International Journal of Technology and Engineering System (IJTES)
Vol 8. No.1 – Jan-March 2016 Pp.101-114
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

REALTIME ANALYSIS OF SOCIAL NETWORK TWITTER FOR RATING AND BLOCKING USER COMMENTS USING NLP

Jayamoorthy
Assistant Professor
MVIT
Puducherry

S.Divya
Student
MVIT
Puducherry

M.Shanmathy
Student
MVIT
Puducherry

R.Nalini Karpagam
Student
MVIT
Puducherry

ABSTRACT

In day to day life, everyone need to share their taught, opinion etc., between their friends, relatives and society in fast and rapid velocity of data exchange range rate for every seconds. But we can't measure the originality of the message and message is having any false, positive or negative, abuse or vulgar words what they share. In Existing social network, Twitter Account the Natural Language Processing is used to interact with the Human words and machine language to identify and to rate the comments shows whether the posted comment is positive or negative/neutral. In this paper, the work is based on sentimental analysis for rating and blocking the castigating words used by the twitter followers from their Bag Of Words. Finally, Dashboard visualizing tools is used to represent the restricted words from the posted comments of twitter followers.

Keywords: Social network, Natural language processing, machine language, castigating words, bag of words and dashboard.

1.INTRODUCTION

There are number of social networking services and loads of usage at present condition. Social networking provides both advantages and disadvantages. People utilize social network only for communication purpose. Twitter is one of the commonly used social media between short times it gained a worldwide popularity. More than 350 millions of posts are tweeted per day. The major drawback of existing and the tweets can be visited by anyone via commented. So twitter does not provide privacy and security. The networking issues have become a matter of serious concern. The user feels insecure while using twitter as it shares all the personal information in common with third parties. In our proposed system we use Natural language processing (NLP) to identify malicious feedback ratings. NLP acts as a detecting technique which detects the negative or malicious comments and also blocks the abused or negative comments stopping it from sharing the personal details to the public blog.

Twitter is used as a great tool for twitting messages and sharing some universal related topic or discussions it is also used for expressing feelings. Twitter when focused in business purpose it is very useful.But when it deals with confidentiality it lacks stability.

Twitter while considered as a great business platform consumes more help in official usage. By using twitter a large group of communication is available and target markets are insured which is useful for the business clients to develop or exaggerate their business. In other side twitter is the first site that allows spammers. Each spammer should be identified, weed out and filtered from the list time to time. The spammer in the twitter may enter the site and misuse the shared data of other users. They may also pass negative comments and spread malicious feedback [1].

1.1Related Work

A balanced stage should be maintained with getting friendly with your followers. When this exceeds may cause major problem and ends in negative commenting and abused words. This happens often in twitter. For instance, consider A and B who shares posts or tweet some information regarding politics or post tweets commenting movies. In that case a quarrel or fight may occur between the two. This causes disturbance to other user who view the comments. There are 90% of chances for other user to publish the fight and comment about it in comment. This promotes a hypothetical state and the state is unavoidable.

In existing system bloom filter where used to find malicious feedback rates and prevent it from abusing comments and attacks. The abusing comments are measured and rated in success ratio recommending the service. This checks the comments of the web service which in turn is employed metric and investigates whether to be recommended to the user or not. Then, the metric sends an authentication that the comments delivery for you is abused. Then the user may tend to erase the comments [2]. This method may not help in all times. If the authentication delivered the results in the delay or corrupted this may affect the user details.

ACS (abusing comment system depends on previous information establishing bond among unknown user. ACS of web services is a issue that no history of new comers is present. In twitter there may be fake tweets and fake addresses which allocate malicious feedback to other users. Malicious feedback in twitters is a preferable assessment of an attribute described on single entity relating observations causing problems. More than one source are simulated deriving abused comments. A reference to aggregated perception is ensured allowing the service requesters for providing abusing comments [3]. Web Service recommendation systems can be employed to recommend the optimal Web service for satisfying user's requirements.

Service recommendations are helpful for uers when two or more Web services have the same functionality but different Natural language processing (NLP) performance. NLP is defined as a set of non-functional properties, including Abusing comments, response time, reliability, etc. When multiple

Web services formulated AC provides the same functionality, then a feedback rating requirement can be used as a secondary criterion for service selection. Language processing is a set of non-functional attributes like service response time, throughput, reliability, and availability. Service computing are used with multiple and separate systems adopting several business domains as a package functionality suiting routines [4]. While using twitter the privacy is in jeopardy only because of the social media which affects the pride of a person.

