International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 10 | Oct 2020 www.irjet.net p-ISSN: 2395-0072

Dynamic Sentiment Analysis using Machine Learning Techniques

Vinaya Vinod Kamat[1]

[1]M.Tech Student, Dept. of Information Science Engineering, DR. Ambedkar Institute of Technology, Bangalore, Karnataka, India.

---***---

Abstract - *This project addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral. Twitter is an online micro-blogging and social-networking platform which allows users to write short status updates of maximum length 140 characters. It is a rapidly expanding service with over 200 million registered users [24] - out of which 100 million are active users and half of them log on twitter on a daily basis - generating nearly 250 million tweets per day [20]. Due to this large amount of usage we hope to achieve a reflection of public sentiment by analyzing the sentiments expressed in the tweets. Analyzing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.*

Key Words: Sentiment analysis, Supervised Machine learning, Twitters.

1. INTRODUCTION

In the past few years, there has been an enormous growth within the use of microblogging platforms like Twitter. Spurred by that growth, firms and media organizations are more and more seeking ways in which to mine Twitter for info concerning what folks assume and feel concerning their merchandise and services. Companies like Twitratr (twitrratr.com), tweetfeel (www.tweetfeel.com), and Social Mention (www.socialmention.com) are simply a number of United Nations agency advertise Twitter sentiment analysis joined of their services.

While there has been a good quantity of analysis on however sentiments are expressed in genres like online reviews and news articles, however sentiments are expressed given the informal language and message-length constraints of microblogging has been a lot of less studied. Options like automatic part-of-speech tags and resources like sentiment lexicons have tested helpful for sentiment analysis in different domains, however can they additionally prove helpful for sentiment analysis in Twitter? During this project, we start to analyse this question.

Another challenge of microblogging is that the unbelievable breadth of topic that's lined. It's not associate exaggeration to mention that folks tweet concerning something and everything. Therefore, to be ready to build systems to mine Twitter sentiment concerning any given topic, we want a way for quickly distinguishing data that may be used for coaching. During this project, we tend to explore one methodology for building such data: using Twitter hashtags (e.g., #bestfeeling, #epicfail, #news) to spot positive, negative, and neutral tweets to use for coaching three-way sentiment classifiers.

The online medium has become a big means for folks to specific their opinions and with social media, there's associate abundance of opinion info out there. Exploitation sentiment analysis, the polarity of opinions may be found, like positive, negative, or neutral by analyzing the text of the opinion. Sentiment analysis has been helpful for firms to induce their customer's opinions on their merchandise predicting outcomes of elections and obtaining opinions from picture show reviews.

The information gained from sentiment analysis is beneficial for firms creating future selections. Many ancient approaches in sentiment analysis uses the bag of words methodology. The bag of words technique doesn't contemplate language morphology, and it might incorrectly classify 2 phrases of having an equivalent which means as a result of it might have an equivalent bag of words. The link between the collection of words is taken into account rather than the link between individual words. When determining the sentiment, the sentiment of every word is set and combined exploitation a function. Bag of words additionally ignores ordination, that ends up in phrases with negation in them to be incorrectly classified. Different techniques mentioned in sentiment analysis embrace Naive mathematician, Maximum Entropy, and Support Vector Machines. Within the Literature Survey section, approaches used for sentiment analysis and text classification are summarized.

Sentiment analysis refers to the broad space of language process that deals with the mathematical study of opinions, emotions and sentiments that are expressed in text. Sentiment Analysis (SA) or Opinion Mining (OM) aims at learning people's opinions, attitudes and emotions towards an entity. The entity will represent people, events or topics. Associate huge quantity of analysis has been performed within the space of sentiment analysis. However, most of them targeted on classifying formal and bigger items of text knowledge like reviews.

