. INTRODUCTION

Machine learning is the science of getting computers to act without being explicitly programmed. It studies how to automatically learn to make accurate predictions based on past observations. Machine learning is a subfield of computer science stemming from research into artificial intelligence. It has strong ties to statistics and mathematical optimization, which deliver methods, theory and application domains to the field. Some of the examples of machine learning are face detection, spam filtering, fraud detection, etc.

Twitter is an ideal place for gathering large amounts of data, as compared to the RSS feeds which were used earlier. On an average, twitter is said to have 87% accuracy on its tweets, having official news accounts on twitter that are easily available.

Using a large set of official twitter feed, the sentiment of the tweets can be collected to form an overall “mood” of the tweets which can be used to gauge the stock market moves. This can be analysed using machine learning algorithms. As social media is maturing and growing, sentiment analysis of online communication has become a new way to gauge public opinions of events and actions in the world. Twitter has become a new social pulpit for people to quickly "tweet" or voice their ideas in a 140 characters or less. They can choose to "retweet" or share a tweet, to promote ideas that they find favourable and elect to follow others whose opinion that they value. The explosion of social media and the proliferation of mobile devices have created a “perfect storm” of opportunities for customers to express their feelings and attitudes about anything and everything at any time. The proposed system gathers tweets and performs Sentiment Analysis (SA) on it at first.

Several methods are available to gather tweets, the existing system uses Twitter4J to retrieve tweets. The algorithm used to analyse these tweets is Max

Entropy. The Max Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. Unlike the Naive Bayes classifier, the Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit our training data, selects the one which has the largest entropy. The Max Entropy classifier is be used to solve a large variety of text classification problems such as language detection, topic classification, sentiment analysis and more. Stanford CoreNLP library is used to employ this type of a classifier to analyse the tweets.

Further improvements can be made to this system by removing or filtering the irrelevant twitter data from the tweets by cleaning the tweets. The analysed tweet after being assigned a value or score is visualised using the JFreeChart library for java. The relevance between the two queries can be used to juxtapose them and find meaning in it.

2. LITERATURE SURVEY

AI and Opinion Mining by Hsinchun Chen and David Zimbra, published by the IEEE Computer Society, 2010.

The advent of Web 2.0 and social media content has stirred much excitement and created abundant opportunities for understanding the opinions of the public and consumers towards social events, political movements, company strategies, marketing campaigns, and product preferences. Opinion mining, a sub discipline within data mining, refers to the computational techniques for extracting, classifying, and assessing the opinions expressed in various online news sources, social media comments, and other user-generated content. Sentiment analysis is often used in opinion mining to identify the sentiment, subjectivity, and other emotional states in online text. The topics covered include how to extract opinion, sentiment, affect, and subjectivity expressed in text. For example, resources online might include opinions

about a product or the violent and racist statements expressed in political forums. Researchers have also been able to classify text segments based on sentiment, and subjectivity by analysing positive or negative sentiment expressed in sentences, the degree of violence expressed in forum messages, etc.,

The most well-studied sub problem is opinion orientation classification, at the document, sentence, and feature levels. The existing reported solutions are still far from perfect because current studies are still coarse and not much has been done on finer details. For accurate evaluation, the benchmark data needs to cover a large number of domains because a system that does well in one domain might not do well in another, as opinions in different domains can be expressed so differently. Precision and recall are commonly used as evaluation measures.

Sentiment Analysis on Twitter Data by Varsha Sahayak, Vijaya Shete, Apashabi Pathan, published in International Journal of Innovative Research in Advanced Engineering (IJIRAE), January 2015.

Now-a-days social networking sites are at the boom, so large amount of data is generated. Millions of people are sharing their views daily on micro blogging sites, since it contains short and simple expressions. In this paper, a paradigm to extract the sentiment from a famous micro blogging service, Twitter, is discussed, where users post their opinions for everything. Twitter is a demandable micro blogging service which has been built to discover what is happening at any moment of time and anywhere in the world. In this paper, the existing analysis of twitter dataset with data mining approach such as use of Sentiment analysis algorithm using machine learning algorithms are discussed. An approach is introduced that automatically classifies the sentiments of Tweets taken from Twitter dataset. These messages or tweets are classified as positive, negative or neutral with respect to a query term.