1.2 Literature Audit

A major concern when incorporating large sets of diverse n-gram features for sentiment classification is the presence of noisy, irrelevant, and redundant attributes. These concerns can often make it difficult to harness the augmented discriminatory potential of extended feature sets. (Ahmed Abbasi, Stephen France, Zhu Zhang and Hsinchun Chen, 2011)proposed a rule-based multivariate text feature selection method called Feature Relation Network (FRN) that considers semantic information and also leverages the syntactic relationships between n-gram features.Furthermore, by incorporating syntactic information about n-gram relations, FRN is able to select features in a more computationally efficient manner than many multivariate and hybrid techniques.

Twitter sentiment analysis has become widely popular. However, stable Twitter sentiment classification performance remains elusive due to several issues: heavy class imbalance in a multi-class problem, representational richness issues for sentiment cues, and the use of diverse colloquial linguistic patterns. These issues are problematic since many forms of social media analytics rely on accurate underlying Twitter sentiments. Accordingly, a text analytics framework is proposed by (Ammar Hassan, and Ahmed Abbasi, Daniel Zeng 2013) for Twitter sentiment analysis. The framework uses an elaborate bootstrapping ensemble to quell class imbalance and representational richness issues.. Consequently, it is able to build sentiment time series that are better able to reflect events eliciting strong positive and negative sentiments from users.

Fast algorithms capable of processing massive volumes of data are now required in the field of power systems. It presents a novel parallel detrended fluctuation analysis (PDFA) approach for fast event detection on massive volumes of PMU data.The algorithm is evaluated using data from installed PMUs on the transmission system of Great Britain from the aspects of speedup, scalability, and accuracy. A revision to the law is then proposedby(Jiang Zheng, Aldo Dagnino, 2014) suggesting enhancements to its capability to analyse the performance gain in computation when parallelizing data intensive applications in a cluster computing environment.

The vast majority of existing approaches to opinion feature extraction rely on mining patterns only from a single review corpus, ignoring the non-trivial disparities in word distributional characteristics of opinion features across different corpora. Zhen Hai, Kuivu Chang, Jung-Jae Kim, 2014) proposed a novel method to identify opinion features from online reviews by exploiting the difference in opinion feature statistics across two corpora, one domain-specific corpus (i.e., the given review corpus) and one domain-independent corpus (i.e., the contrasting corpus). Experimental results on two real-world review domains show the proposed IEDR approach to outperform several other well-established methods in identifying opinion features.

The current Analytics tools and models that are available in the market are very costly, unable to handle Big Data and less secure. The traditional Analytics systems takes a long time to come up with results, so it is not beneficial to use for Real Time Analytics. So, (Gaurav D Rajurkar, Rajeshwari M Goudar, 2015) proposed the work resolve all these problems by combining the Apache Open Source platform which solves the issues of Real Time Analytics using HADOOP.The HANDOOP, is flexible and scalable architecture. The proposed work i- based upon the phenomenon of combination of open source software along with commodity hardware that will increase the profit of IT Industry.

2.MODULE DESCRIPTION

2.1Fetching raw twitter data

First of all ,we need to fetch the twitter data from the original twitter api. Real time Streaming data is obtained from twitter Public API. Connection to twitter is established using Twitter4j API present in spark streaming. Twitter Api uses Consumer Key, Consumer Secret, Access Key, Access Token to connect and authenticate twitter, Which is provided to code at run time.

2.2 Cleansing raw twitter data

Hash tags and @mentions are declared in the code as filter in order to obtained target source data from twitter. Only necessary fields like SCREEN NAME ,TEXT, CREATED_DATE,LANGUAGE,RETWEET _COUNT are extracted from the raw twitter data obtained from the public api . All junk character and other unnecessary symbols apart from English words are removed so that model training can be done efficiently.

2.3 Pre - processing the data

Pre-processing is the main step in data mining process. Normally Data pre-processing is transforming of raw data into an understandable format. In this project, Customized Schema with some additional fields which are necessary for later components are built .

2.4 Analytics

In this module ,three process are need to be done.

(i) Creation, (ii) Training ,(iii) Prediction.

2.5 Model Creation

All the Analytical models are built Using MLIB library present in programming language .Since Sentiment Analysis is classification problem , classification model is built with Naïve Bayes Model as the choice of algorithm. Addition libraries like Lucien English, Hashing TF, Stemmer are used to further remove words like a, the, or and some repeated words which has no weightage is removed.

2.6 Model training

Using around 1 million historical records of cleansed twitter data, which is similar to the real time data obtained from twitter are used to train the model created in the above step. Once the model is trained and after efficiency is evaluated with many different classification algorithms best resulting model like Naïve Bayes Model is saved to the disk. In this phase we store only meta information which can be used in later stages.

2.7Prediction module

In this module the real time data is cleansed and sent to stemmer for removing words which has no weightage . Prediction Function is written and called which in turn will used the already trained and saved model for prediction in real time. All the output of prediction function with some additional information is saved in to MYSQL tables for Data visualization and reporting.