This paper addresses the problem of sentiment analysis in twitter; that is classifying tweets according to the sentiment expressed in them: positive, negative or neutral. In this task, we will play out the wistful examination of the tweets from the web-based life stage Twitter. The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

1.1 Background

First, emoticons, which were considered important by other works in the area (see below) couldn't be used to learn sentiments. Secondly, the authors are unsure if all tweets with ":)" are truly positive or can contain negative or sarcastic sentiments too. Therefore, the dataset used in the experiment was labelled noisy. The task of classification was limited to positive and negative though the need for neutral class was highlighted in the end. Go et al. (2009) used SVM, MaxEnt and Naive Bayes and reported that SVM and Naive Bayes were equally good and beat MaxEnt. The research also found POS (parts of speech) tags were not helpful for their purpose. However, they only tried general purpose POS 4 tagger which is shown to have inaccuracies when ran on microblogging text.

Another popular work in this topic was conducted by Pak and Paroubek (2010). The same distant supervision procedure of tagging tweets positive or negative based on the emoticons it mentions was used. However, highlighting the significance of neutral class, the team also collected neutral tweets. These tweets were strictly objective and were collected from newspapers and magazines. Unlike Go et al. (2009), POS features were deemed useful. However, Tree Tagger (Schmid, 1995) was used which wasn't designed to work with microblogging text like tweets. The research didn't use auto annotated tweets in the test set. 5 Instead, a small size test set was hand annotated (216 tweets). Like Go et al. (2009), Pak and Paroubek (2010) also used linear kernel SVM to run the experiment. The results were not reported as accuracy metric reported by Go was different and hence the results were not comparable to previous research work.

Major boost in the discussed field came as a result of annual SemEval workshop (Nakov et al., 2013). The 6 workshop had more than 30 entries in 2013 when the organizers first introduced twitter sentiment analysis as one of their exercises. Not only this provided opportunity for more frequent research in the area, it provided large dataset of hand annotated tweets with all three sentiment classes positive, negative and neutral (see Section 4.1). The consistent scoring metric was also available for the community to gauge their research with other teams in the field.

The best performing system in SemEval 2013 workshop was by Mohammad et al. (2013). The work utilized the hand annotated datasets by workshop organizers and build a classifier that does the same job as the tool built in this project. Unlike previous work, this system made use of POS tagger and tokenizer designed specifically for Twitter (Gimpel et al., 2011). The feature set included a large number of features with some focusing on the specific nature of tweet like the use of elongated words, emoticons, all caps words, URL link etc. They also relied heavily on lexicons that assigned a sentiment to a token as a positive real number (if a positive word) or negative. These features are called lexical features. It was reported that after bag of words (that almost all research works have used), lexical features were most useful. Like the aforementioned research work, a linear kernel SVM was used. The results are published using the metric provided by workshop organizers: the average F1 measure of positive and negative class (see Section 5.1). With all features included, the system received the score of 69.02 on Mohammad et al. (2013) provided test set. The best performing system in the second run of SemEval workshop (Rosenthal et al., 2014) was by TeamX (Yasuhide, 2014). Building on the system reported by Mohammad et al. (2013), this system also relied on lexical features. The system introduced the concept of cost sensitive classification in the task of Twitter sentiment classification. They found the training data to be unbalanced, biasing the learner towards neutral class. To account for this bias, the paper proposed to assign higher penalty for misclassification of polar classes. This was reported to nullify the bias and increase overall accuracy.

Looking at the broader field of sentiment analysis of any text, results reported by Pang et al. (2002) were insightful for our research. First, they compared different machine learning algorithms to suggest which classification algorithms work better for this type of text categorization and reasoning for their better performance. Also, the challenges posed by the sentiment classification were highlighted which helped us understand this task better and then building the right feature vector for our system.

