Detection of insult in social commentary

Parul Gupta
IIITD, India
Email: parul1370@iiitd.ac.in

Veronica Sharma
IIITD, India
Email: veronical428@iiitd.ac.in

I. Abstract

Nowadays, most of the people are using text messages as a channel for personal and professional communication. Most of the text messages are exchanged over the internet on social media. The role of technology has been a two fold sword in such scenario, as some people have used social media exchanging messages, keeping in touch with the known ones, disseminating information, sharing life's experiences and many other positive things; while others have misused it for abusing others thus escalating the hatred among the users. Therefore, it becomes a necessity to regulate the text content on social media. The "social comments" are highly unstructred, informal and often misspelled and therefore it becomes a daunting task to detect offensive comments. In this project, we try to deal with with this problem. We broadly classify the social comments in 2 main categories i.e, a positive or a negative comment based on whether the comment was insulting or not. We make use of sentiment analysis in our work nd evaluate the accuracy of the classification using 3 classifiers namely: SVM, Linear Regression and Bayes Classifier.

II. Introduction

Today, people around the globe are using various web sources to gain knowledge, share experience, keep in touch with their acquaintances, networking and many other things. In doing so, people generally post comments on web forums, social media websites like twitter; face book, newsgroups, blogs, etc. These comments may be inappropriate for the other users of the web and may hurt their feelings thus escalating feeling of hatred among the users. It may cause other users to leave the conversation and acts as an obstacle in their participation. Although various social websites providers have tried to combat this situation and their *terms of service* prevents user from posting any content which may be insulting and harass any other but they have not successfully achieved their goal as the content that is being posted only gets filtered for some particular collection of offensive words. Some web forums, Youtube, etc. have tried to trust the user and provide them with the control to flag the content which they feel is abusive and unlawful. Having human moderator for detecting insult is also not a feasible solution as insult is a highly subjective matter and the data that is being generated everyday is enormous.All these advancements have not completely filtered out the insulting content till today. Hence, the need of the hour is to have an automatic classifier which can detect insulting and non insulting comments.

In this project, we have worked towards implementing such classifier. The overall aim is to build a classifier which can successfully identify whether a social comment is an insulting comment or not. To build such classifier we make use of sentiment analysis. We obtained the annotated data from Kaggle website and used it to train and test our classifier.

III. Related Work

Finding insults in social comments is a hot area. It has attracted researchers from different research areas including machine learning [3], text mining, pattern recognition, artificial intelligence, natural language processing, etc. Mike et al.[4] presented an improved version of algorithm SentiStrength for sentiment strength detection across the social web. Chetan et al. [1] tried to identify the insults in a natural language i.e, Hindi. Priya et al.[2] extended the concept of insult detection and identified peer to peer insults in the social comments.

IV. Data Description

For our project we obtained the data from the Kaggle website. It provided us with both the training and the test data. The training data was labelled and contained nearly 4000 records whereas the test data contains 2648 labelled records. Every record of training dataset has 3 attributes associated with it: class label, date and comment. Whereas the test record has following attributes: class label, date, comment and usage.

V. Sentiment Analysis

We have performed the following techniques to find whether a social comment is an insulting comment or not.

A. *Preprocessing Techniques*

The preprocessing techniques that we have performed are:

- **Expanding Short Forms-** Our dataset contains many short forms like you're, can't, im, that's, etc. For efficient data analysis we need to convert these short forms into their full forms like you are, can not, i am, that is respectively. We manually created the dictionary of such expansions as per our data set and tried to incorporate some other commonly used short forms.
- **Removing HTML tags and URLS-** Since HTML tags and URLs are of no use in our evaluation of comment we have removed them from our data.
- **Removing special characters-** Special characters like.,[](), etc should be removed in order to remove discrepancies during the assignment of polarity. We also deleted new line characters from the data.

- **Removing punctuations-** We removed the punctuations from our data set. In our current implementation punctuations do not play any role in determining the sentiment of a comment. We also removed digits from our data. Simultaneous multiple occurrences of the punctuations like \\\\, ..., !!!!, etc were also removed. Line tags were also eliminated.

- **Removing Stop Words-** We can ignore common words such as a,an,the, etc. since their appearance in a post does not provide any useful information in classifying a sentence.

- **Removing encoded data-** Our data set has latin, utf-8 encoding. The encoding were in the form of \\xc2, \\xa0, \xce, etc. We removed them from our data as they are not required for our evaluation.

- **Stemming-** It will help in reducing the vocabulary size, thereby sharpening the results. We performed Snowball stemming as it provides the most accurate results among the other algorithms available. Stemming converted the words to their root words like girls was transformed into girl.

- **Lemmatization-** It further helped us in removing similar unwanted vocabulary from our feature set. The words like accommodates, accommodation, etc were removed and only accommodate was preserved. Thus reducing the overall feature set. These results were not captured by stemming.