2.8Data visualization

This module is to visualize the date that are gathered in th above process.The gathered data are then visualized in form of graph or charts.The two kind of development used in this module are

- ❖ Front End Development
- ❖ Back End Development

Front end development

Front End View is further divided into 2 parts

- ➢ Archival Stats
- ➢ Real Time Tweets with sentiment

The languages used in this project for front end are HTML,CSS and JSP. Dashboard skeletal layouts are created. Other Navigation links to and between home and real time tweets with sentiment page is created. Charts and Dashboards are set to auto refresh to show the real time data changes with minimal delay. Dashboards are mainly used to visualize the data in a linear manner.

Back end development

Data is sourced from 2 different MYSQL tables

- ➢ Real Time Table
- ➢ Archive Table

These tables are used to populate real time graphs, Real time tweets and archival charts.

Servlet with JDBC connection code is written to source data from MYSQL tables. Data is grouped and drill down is created with the help of DC inbuilt functionalities Charts and other metrics are created using D3 and Dc JavaScript libraries .Necessary function which are to be called through frontend is created to show dashboard seamlessly .

2.9Data integration

The main purpose of this module is to integrate the data from the database. servlet is used as the Integration layer to pull the data from the MYSQL tables to dashboards periodically with minimal delay.

Real-time tables are truncated for every new data pulled from twitter .Stored Procedure is created to move the data from real time table to archival table .Stored Procedure pulls the data before it gets truncated .All the backend data movement is automated/managed by writing schedule script and calling the stored procedure periodically in MYSQL.

All the Front end components are periodically refreshed based out on the predefined calculated interval set to show/populate the dashboard with very minimal delay. Integration is otherwise termed as scheduling.

3.Implementation and results

Fig 1: Twitter login page

Fig 2: Our Created Twitter App Login Page

Fig 3:Admin Sign In Page

Fig 4:Our App Sign In Page

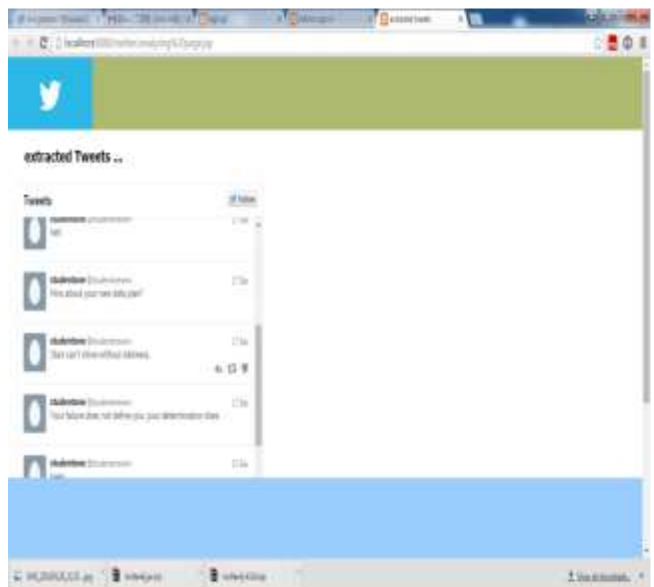

Fig 5:After Sign In Process ,Extraction Of Tweets shoul be processed

Fig 6:Twitter Welcome Page

4.EVALUATING MALICIOUS FEEDBACK/ABUSE COMMENTS DETECTION

Malicious feedbacks are rated accordingly with ACS (abusing computing system). ACS acts as a detecting sector in which the comments are rated and measured. The comments tweeted are separated as division.

The abused and malicious feedback can be classified by measuring:

- Positive feedbacks
- Negative feedbacks

In twitter the feedback ratings are calculated by identifying the indications. The indications are simultaneously recruited and verified. Abusing comments are boots trappers that access newly deployed services. The rating defines a theoretical analysis in which measurements are profiled. The reputation feedback is also measured alternative ways [5].

If it shows positive indication then the comments are positive and the feedback is in stable condition. There is no need for any malicious prevention in this stage of indication.

If it indicates negative indication then the comments are negative and the feedback is not in a stable condition. In this stage there is need of security and privacy protection.

- **Public Tweets**PT (the default setting) PT is a default setting which are visible to anyone, whether or not they have a account. They do not consider any followers or do not calculate the user needs. In public tweets the comments flow are also publicized [6].
- **Protected Tweets**may only be visible to your friends and approves only Twitter followers. Only particular users are granted permission to follow the tweets we post. If others try to interrupt an

indication to admin will be delivered and the admin may block that particular user tweet ID and details [7].

The proposed solutions employ different techniques measuring Web service reputations based on user feedback ratings regarding abusing words or comments. We validate our proposed malicious feedback rating prevention scheme through theoretical analysis, and also evaluate our proposed measurement. A reputation derivation model had also been proposed to aggregate feedbacks into a reputation value that better reflects the behaviour of the service at selection time. The proposed method reduces the abnormality of the reputation measurement. The success ratio of the web service recommendation can be improved [8].