In summary, we reviewed numerous papers that highlighted the positive impact of lexical features. Each system used their own methodology to represent these lexical features in the feature vector. We noted that most papers combined these lexical features into one score that was added as a value in the feature vector. There wasn't enough work that explored the idea of converting this lexical analysis into numerous features that captures the independent nature of lexical features and let the classification algorithm deduce the relationship between these features using the training data. We explored this promising area of lexical features in depth through this research. Also, unlike research done earlier, this work uses a very conservative approach in adding features to the system. We also build on Yasuhide (2014) ideas about cost sensitive classification and identify areas of improvement. Most of the remaining aspects of our system were inspired by earlier reported works that are cited in this section.

1.2 Objectives

- The objective of this paper is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.
- Formally, given a training sample of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist, your objective is to predict the labels on the given test dataset.

2. PROPOSED WORK

Sentiment Analysis can be characterized as a procedure that robotizes mining of mentalities, feelings, perspectives and feelings from content, discourse, tweets and database sources through Natural Language Processing (NLP). Estimation examination includes characterizing feelings in content into classifications like "positive" or "negative" or "nonpartisan". It's likewise alluded as subjectivity examination, feeling mining, and evaluation extraction.

This work tends to cover the issue of opinion investigation in twitter; that is characterizing tweets as indicated by the slant communicated in them: positive, negative, or nonpartisan. Twitter is an online small-scale blogging and long-range informal communication stage which permits clients to compose short notices of most extreme length 140 characters. It is a quickly extending administration with more than 200 million enlisted clients [24] - out of which 100 million are dynamic clients and half of them sign on twitter consistently - producing about 250 million tweets for each day [20]. Because of this huge measure of use we plan to accomplish an impression of open opinion by breaking down the slants communicated in the tweets.

Tweets are gathered utilizing Tweepy library followed by the choice of helpful highlights to my assignment and transformation to CSV. At that point, information cleaning is performed like expelling URLs, retweet images, username-labels, and hashtags. Sent WordNet dictionary is utilized to name the opinion of the tweets. Steps like Stop words evacuation, Lemmatizing, Stemming are performed on the content information. Afterward, WordCloud is utilized for Data Visualization. In the wake of parting the information, Count Vectorizer and Tfidf Vectorizer are utilized for scientific portrayal of the content. At that point, nine grouping calculations are actualized on the vectors acquired already. Afterward, doc2vec models (DBOW, DMC, DMM) are prepared and utilized vectors (got through these models) for characterization reason as these models save semantic connections between words. At long last, the best model is assessed utilizing the test information. The following figure 4.1 shows the proposed architecture:

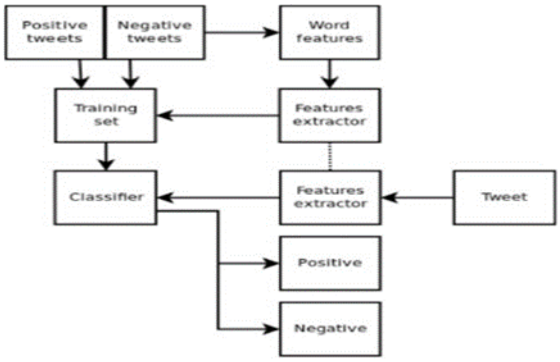

Figure 4.1 Proposed Architecture

The design flow of the proposed work is as follow:

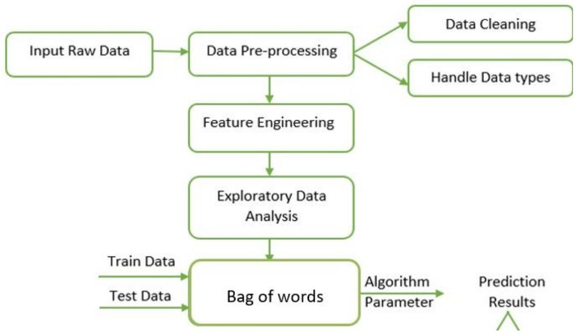

Figure 4.2 Design flow

The process of designing a functional classifier for sentiment analysis can be broken down into five basic categories. They are as follows:

