Machine Learning based Twitter Sentimental Analysis in Business Field

Rohit Ningappa Padti, Shashank H G, Syed Azam H S, Vignesh Pai, Ramesh B

Abstract*— Social networking sites like twitter have millions of people share their thoughts day by day as tweets. 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. 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, 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. 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 project is to develop a functional classifier for accurate and automatic sentiment classification of an unknown tweet stream.

Keywords—sentiment analysis, micro-blogging, socioeconomic

I. INTRODUCTION

The social networking sites like Twitter, Facebook, and YouTube have obtained so much popularity now days. The process of analyzing sentiments of tweets comes under the domain of "Pattern Classification" and "Data Mining". The project would heavily rely on techniques of "Natural Language Processing" in extracting significant patterns and features from the large data set of tweets and on "Machine Learning" techniques for accurately classifying individual unlabeled data samples (tweets) according to whichever pattern model best describes them. The features that can be used for modelling patterns and classification can be divided into two main groups: formal language based, and informal blogging based.

Revised Manuscript Received on December 14, 2019.

Dr. Ramesh B Professor in Department of CSE, Malnad College of Engineering, Hassan.

Rohit Ningappa Padti, Department of CSE, Malnad College of Engineering, Hassan.

Shashank H G, Department of CSE, Malnad College of Engineering, Hassan.

Syed Azam H S, Department of CSE, Malnad College of Engineering, Hassan.

Vignesh Pai, Department of CSE, Malnad College of Engineering, Hassan.

Language based features are those that deal with formal linguistics and include prior sentiment polarity of individual words and phrases, and parts of speech tagging of the sentence. Prior sentiment polarity means whenever a word with positive feeling is used in a sentence, the entire sentence would be expressing a positive sentiment. On the other hand, parts of speech tagging automatically identify which part of speech each individual word of a sentence belongs to: noun, pronoun, verb etc. Patterns can be extracted from analyzing the frequency distribution of these parts of speech in a class of labelled tweets. Classification techniques are divided into two categories: Supervised vs. unsupervised and non-adaptive vs. adaptive techniques. Supervised approach is when we have pre-labelled data samples available and we use them to train our classifier. Unsupervised classification is when we do not have any labelled data for training. Adaptive classification techniques deal with feedback from the environment. In our case feedback from the environment can be in form of a human telling the classifier whether it has done a good or poor job in classifying a tweet and the classifier needs to learn from this feedback. There are two further types of adaptive techniques: Passive and active. Passive techniques are the ones which use the feedback only to learn about the environment but not using this improved learning in our current classification algorithm, while the active approach continuously keeps changing its classification algorithm according to what it learns at real-time.

II. LITERATURE SURVEY

Alexander Pak and Patrick Paroubek [5] proposed the method where they automatically collect a corpus for sentiment analysis and opinion mining purposes in twitter. Then perform linguistic analysis of the collected corpus and explain discovered phenomena. Using the corpus, they build a sentiment classifier, which is able to determine positive, negative and neutral sentiments for a document.

Efthymios and Theresa Wilson [6] investigated the utility of linguistic features for detecting the sentiment of Twitter messages. They evaluated the usefulness of existing lexical resources as well as features that capture information about the informal and creative language used in microblogging.

They took a supervised approach to the problem, but leverage existing hashtags in the Twitter data for building training data. Their experiments on twitter sentiment analysis show that part-of-speech features may not be useful for sentiment analysis in the microblogging domain. More research is needed to determine whether the POS features are just of poor quality due to the results of the tagger or whether POS features are just less useful for sentiment analysis in this domain. Features from an existing sentiment lexicon were somewhat useful in conjunction with microblogging features, but the microblogging features (i.e., the presence of intensifiers and positive/negative/neutral emoticons and abbreviations) were clearly the most useful.

Peter Turney [8] presented a simple unsupervised learning algorithm for classifying reviews as recommended (thumbs up) or not recommended (thumbs down). The classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. In his paper the semantic orientation of a phrase is calculated as the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". A review is classified as recommended if the average semantic orientation of its phrases is positive. The algorithm achieves an average accuracy of 74%.