In service-oriented environments where honest and malicious service providers co-exist, finding the exact balance between fairness and accuracy for abusing comments bootstrapping is non-trivial. For instance, a malicious service provider may attempt to clear its (negative) Abusing comments history by discarding its original identity and entering the system with a new one. In contrast, a service provider may be entering the system for the first time without any malicious motives [9]. This can also be avoided and protection can be provided.

5.NATURAL LANGUAGE PROCESSING (NLP) AS MALICIOUS FEEDBACK DETECTOR

Natural Language Processing is defined as a language detector otherwise known as malicious feedback detector. The detector is used to identify the abusing comments provided by other crackers. NLP acts as a language processing detector. NLP at first trace for harsh or abusing comments it validate through indication meter. In indication meter if the indication points to positive then the comments validated are positive so the server allows the comments to be posted. If the indication points towards negative sign then there is detection of abused word usage. In that case the server will be blocked the particular comment is kept for verification in the main server. If the admin allows/permit then the verification is done again and the particular comment is revaluated [10]. But in this process it never allows the comment to pass through or to be posted. Steps followed in detection of malicious feedback/abusing word.

a) Indication check: Check for the indication positive allows user to post comment. Negative indication then do not permit user to post the malicious feedback.
b) Admin verification: At once when the malicious feedback is detected pin point the comment. Picks out the abusing comment sent to main server for Admin verification.
c) Provision to permit: the abusing comment is detected through NLP and the language processing transfers automatically to admin for verification not allowing the server to tweet the abusing comment/post in the twitter.
d) Block the post: if the doubt is clarified and the comment is abused then at once the admin block the post/malicious feedback from the user. The admin also pick out and block or mark list the particular user from twitter.

By this the malicious feedback will not be posted. This protects the other user from quarrel and fights. When the abusing comment or malicious feedback is blocked there won't be any problem occurrence relating the feedback [11].

6.CONCLUSION

The service comment (positive & negative comments) score is usually calculated using feedback ratings provided by users. Although the reputation measurement of Web service has been studied in the recent literature, existing malicious and subjective user feedback ratings often lead to a bias that degrades the performance of the service recommendation system. In this paper, we propose a safer comment passing for twitter by using Natural language processing (NLP) measurement approach for Web service recommendations. In this proposed system the feedback measurement in the twitter approach utilizes malicious feedback rating detection and also feedback similarity computation to measure the reputation and harmful quarrel ob web services in common. The prevention scheme can also identify the IP address with abusing/offending comment ratings and block them using the NLP [12]. NLP as a detecting technique finds the wrong comment and transact the comment to admin for verification. And blocks the abusing feedback ratings inside the user web recommended system and protect the user.

7.Reference

[1] X. Chen, X. Liu, Z. Huang, and H. Sun. Region KNN: A scalable hybrid collaborative

filtering algorithm for personalized web service recommendation, In Proceedings of the 8th IEEE International Conference on Web Services (ICWS'10), pages 9- 16, 2010.

[2] Z. Zheng and M. R. Lyu.Collaborative reliability prediction of service-oriented systems.In Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10), pages35-44, 2010.

[3] E. M. Maximilien and M. P. Singh.Conceptual model of web service reputation. SIGMOD Record: 31(4): 36-41, 2002.

[4] Z. Malik and A. Bouguettaya.Evaluatingrater credibility for reputation assessment of web services. In Proceedings of the 8 th International Conference on Web Information Systems Engineering (WISE'07), pages 38-49, 2007.

[5] Z. Xu, P. Martin, W. Powley, and F. Zulkernine. Reputation enhanced QoS-based web services discovery. In Proceedings of the IEEE International Conference on Web Services (ICWS'07),pages 249-256, 2007.

[6] D. Ardagna and B. Pernici. Adaptive service composition inflexible processes. IEEE Transactions on Software Engineering, 33(6): 369-384, 2007.

[7] W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, and K.Nahrstedt. A trust management framework for service oriented environments. In Proceedings of the 18th international conference on World Wide Web (WWW'09), pages 891-900, 2009.

[8] S. Nepal, Z. Malik, and A. Bouguettaya. Reputation Propagation in Composite Services. In Proceedings of the IEEE International Conference on Web Services (ICWS'09), pages 295-302, 2009.

[9] R. Jurca, B. Faltings, and W. Binder. Reliable QoS monitoring based on client feedback. In Proceedings of the 16th international conference on World Wide Web (WWW'07), pages 1003-1012, 2007.

[10] N. Limam and R. Boutaba.Assessing Software Service Quality and Trustworthiness at Selection Time. IEEE Transactions on Software Engineering, 36(4): 559-574, 2010.