- Data Acquisition
- Data Pre-processing
- Feature Extraction
- Classification
- Tweet Classifer Web app

Data Acquisition: Data in the form of raw tweets is acquired by using the python library "tweestream" which provides a package for simple twitter streaming API. This API allows two modes of accessing tweets: SampleStream and FilterStream. SampleStream simply delivers a small, random sample of all the tweets streaming at a real time. FilterStream delivers tweet which match a certain criteria. It can filter the delivered tweets according to three criteria:

- Specific keyword(s) to track/search for in the tweets.
- Specific Twitter user(s) according to their user-id's Tweets originating from specific location(s) (only for geo-tagged tweets).
- A programmer can specify any single one of these filtering criteria or a multiple combination of these. But for our purpose we have no such restriction and will thus stick to the SampleStream mode.

Data Pre-processing

It is significant advance before further handling; it channels the audits with the goal that it improves precision and furthermore expels pointless aggravations. It incorporates disposal of stop words. Uncommon characters and furthermore Unicode characters like ❤,✓,☀,★, ☂, ♞, ☯ and so on are evacuated before notion investigation. Likewise front and back slang, wrong spelling, URL's, RT, @, hash labels # are completely expelled. The inquiries, for example, what, which, how and so forth., are not assuming a job for figuring the extremity of sentence consequentlytheseare evacuated. At the point when information from twitter is recovered then different tweets recovered more than each in turn time, so it is important to evacuate copy line for time multifaceted nature. Tweets that are in capitalized convert to that in lower case, so it makes simple for correlation with seed words.

Preprocessing of tweet incorporate after focuses,

- Remove all URLs (for example www.xyz.com), hash labels (for example #topic), targets (@username)
- Correct the spellings; grouping of rehashed characters is to be taken care of
- Replace all the emojis with their assumption. Remove all accentuations, images, numbers
- Remove Stop Words
- Expand Acronyms(we can utilize an abbreviation word reference)
- Remove Non-English Tweets

Feature Engineering

1. Tokenisation

Tokenization is the way toward partitioning the given content or sentence into tokens and tokens are might be word, expression, characters or other significant unit. The case of tokenization is as per the following:

Info: Heavy congested driving conditions at JM street because of mishap. Yield: [Heavy | traffic | jam | at | JM | street | due | to | accident]

Tokenization is the showing of isolating a progression of strings into pieces, for instance, words, expressions, images and various segments called tokens.

2. Pos Tagging

Preparing information for the most part takes a great deal of work to make, so a prior corpus is commonly utilized. These typically utilize the Penn Treebank. We are finished with the fundamental cleaning some portion of content information. In the ML calculations that we are going to actualize 'full text' of the tweet goes about as an indicator variable(other factors that can be utilized are retweet-check, top choice tally on the off chance that we need to foresee the effect of a tweet, however that is not a piece of our undertaking as of now). As it is obvious, we have to make target factors (notion scores) for our information. For this reason, we use SentiWordNet.

SentiWordNet is an improved lexical asset unequivocally concocted for supporting slant grouping and conclusion mining applications. It has an enormous corpus of POS-labeled English words alongside their assumption. POS labeling of unrefined substance is a significant structure square of various NLP pipelines, for instance, word-sense disambiguation, question taking note of and feeling examination. In its least troublesome structure, given a sentence, POS labeling is the task of perceiving things, action words, descriptive words, verb modifiers, and the sky is the limit from there.

3. Bag of Words Model

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order. The bag-of-words model is commonly used in methods of document classification, where the (frequency of) occurrence of each word is used as a feature for training a classifier.

Most commonly, we use a word list where each word has been scored. Positivity/negativity or sentiment strength and overall polarity is determined by the aggregate of polarity of all the words in the text.