Machine learning algorithms were used for classifying the sentiment of Twitter messages using distant supervision. The training data consists of Twitter messages with emoticons, acronyms which are used as noisy labels. Sentiment analysis on Twitter data is examined. The contributions of this survey paper are: (1) Parts Of Speech (POS)-specific prior polarity features. (2) A tree kernel to prevent the need for monotonous feature engineering was used. It was found that social media related features can be used to predict sentiment in Twitter. Three machine learning algorithms which will contribute to outperform three models namely unigram, feature based model and tree kernel model by using Weka. The proposed system concludes the sentiments of tweets which are extracted from twitter.

Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis by Raymond Y.K. Lau, Chunping Li, Stephen S.Y. Liao Published in 2014 Elsevier B.V.

In the era of Web 2.0, there has been an explosive growth of consumer-contributed comments at social media and electronic commerce Web sites. Applying state-of-the-art social analytics methodology to analyze

the sentiments embedded in these consumer comments facilitates both firms' product design strategies and individual consumers' comparison shopping. However, existing social analytics methods often adopt coarse-grained and context-free sentiment analysis approaches. Consequently, these methods may not be effective enough to support firms and consumers' demands of fine-grained extraction of market intelligence from social media. Guided by the design science research methodology, the main contribution of our research is the design of a novel social analytics methodology that can leverage the sheer volume of consumer reviews archived at social media sites to perform a fine-grained extraction of market intelligence. More specifically, the proposed social analytics methodology is underpinned by a novel semi-supervised fuzzy product ontology mining algorithm.

3. IMPLEMENTATION

3.1SYSTEM ARCHITECTURE

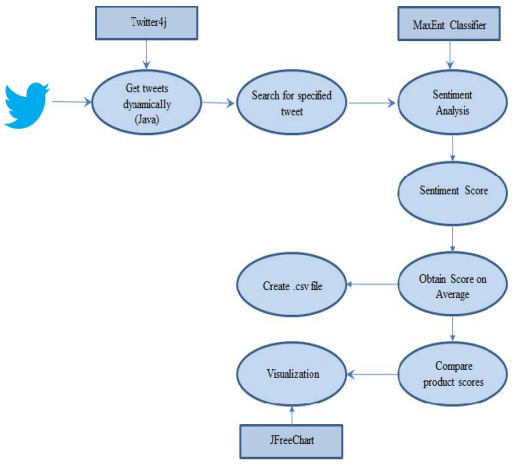


Figure 1. System Architecture

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. Here, we have picturized the basic model we had in mind during the development and implementation of our project. Initially we create a Twitter account that enables the creation of a Twitter Application. Having created a Twitter Application, we then focus on establishing a connection with the Twitter interface. This is done by including an unofficial Java library file called Twitter4j. We are now equipped to dynamically collect tweets from the Twitter website.

Once this set up is done, the system requests the user to enter the names of the products or services he/she wishes to make an analysis of based on user-reviews, and view the variation on a graph. A typical search is made of the tweets still online, and we accumulate those pertaining to the topics input by the user alone. Then, we enter the most crucial part of our project, in which we analyse each of the tweets retrieved, individually and assign a Sentiment Score to them. This is done by Maximum Entropy Classifier, which is very much suitable for works on SA, since words should not be considered as independent of each other, which is exactly what this classifier does. The scores are assigned on a scale of 1 to 3, and very rarely 4, such that the happier the sentiment of the tweet the higher its score. Once the Sentiment Score is obtained, we run a simple function that calculates the cumulative average of the score for each product. We then compare this data and visualize the variation in the liking of the product, using JFreeChart. In addition, we also store the average scores obtained for the products analysed, in a .csv file, as aid for future work.

3.2 DATA COLLECTION AND PROCESSING

Twitter4J is an unofficial Java library for the Twitter API. With Twitter4J, one can easily integrate their Java application with the Twitter service. Twitter4J is an unofficial library. This library provides functions that can be used to extract tweets we can search for tweets using the Query class and Twitter.search (twitter4j.Query) method. With OAuth authorization scheme, an application can access the user account without userid/password combination given. The application is registered at http://twitter.com/oauth_clients/new to acquire consumer key, and consumer secret in advance. Key / secret pair can be set via Twitter#setOAuthConsumer (). After the tweets are retrieved, they are fed into the pipeline to be analysed for sentiment.