B. *Feature Extraction Techniques*

Different features are extracted and are compared to find the efficiency of the system. These features once extracted are used to form a numerical vector which helps to calculate tf-idf and term position values to be fed as input to feature Selection process.

- **Unigrams Extraction-** Each Sentence is broken down into individual words called tokens. The frequency of each word in the document is counted based on which its tf-idf and term position values are calculated. Extraction of Unigrams can be done as, consider a sentence I am going to market to buy some vegetables and fruits. The resulted Unigrams will be: {I, am, going, to , market, to , buy , some, vegetables, and fruits}. Here, the stop words will be removed in pre processing stage.

- **Bigram Extraction-** Words in a group of two are extracted called bigrams. These bigrams based calculated tf-idf and term position values are used to compare the results obtained using unigrams as features. The Extracted Bigrams of given example are: {I_am, am_going, going_to, to_market, market_to, to_buy, buy_some, some_vegetables, vegetables_and, and_fruits} where again the bigrams are formed after removal of stop words.

- **Unigram + Bigram Extraction-** In a liu to find better efficiency, feature extraction approach of extracting Unigram and bigram is considered. The tf-idf and term positions are calculated taking all the unigrams and bigrams extracted in Step1 and Step2.

- **Trigram-** Each sentence is broken down into combination of consecutive three tokens.

- **Concepts Extraction-** Multi Word Expressions are extracted from the text. These multiwords are called as Concepts which are extracted using concept parser. They are helpful in deconstructing natural language into meaningful pairs e.g.: adj+noun, verb+noun, noun+noun. Concepts extracted from above given example are: {go_buy, market, buy_vegetable, buy_fruit, some_fruits} .

VI. Feature Selection

We performed Chi- Square test as a feature selection technique. The test assigns high score to the features which are more distinguishing features for a class. We then select top K features for classification.

VII. Classifiers Used

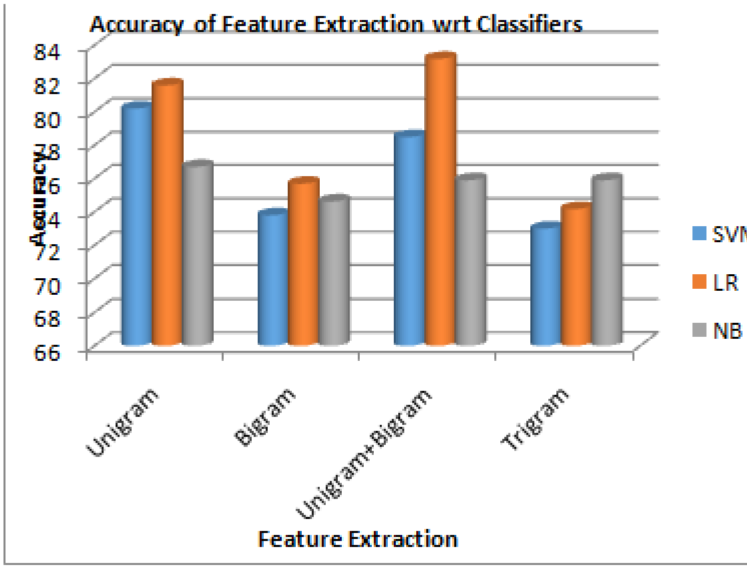
We have used Naive Bayes, SVM and Linear Regression classifiers for our analysis. The results are mentioned in the results section.

VIII. Results

The following table depicts accuracy of each classifier with respect to different feature extraction methods.

Feature Extraction	**SVM**	**LR**	**NB**
Unigram	80.2	81.59	76.7
Bigram	73.8	75.69	74.63
Unigram + Bigram	78.49	83.18	75.9
Trigram	73.0	74.17	75.9

The accuracy of Unigram is higher as compared to other feature extraction techniques in 2 of the classifiers namely SVM and Naive Bayes. However, in case of Linear Regression Unigram + Bigram seem to perform better than other feature extraction. The concept model's accuracy in Naive Bayes was 76.2



IX. Work Distribution

The first author was responsible for performing the preprocessing techniques, feature selection and making this report and the second author was responsible for performing feature extraction techniques and implementing classifiers. The literature survey was a combined effort of both the authors.

X. Future Work

In our current work, we have not given special attention to negative terms and have treated them just as any other word. The negative words changes the polarity of the sentence and should be treated differently. Another possible extension of our work can be finding out the subject to whom insult is intended. We can also manually build a dictionary for expanding some commonly used abbreviations and slangs.

References

[1] Chetan Dalal, Shivyansh Tandon, and Amitabha Mukerjee. “Insult Detection in Hindi”. In: (2014).

[2] Priya Goyal and Gaganpreet Singh Kalra. “Peer-to-Peer Insult Detection in Online Communities”. In: *IITK, unpublished* (2013).

[3] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs up?: sentiment classification using machine learning techniques”. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics. 2002, pp. 79–86.

[4] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. “Sentiment strength detection for the social web”. In: *Journal of the American Society for Information Science and Technology* 63.1 (2012), pp. 163–173.