Machine Learning Classification

Machine learning prediction has these following steps:

1. Split data into training and test sets.

2. Defining the algorithms namely Decision tree algorithm.

3. Training and testing against the algorithms.

4. Updating the User Interface with the calculated values.

Split data into training and test sets

1. Divide the available data into parts in a certain ratio.

2. Train the algorithm on the X% of the actual data and test on the remaining (100-X) % of the data

3. RESULTS AND EVALUATION

In the evaluation, we want to understand, for a number of metrics, whether our method works well for the problem statement we are trying tackle. I identify the polarity and classify the sentiments into whether positive, negative or neutral.

Figure 5.1 shows the sample dataset used.

A	B	C
id	label	tweet
1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction.
2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx.
3	0	bihday your majesty
4	0	#model i love u take with u all the time in urÃ°ÂŸÂ"Â±!!! Ã°ÂŸÂ˜Â™Ã°ÂŸÂ˜ÂŽÃ°ÂŸÂ'Â„Ã°ÂŸÂ'
5	0	factsguide: society now #motivation
6	0	[2/2] huge fan fare and big talking before they leave. chaos and pay disputes when they get t
7	0	@user camping tomorrow @user @user @user @user @user @user @user dannyÃ¢Â€Â¦
8	0	the next school year is the year for exams.Ã°ÂŸÂ˜Â¯ can't think about that Ã°ÂŸÂ˜Â #school
9	0	we won!!! love the land!!! #allin #cavs #champions #cleveland #clevelandcavaliers Ã¢Â€Â¦
10	0	@user @user welcome here ! i'm it's so #gr8 !

Figure 5.1 Dataset used

Figure 5.2 shows the distribution of Positive and Negative reviews against the count of their occurrences

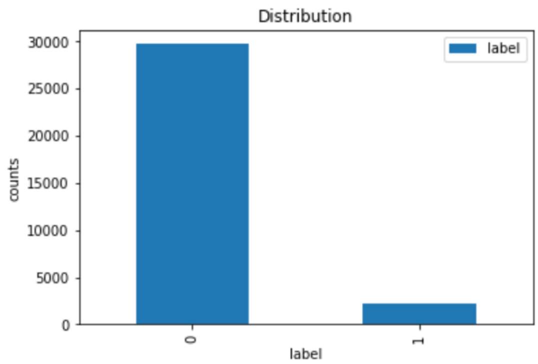

Figure 5.2 Distribution of Positive and Negative reviews against the count of their occurrences

Figure 5.3 shows the distribution of training and testing tweets based on their length

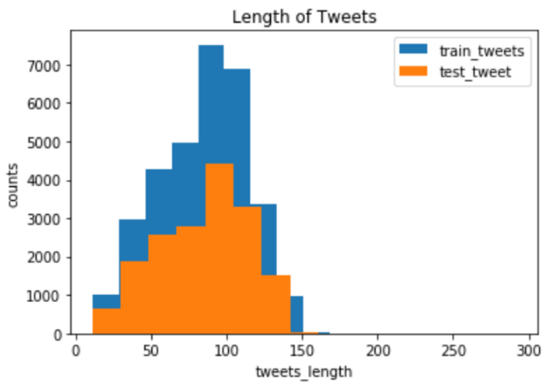

Figure 5.3 Distribution of training and testing tweets based on their length

Figure 5.4a shows the removal of short words from the original tweets

	id	label	tweet	tidy_tweet
0	1	0.0	@user when a father is dysfunctional and is s...	when father dysfunctional selfish drags kids i...
1	2	0.0	@user @user thanks for #lyft credit i can't us...	thanks #lyft credit cause they offer wheelchai...
2	3	0.0	bihday your majesty	bihday your majesty
3	4	0.0	#model i love u take with u all the time in ...	#model love take with time
4	5	0.0	factsguide: society now #motivation	factsguide society #motivation