The tweets are fed into the Standford Core NLP pipeline. Stanford CoreNLP is an integrated framework. Its goal is to make it very easy to apply a bunch of linguistic analysis tools to a piece of text. A CoreNLP tool pipeline can be run on a piece of plain text with just two lines of code. It is designed to be highly flexible and extensible. The backbone of the CoreNLP package is formed by two classes: Annotation and Annotator. Annotations are the data structure which hold the results of annotators. Annotations are basically maps, from keys to bits of the annotation, such as the parse, the part-of-speech tags, or named entity tags. Annotators are a lot like functions, except that they operate over Annotations instead of Objects. They do things like tokenize, parse, or NER tag sentences. Annotators and Annotations are integrated by AnnotationPipelines, which create sequences of generic Annotators. Stanford CoreNLP inherits from the AnnotationPipeline class, and is customized with NLP Annotators.

The annotators are called to initialise the pipeline. Annotations are the data structure which hold the results of annotators. Annotations are basically maps, from keys to bits of the annotation, such as the parse, the part-of-speech tags, or named entity tags. Annotators are a lot like functions, except that they operate over Annotations instead of Objects. The syntactically arranged tree from the parser is worked with for performing SA on it.

3.3 PROBABLITY DISTRIBUTION

In probability and statistics, a probability distribution assigns a probability to each measurable subset of the possible outcomes of a random experiment, survey, or procedure of statistical inference. Examples are found in experiments whose sample space is non-numerical, where the distribution would be a categorical distribution; experiments whose sample space is encoded by discrete random variables, where the distribution can be specified by a probability mass function; and experiments with sample spaces encoded by continuous random variables, where the distribution can be specified by a probability density function. More complex experiments, such as those involving stochastic processes defined in continuous time, may demand the use of more general probability measures. To define probability distributions for the simplest cases, one

needs to distinguish between discrete and continuous random variables. In the discrete case, one can easily assign a probability to each possible value: for example, when throwing a fair die, each of the six values 1 to 6 has the probability 1/6. In contrast, when a random variable takes values from a continuum then, typically, probabilities can be nonzero only if they refer to intervals: in quality control one might demand that the probability of a "500 g" package containing between 490 g and 510 g should be no less than 98%.

The target is to use the contextual information of the document (unigrams, bigrams, other characteristics within the text) in order to categorize it to a given class (positive / neutral / negative, objective / subjective etc). Following the standard bag-of-words framework that is commonly used in natural language processing and information retrieval, let $\{w_1,\ldots,w_m\}$ be the m words that can appear in a document. Then each document is represented by a sparse array with 1s and 0s that indicate whether a particular word wi exists or not in the context of the document.

The target is to construct a stochastic model, as described by Adam Berger (1996), which accurately represents the behaviour of the random process: take as input the contextual information x of a document and produce the output value y. As in the case of Naive Bayes, the first step of constructing this model is to collect a large number of training data which consists of samples represented on the following format: (x_i,y_i) where the x_i includes the contextual information of the document (the sparse array) and y_i its class. The second step is to summarize the training sample in terms of its empirical probability distribution:

$\boldsymbol{p'(x,y)} \equiv \frac{1}{N} \times$ number of times that (x,y) occurs in the sample where, N is the size of the training dataset

The above empirical probability distribution is used in order to construct the statistical model of the random process which assigns texts to a particular class by taking into account their contextual information. The building blocks of our model will be the set of statistics that come from our training dataset i.e. the empirical probability distribution.

For a particular feature (text in the case of text classification), the following indicator function is used:

$$f_j(x,y) = \begin{cases} 1, & if\ y = c\ and\ x\ contains\ w_k \\ 0, & otherwise \end{cases}$$

The above indicator function is called a "feature". This binary valued indicator function returns 1 only when the class of a particular document is ci and the document contains the word w_k.

When a particular statistic is useful to our classification, we require our model to accord with it. To do so, we constrain the expected value that the model assigns to the expected value of the feature function fj. The expected value of feature fj with respect to the model $p(y|x)$ is equal to:

$$\boldsymbol{p(f_j)} \equiv \sum p'(\boldsymbol{x})\boldsymbol{p(y|x)}\boldsymbol{f_j(x,y)}$$

Where, p′(x) is the empirical distribution of x in the training dataset and it is usually set equal to 1/N.