Stefano and Andrea [7] presented SENTIWORDNET 3.0, a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications. SENTIWORDNET 3.0 is an improved version of SENTIWORDNET 1.0, a lexical resource publicly available for research purposes. SENTIWORDNET is the result of the automatic annotation of all the synsets of WORDNET according to the notions of "positivity", "negativity", and "neutrality". Each synset is associated to three numerical scores Pos(s), Neg(s), and Obj(s) which indicate how positive, negative, and "objective" (i.e., neutral) the terms contained in the synset are. At the end they evaluated results of both version and indicate accuracy improvements of about 20% with respect to SENTIWORDNET 1.0.

Theresa and Janyce [9] presented a new approach to phrase-level sentiment analysis that first determines whether an expression is neutral or polar and then disambiguates the polarity of the polar expressions. With this approach, the system is able to automatically identify the contextual polarity for a large subset of sentiment expressions, achieving results that are significantly better than baseline.

Alec and Richa [10] introduced a novel approach for automatically classifying the sentiment of Twitter messages. These messages are classified as either positive or negative with respect to a query term. They presented the results of machine learning algorithms for classifying the sentiment of Twitter messages using distant supervision. Their training data consists of Twitter messages with emoticons, which are used as noisy labels. They showed that machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) have accuracy above 80% when trained with emoticon data. They also described the pre-processing steps needed in order to achieve high accuracy. The main contribution of Alec and Janyce is the idea of using tweets with emoticons for distant supervised learning.

III. SENTIMENT ANALYSIS

The process of designing a functional classifier for sentiment analysis can be broken down

Into four basic categories. They are as follows:

A. Data Acquisition
B. Human Labelling
C. Feature Extraction
D. Classification

Data Acquisition:

Data in the form of raw tweets is acquired by using the python library "**tweet stream**" which provides a package for simple twitter streaming API. This API allows two modes of accessing tweets: Sample Stream and Filter Stream. Sample Stream simply delivers a small, random sample of all the tweets streaming at a real time. Filter Stream delivers tweet which match a certain criterion. A tweet acquired by Sample Stream method has a lot of raw information in it which we may or may not find useful for our particular application. It comes in the form of the python "dictionary" data type with various key-value pairs. A list of some key-value pairs are given here: Whether a tweet has been favorited, User ID, Screen name of the user, Original Text of the tweet, Presence of hashtags, Whether it is a re-tweet, Language under which the twitter user has registered their account, Geo-tag location of the tweet, Date and time when the tweet was created. Since this is a lot of information, we only filter out the information that we need and discard the rest. For our particular application we iterate through all the tweets in our sample and save the actual text content of the tweets in a separate file given that language of the twitter is user's account is specified to be English.

Human iLabelling:

In human labelling there are three copies of the tweets so that they can be labelled by four individual sources. This is done so that we can take average opinion of people on the sentiment of the tweet and in this way the noise and inaccuracies in labelling can be minimized.

Labelling of tweets in four classes according to sentiments expressed/observed in the tweets: positive, negative, neutral/objective and ambiguous.

Positive: If the entire tweet has a positive/happy/excited/joyful attitude or if something is mentioned with positive connotations. Also if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant.

Negative: If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant.

Neutral/Objective: If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information. Advertisements of different products would be labelled under this category.

Ambiguous: If more than one sentiment is expressed in the tweet which are equally potent with no one particular sentiment standing out and becoming more obvious. Also if it is obvious that some personal opinion is being expressed here but due to lack of reference to context it is difficult/impossible to accurately decipher the sentiment expressed.

<Blank>: Leave the tweet unlabeled if it belongs to some language other than English so that it is ignored in the training data.

Once we had labels from four sources our next step was to combine opinions of three people to get an averaged opinion. This can be done through majority vote.

Feature Extraction:

In this process we need to extract useful features from it which can be used in the process of classification. For this we use some text formatting techniques such as Tokenization, Lowercase Conversion, Stemming, Stop-words removal, Parts-of-Speech Tagging etc. There are two kinds of classification in our system, the objectivity / subjectivity classification and the positivity / negativity classification. As the name suggests the former is for differentiating between objective and subjective classes while the latter is for differentiating between positive and negative classes.

Next we will calculate the unigram word models using Naive Bayes. The basic concept is to calculate the probability of a word belonging to any of the possible classes from our training sample. Using mathematical formulae we were calculating probability of word belong to objective and subjective class. Similar steps would need to be taken for positive and negative classes as well.