Figure 5.4b shows the result of the tokenization

```
0    [when, father, dysfunctional, selfish, drags, ...
1    [thanks, #lyft, credit, cause, they, offer, wh...
2                               [bihday, your, majesty]
3                        [#model, love, take, with, time]
4                      [factsguide, society, #motivation]
Name: tidy_tweet, dtype: object
```

Figure 5.4c shows the results of stemming

```
0    [when, father, dysfunct, selfish, drag, kid, i...
1    [thank, #lyft, credit, caus, they, offer, whee...
2                               [bihday, your, majesti]
3                        [#model, love, take, with, time]
4                            [factsguid, societi, #motiv]
Name: tidy_tweet, dtype: object
```

Figure 5.5 shows the TF-IDF model pseudocode

```
In [27]:  train_tfidf = tfidf[:31962,:]
          test_tfidf = tfidf[31962:,:]

          xtrain_tfidf = train_tfidf[ytrain.index]
          xvalid_tfidf = train_tfidf[yvalid.index]

          lreg.fit(xtrain_tfidf, ytrain)

          prediction = lreg.predict_proba(xvalid_tfidf)
          prediction_int = prediction[:,1] >= 0.3
          prediction_int = prediction_int.astype(np.int)

          f1_score(yvalid, prediction_int)

Out[27]: 0.5091240875912408
```

Figure 5.6 and 5.7 represents the output of the User Input using the proposed model.

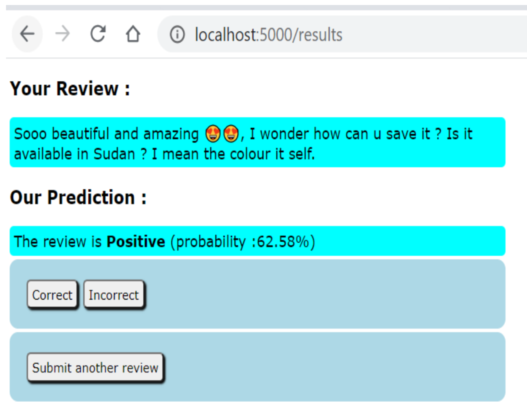

Figure 5.6 Positive Review

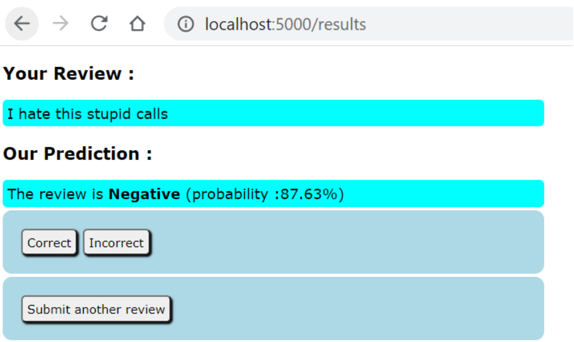

Figure 5.7 Negative Review

4. CONCLUSIONS

During this research, we achieved two goals in the areas of machine learning and interactive visualizations. First, we developed state of the art Twitter sentiment prediction system using supervised text classification technique and support vector machines. A mixture of proven published ideas (bag of words, lexical features etc.) and novel ideas (polarity buckets, weight adjusted negation etc.) contributed in achieving a high-performance system. Cost sensitive classification technique helped in removing bias towards neutral class due to unbalanced training data.

On SemEval 2013 test dataset, we achieved the score (average polar F measure) of 72.25, higher than any published system. Second, we used this sentiment prediction model to build interactive visualization tool empowering brand managers to visualize and interpret public sentiments regarding their brand in real time. The system was built using design principles guided by Chuang et al. (2012).

This tool visualized sentiments over the dimensions of time, location, platform and influencing power of a user. The tool could take in search queries to update all the visualizations. These queries can limit the data used for visualization. The adaptive dashboard enabled the system to update visualization every 5 seconds with incoming stream of Tweets. Several aspects of the system can be improved with further commitment to this research.

(1) In building prediction model, an area of ensemble machine learning is trending. Similar tasks have reported performance boost using this technique.