By constraining the expected value to be the equal to the empirical value and from the above equations, we have that

$$\sum_{x,y} p'(x)\, p(y|x)\, f_j(x,y) = \sum_{x,y} p'(x,y)\, f_j(x,y)$$

The above constrains can be satisfied by an infinite number of models. So in order to build the model, select the best candidate needs to be selected based on a specific criterion. According to the principle of Maximum Entropy, the model that is as close as possible to uniform must be selected. In order words, the model p* with Maximum Entropy must be selected:

$$p^* = \arg\max(-\sum_{x,y} p'(x)p(y|x)\log p(y|x))$$

Given that,

1. $p(y|x) \geq 0$,for all x,y
2. $\sum_y(y|x)=1$,for all x
3. $\sum_{x,y} p'(x)\, p(y|x)\, f_j(x,y) = \sum_{x,y} p'(x,y)\, f_j(x,y)$ for j€ {1,2…n}

3.4 CLASSIFICATION USING MAXIMUM ENTROPY

The Max Entropy classifier is a discriminative classifier commonly used in Natural Language Processing, Speech and Information Retrieval problems. Implementing Max Entropy in a standard programming language such as JAVA, C++ or PHP is non-trivial primarily due to the numerical optimization problem that one should solve in order to estimate the weights of the model. It must be noted that Max Entropy classifier performs very well for several Text Classification problems such as Sentiment Analysis and it is one of the classifiers that is commonly used to power up the Machine Learning API. The Max Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. Unlike the Naive Bayes classifier, the Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit the training data, selects the one which has the largest entropy. The Max Entropy classifier can be used to solve a large variety of text classification problems such as language detection, topic classification, sentiment analysis and more. Due to the minimum assumptions that the Maximum Entropy classifier makes, it is regularly used when nothing about the prior distributions are known and when it is unsafe to make any such assumptions. Moreover Maximum Entropy classifier is used when the conditional independence of the features cannot be assumed. This is particularly true in Text Classification problems where the features are usually words which obviously are not independent. The Max Entropy requires more time to train comparing to Naive Bayes, primarily due to the optimization problem that needs to be solved in order to estimate the parameters of the model. Nevertheless, after computing these parameters, the method provides robust results and it is competitive in terms of CPU and memory consumption.

3.5 WORKING WITH THE CLASSIFIER

The Stanford Classifier is a general purpose classifier - something that takes a set of input data and assigns each of them to one of a set of categories. It does this by generating features from each datum which are associated with positive or negative numeric "votes" (weights) for each class. In principle, the weights could be set by hand, but the expected use is for the weights to be learned automatically based on hand-classified training data items. (This is referred to as "supervised learning".) The classifier can work with (scaled) real-valued and categorical inputs, and supports several machine learning algorithms. It also supports several forms of regularization, which is generally needed when building models with very large numbers of predictive features. The underlying principle of maximum entropy is that without external knowledge, one should prefer distributions that are uniform. Constraints on the distribution, derived from the labelled training data, inform the technique where to be minimally non-uniform. The maximum entropy formulation has a unique solution which can be found by the improved iterative scaling algorithm.

A Recursive Neural Network (RNN) is a kind of deep neural network created by applying the same set of weights recursively over a structure, to produce a structured prediction over variable-length input, or a scalar prediction on it, by traversing a given structure in topological order. RNNs have been successful in learning sequence and tree structures in natural language processing, mainly phrase and sentence continuous representations based on word embedding. RNNs have first been introduced to learn distributed representations of structure, such as logical terms. Using Recursive Neural Networks, SA is performed. After performing sentiment analysis on the tweets, the sentiment scores are displayed with the tweet, on the console.

3.6 SENTIMENT ANALYSIS AND VISUALISING THE DATA

Bag of words classifiers can work well in longer documents by relying on a few words with strong sentiment like ‘awesome’ or ‘exhilarating.’ However, sentiment accuracies even for binary positive/negative classification for single sentences has not exceeded 80% for several years. For the more difficult multiclass case including a neutral class, accuracy is often below 60% for short messages on Twitter. From a linguistic or cognitive standpoint, ignoring word order in the treatment of a semantic task is not plausible, and, as we will show, it cannot accurately classify hard examples of negation. Correctly predicting these hard cases is necessary to further improve performance. Analysis for the new Sentiment Treebank which includes labels for every syntactically plausible phrase in thousands of sentences, allowing us to train and evaluate compositional models is introduced here. The Stanford Parser (Klein and Manning, 2003) is used to parses all 10,662 sentences. In approximately 1,100 cases it splits the snippet into multiple sentences. Starting at length 20, the majority are full sentences. One of the findings from labelling sentences based on reader’s perception is that many of them could be considered neutral. It is noticed that stronger

sentiment often builds up in longer phrases and the majority of the shorter phrases are neutral.