The list of features explored for objective / subjective classification is as below:

- Number of exclamation marks in a tweet
- Number of question marks in a tweet
- Presence of exclamation marks in a tweet
- Presence of question marks in a tweet
  Presence of url in a tweet
- Presence of emoticons in a tweet
- Unigram word models calculated using Naive Bayes
- Prior polarity of words through online lexicon MPQA
- Number of digits in a tweet
- Number of capitalized words in a tweet
- Number of capitalized characters in a tweet
- Number of punctuation marks / symbols in a tweet

The list of features explored for positive / negative classification are given below

- Unigram word models calculated using Naive Bayes
- Number of total emoticons in the tweet
- Number of positive emoticons in a tweet
- Number of negative emoticons in a tweet
- Number of positive words from MPQA lexicon in tweet
- Number of negative words from MPQA lexicon in tweet
- Number of base-form verbs in a tweet
- Number of past tense verbs in a tweet
- Number of present participle verbs in a tweet

Classification:

Pattern classification is the process through which data is divided into different classes according to some common patterns which are found in one class which differ to some degree with the patterns found in the other classes. The ultimate aim of our project is to design a classifier which accurately classifies tweets in the following four sentiment classes: positive, negative, neutral and ambiguous.

There can be two kinds of sentiment classifications in this area: contextual sentiment analysis and general sentiment analysis. Contextual sentiment analysis deals with classifying specific parts of a tweet according to the context provided. On the other hand general sentiment analysis deals with the general sentiment of the entire text as a whole. We prefer general sentiment analysis. The classification approach generally followed in this domain is a two-step approach. First Objectivity Classification is done which deals with classifying a tweet or a phrase as either objective or subjective. After this we perform Polarity Classification to determine whether the tweet is positive, negative or both. We propose a novel approach which is slightly different from the approach proposed by Wilson et al. [9]. We propose that in first step each tweet should undergo two classifiers: the objectivity classifier and the polarity classifier.

The former would try to classify a tweet between objective and subjective classes, while latter would do so between the positive and negative classes.

We use the short-listed features for these classifications and use the Naive Bayes algorithm so that after the first step we have two numbers from 0 to 1 representing each tweet. One of these numbers is the probability of tweet belonging to objective class and the other number is probability of tweet belonging to positive class. Since we can easily calculate the two remaining probabilities of subjective and negative by simple subtraction by 1, we don't need those two probabilities. So in the second step we would treat each of these two numbers as separate features for another classification, in which the feature size would be just 2. We apply the following Machine Learning algorithms for this second classification to arrive at the best result:

A. K-Means Clustering
B. Support Vector Machine
C. Logistic Regression
D. K Nearest Neighbors
E. Naive Bayes
F. Rule Based Classifiers

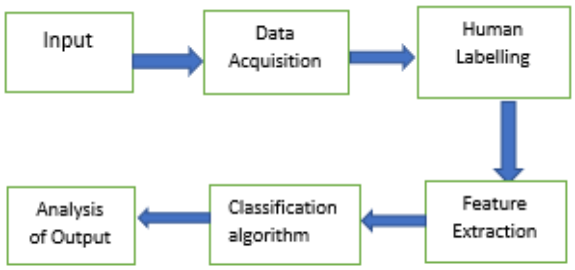

Fig 1. Block Diagram

IV. COMPARING WITH EXISTING SYSTEMS

If we compare our results to those provided by Wilson [9], we expect that the accuracy of neutral class may/may not fall if we use our classification instead of theirs. However, for all other classes we aim to report significantly greater results. Although the results presented by Wilson are not from Twitter data, they are of phrase level sentiment analysis which is very close in concept to our Twitter sentiment analysis.

Next if we compare our results with those presented by Alec Go [10], we expect that they are more or less similar. However, we plan to arrive at comparable results with just 10 features and about 9,000 training data. In contrast to this, they used about 1.6 million noisy labels. Their labels were noisy in the sense that the tweets that contained positive emoticons were labelled as positive, while those with negative emotions were labelled negative. The rest of the tweets (which did not contain any emoticon) were discarded from the data set. So, in this way they hoped to achieve high results without human labelling but at the cost of using humongous large number amount of data set.