(2) Hashtags can significantly help in identifying tweet sentiment. However, since they are compound words, the system was not able to leverage their full potential in indicating sentiment.

(3) We reported high performance increase using sentiment lexicons. Since some of these lexicons are automatically compiles and contain some noise, there is a chance of further performance boost if this noise can be reduced. We also did analyses on emojis and build them into an emotion dictionary which is a good way to improve the sentiment analysis of Twitter data.

5. FUTURE WORK

Right now, we are exploring Parts of Speech separate from the unigram models, we can try to incorporate POS information within our unigram models in future. So say instead of calculating a single probability for each word like P(word | obj) we could instead have multiple probabilities for each according to the Part of Speech the word belongs to. For example we may have P(word | obj, verb), P(word | obj, noun) and P(word | obj, adjective). Pang et al. used a somewhat similar approach and claims that appending POS information for every unigram results in no significant change in performance (with Naive Bayes performing slightly better and SVM having a slight decrease in performance), while there is a significant decrease in accuracy if only adjective unigrams are used as features.

However, these results are for classification of reviews and may be verified for sentiment analysis on micro blogging websites like Twitter. One more feature we that is worth exploring is whether the information about relative position of word in a tweet has any effect on the performance of the classifier. Although Pang et al. explored a similar feature and reported negative results, their results were based on reviews which are very different from tweets and they worked on an extremely simple.

REFERENCES

[1] Batuwita, R., & Palade, V. (2012). Class imbalance learning methods for support vector machines.

[2]Białecki, A., Muir, R., & Ingersoll, G. (2012). Apache lucene 4. In SIGIR 2012 workshop on open source information retrieval (pp. 1724).

[3]Bin wasi, S., Neyaz, R., Bouamor, H., Mohit, B. (2014). CMUQ@Qatar: Using Rich Lexical Features for Sentiment Analysis on Twitter. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). Association for Computational Linguistics.

[4]Bollen, J., Pepe, A., & Mao, H. (2009). Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena. arXiv preprint arXiv:0911.1583.

[5] B. Pang, L. Lee, S. Vaithyanathan (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 79–86).

[6] Chuang, J., Ramage, D., Manning, C., & Heer, J. (2012). Interpretation and trust: Designing modeldriven visualizations for text analysis. InProceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 443452). ACM. Councill, I. G.,

[7] McDonald, R., & Velikovich, L. (2010). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In Proceedings of the workshop on negation and speculation in natural language processing (pp. 5159). Association for Computational Linguistics.

[8] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research, 9, 18711874.

[9] Fisher, D., DeLine, R., Czerwinski, M., & Drucker, S. (2012). Interactions with big data analytics. interactions, 19(3), 5059.

[10] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., ... & Smith, N. A. (2011, June). Partofspeech tagging for twitter: Annotation, features, and experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papersVolume 2 (pp. 4247). Association for Computational Linguistics. Go, A., Bhayani, R., & Huang, L. (2009).

[11] Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 112. Günther, T., & Furrer, L. (2013). Gumltlt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent.

[12] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 1018. Hu, M., & Liu, B. (2004, August).

[13] Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 168177). ACM.

[14] Kavitha, A. S., Kavitha, R., & Viji Gripsy, J. (2012). Empirical Evaluation of Feature Selection Technique in Educational Data Mining. ARPN Journal of Science and Technology, 2(11).

[15] Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research, 723762. Kong, L., Schneider, [16] N., Swayamdipta, S., Bhatia, A., Dyer, C., Smith, N. A. (2014). A Dependency Parser for Tweets. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

[17] Mohammad, S., Kiritchenko S., and Zhu, X. 2013. NRC-Canada: Building the stateoftheart in sentiment analysis of tweets. In Proceedings of the Seventh international workshop on Semantic Evaluation Exercises, SemEval2013, pages 321–327, Atlanta, Georgia, USA.