Another observation is that most annotators moved the slider to one of the five positions: negative, somewhat negative, neutral, positive or somewhat positive. The extreme values were rarely used and the slider was not often left in between the ticks. Hence, even a 5-class classification into these categories captures the main variability of the labels. The five classes work in that, higher the sentiment, greater the score.

In other words, 0 for highly negative, 1 for not so negative, 2 for neutral and 3 for somewhat positive and 4 for highly positive. This is called fine-grained classification. Sentiment analysis with a compositional model over trees using deep learning is done. Nodes of a binarized tree of each sentence, including, in particular, the root node of each sentence, are given a sentiment score. The data thus computed is juxtaposed and visualised using JFreeChart.

3.7 PERFORMANCE OF MAXIMUM ENTROPY

In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

The F1 score (also F-score or F-measure) is a measure of a test's accuracy in statistical analysis of binary classification. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

3.8 CREATION OF THE CSV FILE

CSV Files. A CSV is a comma separated values file, which allows data to be saved in a table structured format. CSVs look like a garden-variety spreadsheet but with a .csv extension. In computing, a comma-separated values (CSV) file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. In popular usage, however, the term "CSV" may denote some closely related delimiter-separated formats, which use a variety of different field-delimiters. These include tab-separated values and space-separated values, both of which are popular. Such files are often even given a .csv extension, despite the use of a different field separator (not a comma). CSV is a delimited text file that uses a comma to separate. Simple CSV implementations may prohibit field values that contain a comma or other special characters such as newlines. More sophisticated CSV implementations permit

them, often by requiring " (double quote) characters around values that contain reserved characters (such as commas, double quotes, or less commonly, newlines). Embedded double quote characters may then be represented by a pair of consecutive double quotes,or by prefixing an escape character such as a backslash (for example in Sybase Central).

CSV formats are not limited to a particular character set.[1] They work just as well with Unicode character sets as with ASCII. CSV files normally will even survive naive translation from one character set to another. CSV does not, however, provide any way to indicate what character set is in use, so that must be communicated separately, or determined at the receiving end (if possible). Databases

that include multiple relations cannot be exported as a single CSV file.

Similarly, CSV cannot naturally represent hierarchical or object-oriented database or other data. This is because every CSV record is expected to have the same structure. CSV is therefore rarely appropriate for documents such as those created with HTML, XML, or other mark-up or word-processing technologies.

Statistical databases in various fields often have a generally relation-like structure, but with some repeatable groups of fields. For example, health databases such as the Demographic and Health Survey typically repeat some questions for each child of a given parent. Statistical analysis systems often include utilities that can "rotate" such data; for example, a "parent" record that includes information about five children can be split into five separate records, each containing (a) the information on one child, and (b) a copy of all the non-child-specific information. CSV can represent either the "vertical" or "horizontal" form of such data.

In a relational database, similar issues are readily handled by creating a separate relation for each such group, and connecting "child" records to the related "parent" records using a foreign key (such as an ID number or name for the parent). In mark-up languages such as XML, such groups are

typically enclosed within a parent element and repeated as necessary (for example, multiple <child> nodes within a single <parent> node). With CSV there is no widely accepted single-file solution.