In comparison with these results of Koulompis [6], average F-measure is of 68%. However, when they include another portion of their data into their classification process (which they call the HASH data), their average F-measure drops to 65%. In contrast to this we plan to achieve average F-measure of more than 70% which shows better performance than either of these results. Moreover, we make use of only 10 features and 9,000 labelled tweets, while their process involves about 15 features in total and more than 220,000 tweets in their training set. Our unigram word models are also simpler than theirs, because they incorporate negation into their word models. However, in his result, their tweets are not labelled by humans, but rather undergo noisy labelling in two ways: labels acquired from positive and negative emotions and hashtags.

V. RESUTLS AND DISCUSSIONS

We will first present our results for the objective / subjective and positive / negative classifications. These results act as the first step of our classification approach. We only use the short-listed features for both of these results. This means that for the objective / subjective classification we have 5 features and for positive / negative classification we have 3 features. For both of these results we use the Naive Bayes classification algorithm, because that is the algorithm, we are employing in our actual classification approach at the first step. The values we get are the result of 10-fold cross validation. We take an average of each of the 10 values we get from the cross validation.

In addition to these values, we make a condition while reporting the results of polarity classification (which differentiates between positive and negative classes) that only subjective labelled tweets are used to calculate these results. However, in case of final classification approach, any such condition is removed and basically both objectivity and polarity classifications are applied to all tweets regardless of whether they are labelled objective or subjective.

Next, we will present our results for the complete classification. We note that the best results are reached through Support Vector Machine being applied at the second stage of the classification process. Hence the results will only pertain to those of SVM. These results use a total of two features: P (objectivity | tweet) and P (positivity | tweet). But if we include all the features employed in step 1 of the classification process, we have a list of 8 shortlisted features (3 for polarity classification and 5 for objectivity classification).

VI. CONCLUSION

The task of sentiment analysis, especially in the domain of micro-blogging, is still in the developing stage and far from complete. Right now, project worked with only the very simplest unigram models. As reported in the literature review section when bigrams are used along with unigrams this usually enhances performance. Additionally, Parts of Speech separate from the unigram models are explored.

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. [5] 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.

In the paper main focus is on general sentiment analysis. There is potential of work in the field of sentiment analysis with partially known context. For example, it is noticed that users generally use this website for specific types of keywords which can divided into a couple of distinct classes, namely: politics/politicians,celebrities, products/brands, sports/sportsmen, media/movies/music. So, it is attempted to perform separate sentiment analysis on tweets that only belong to one of these classes (i.e. the training data would not be general but specific to one of these categories) and compare the results , if we apply general sentiment analysis on it instead.

Last but not the least, we can attempt to model human confidence in our system. For example, if we have 5 human labelers labelling each tweet, we can plot the tweet in the 2-dimensional objectivity / subjectivity and positivity / negativity plane while differentiating between tweets in which all 5 labels agree, only 4 agree, only 3 agree or no majority vote is reached. We could develop our custom cost function for coming up with optimized class boundaries such that highest weightage is given to those tweets in which all 5 labels agree and as the number of agreements start decreasing, so do the weights assigned. In this way the effects of human confidence can be visualized in sentiment analysis.

REFERENCES

1. RealTime Sentiment Analysis Of Twitter Posts V.Prakruthi ; D.Sindhu ; Dr.S.Anupama Kumar 2018 3rdInternational Conference on ComputationalSystems and Information Technology for Sustainable Solutions (CSITSS)
2. Sentimental analysis of Twitter corpus related to artificial intelligence assistants Chae Won Park ; Dae Ryong Seo 2018 5th International Conference on Industrial Engineering and Applications (ICIEA)
3. Sentiment analysis In Twitter Using Lexicon Based and Polarity Multiplication Kusrini ; Mochamad Mashuri 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)
4. Surveyon Sentiment analysis using Twitter Dataset Rasika Wagh ; Payal Punde 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)
5. Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.
6. Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.
7. Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010
8. Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002
9. Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In the Annual Meeting of Association of Computational Linguistics: Human Language Technologies (ACL-HLT), 2005.
10. Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

AUTHORS PROFILE

Dr. Ramesh B is currently working as Professor in Department of CSE, Malnad College of Engineering, Hassan. He has published more than 50 research papers in prominent National and International Journals. His research areas are Computer Networks, Multimedia Computing, Mobile AdHoc Networks, Cloud Computing and Network Secuirty