4. SCREENSHOTS

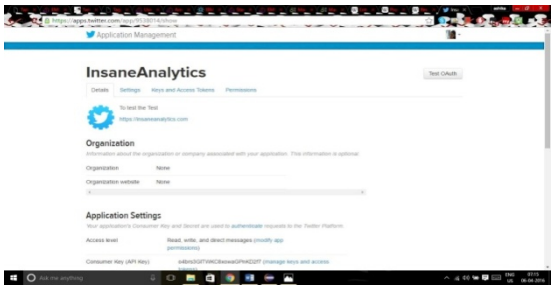


Figure 3 Creation of Twitter Application

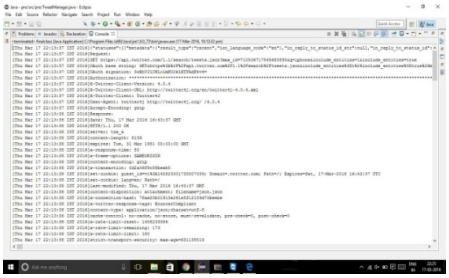


Figure 4 Connection Establishment

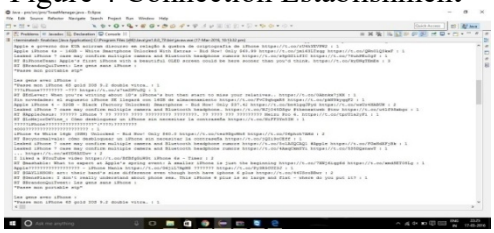


Figure 5 Processing of Tweets

Figure 6 Computing the Average

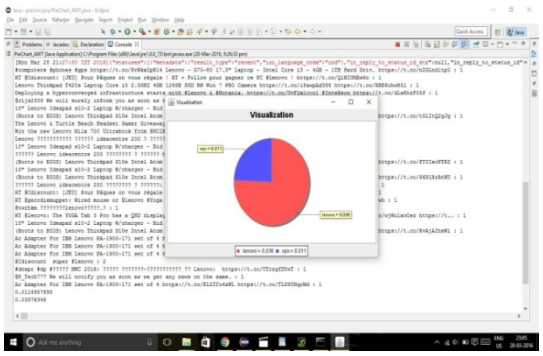


Figure 7 Visualization of the Data

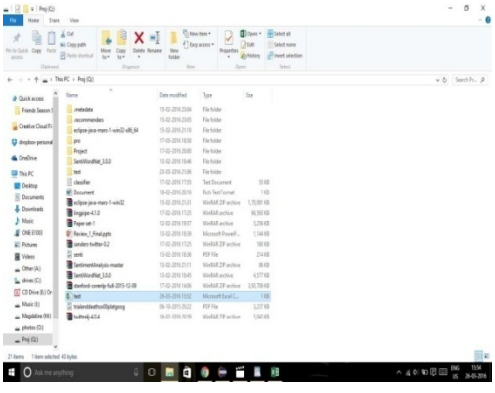


Figure 8 Creation of .csv file

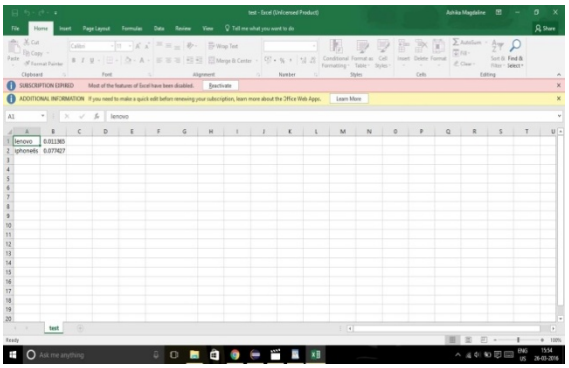


Figure 9. Average Scores stored in the CSV file

5. PERFORMANCE METRICS

Performance metrics measure an organization's activities and performance. It should support a range of stakeholder needs from customers, shareholders to employees. While traditionally many metrics are finance based, inwardly focusing on the performance of the organization, metrics may also focus on the performance against customer requirements and value. Performance of the models are calculated by taking the average of the means of the predicted training scores, test scores and précision recall scores of the training and the test data. A criticism of performance metrics is that when the value of information is computed using mathematical methods, it shows that even performance metrics professionals choose measures that have little value. This is referred to as the "measurement inversion". For example, metrics seem to emphasize what organizations find immediately measurable even if those are low value and tend to ignore high value measurements simply because they seem harder to measure (whether they are or not).To correct for the

measurement inversion other methods, like applied information economics, introduce the "value of information analysis" step in the process so that metrics focus on high-value measures. Organizations where this has been applied find that they define completely different metrics than they otherwise would have and, often, fewer metrics. There are a variety of ways in which organizations may react to results. This may be to trigger specific activity relating to performance or to use the data merely for statistical information. Often closely tied in with outputs, performance metrics should usually encourage improvement, effectiveness and appropriate levels of control. Performance metrics are often linked in with corporate strategy and are often derived in order to measure performance against a critical success factor.

The maxent classifier has been trained on 1.6 million tweets, 800k with positive sentiment and 800k with negative sentiment. The 1.6 million tweets used to train the maxent classifier were not manually classified, but rather their classifications were inferred from emoticons (an example of distant supervision)

A. PRECISION AND RECALL

To more accurately compare classifier performance, we look at their rates of precision and recall: Precision is the percentage of cases predicted to be of class X which are in fact of class X. It answers the question: "to what extent can we trust a prediction?"

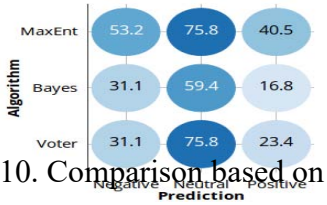


Figure 10. Comparison based on Precision

To the left you can see the precision rates of the three algorithms with respect to each classification bin.

MaxEnt is the best-performing algorithm in terms of precision. It weakly dominates Voter and strongly dominates Bayes. Precision is consistently achieved for the positive class. The precision rates are calculated using the Precision Rate function. We can tune an algorithm for high precision simply by predicting only cases which are (relatively) clear-cut. Hence we must also look at the rate of recall: Recall is the percentage of cases which actually have class X which are correctly predicted to have class X by the algorithm. This answers the question: "given a case of class X how likely is it to be correctly identified as such?"

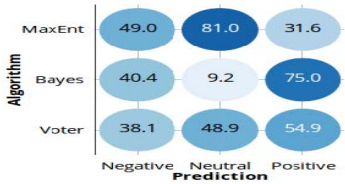


Figure 11 Comparison based on Recall

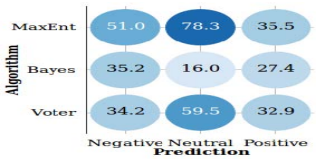
To the left you can see the recall rates of the three algorithms by class {"negative", "neutral", "positive}. The best performance in terms of recall rates is shown by the maxent algorithm, although its recall rate for "positive" is also low (31.6%).

The Bayes algorithm shows an extremely low recall rate for "neutral" —it tends to classify the tweets as either "positive" or

"negative". The rates of recall are calculated using the RecallRate function.

B. F-MEASURES

There is a trade-off between precision and recall: The more cases you try to catch into your classification, the more false classifications you are going to make as you move from the easy to the hard cases. Let us therefore look at the F-measures of the algorithms: the harmonic mean of precision and recall: F = 2 * (precision * recall) / (precision + recall) To the left you can see the F-measures by algorithm and classification bin.



The maxent algorithm performs best on this measure and strictly dominates the other two algorithms. The voter algorithm comes close for the "positive" bin.

6. CONCLUSION AND FUTURE WORK

Sentiment analysis is a special field of text analysis. In short, focus and analyse the extracted opinions from the posted comments. This project's goal is to analyse the sentiments on a topic which are extracted from the Twitter and conclude a remark (positive/negative) of the defined topics. We have implemented an easier procedure to analyse sentiments on any interested field or topic. In addition, we have also implemented a process to juxtapose or compare two or more products or services, thus giving the producer or the customer a more intuitive method of being decisive of his/her purchase. Hope this project would helpful for anyone in any way to meet up their interests or what they deserve. This is our major goal of this project and waiting to provide much more worthy works in our future work.

Currently, although MaxEnt Classifier is a probabilistic classifier and works better than Naïve Bayes classifier for a Corpus-based approach, it is evident that the algorithm needs further improvement in terms of an optimal feature selection. We also plan to make utmost use of the CSV file we have created by storing the averages generated for each topic and taking that into consideration when the same topic is wanted to be juxtaposed the next time, even by another user.

REFERENCES

[1] AI and Opinion Mining by Hsinchun Chen and David Zimbra Published by the IEEE Computer Society

[2] Sentiment Analysis on Twitter Data by Varsha Sahayak Vijaya Shete Apashabi Pathan, Published in International Journal of Innovative Research in Advanced Engineering (IJIRAE)

[3] Sentiment Analysis Algorithms and Applications – A Survey by Walaa Medhat, Ahmed Hassan, Hoda Korashy, Published in Ain Shams Engineering Journal

[4] Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis by

Raymond Y.K. Lau, Chunping Li, Stephen S.Y. Liao, Published in 2014 Elsevier B.V.

[5] Sentiment Analysis using Fuzzy Logic by Md. Ansarul Haque and Tamjid Rahman, Published in International Journal of Computer Science, Engineering and Information Technology