International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Proposed Techniques to Remove Flaming Problems from Social Networking Sites and outcome of Naïve Bayes Classifier for Detection of Flames

Vishakha Mal[1], Dr. A. J. Agrawal[2]

[1]Shri Ramdeobaba College of Engineering & Management, Gittikhadan, Nagpur-13, Maharashtra, India

[2] Shri Ramdeobaba College of Engineering & Management, Gittikhadan, Nagpur-13, Maharashtra, India

Abstract: ***Natural Language Processing (NLP)*[1][2][5] *is a field of Computer Science concerned with the interactions between Computer and Human (Natural) Languages. Social Networking Sites are amongst the most effective communication tools now a days. But it also gave rise to the problem of flaming which is difficult to deal with. A flaming incident is triggered by comments and actions of users in SNS that ends up damaging reputation or causing negative impact on the target party. So in this paper, in order to prevent such damage, Semi-Supervised Learning*[1] *Approach is presented. It includes a proposed architecture that involves Naïve Bayes Classifier, Maximum Entropy Classifier and Feature Selection based on Entropy method. This paper also includes application and result of Naïve Bayes Classifier, trained so as to detect the class of the input comments as whether it belong to positive, negative or neutral class. Sentiment Analysis*[1] *Dataset containing tweets and there polarity class(positive, negative or neutral) has been used. A training set of 17282 tweets has been used to train Naïve Bayes Classifier which was able to correctly classify 70% of the test set of 4332 tweets.***

Keywords: Natural Language Processing, Flaming, Semi-Supervised Learning, Class, Naïve Bayes Classifier

1. Introduction

SNS has achieved a huge popularity in the previous decade. It has made the transmission of information across the globe very easy for everyone. But it has also brought with it a problem called *flaming*[1] which are one of the current hazards of online communication. They are caused due to negative comments made by users that can have a huge impact on the target in a negative way. So, in order to overcome this problem, sentiment analysis, which aims to classify sentiment of documents, can be used. In this paper an architecture has been proposed that describes various techniques for the purpose of finding polarity (negativity, positivity or neutrality) of the input document. A training set of 17282 tweets already annotated(as positive, negative or neutral) was preprocessed[4] by applying tokenization, stop words removal and stemming to construct a sentiment dictionary[1]. This dictionary contains the word and its positive, negative and neutral frequencies. The dictionary is context dependent as it depends on the training dataset. This is a problem of text classification[1]. Now this dictionary is used for the assigning polarity to the input document. The proposed approach describes various techniques, like Naïve Bayes Classifier[1][2], Maximum Entropy Classifier[3] and Feature Selection based on Entropy[1] method, that can use the constructed dictionary for sentiment analysis of input. This paper also describes in detail, Naïve Bayes Classifier which was trained and tested.

Sentiment Analysis can help us to know the polarity of a given document. First step is to apply pre-processing to the labeled dataset and construct sentiment polarity dictionary for the target company. Second step is to pre-process input document (comment) and then classify its sentiment using constructed dictionary, Classifiers and Feature Selection based on Entropy method.

2. Proposed Architecture

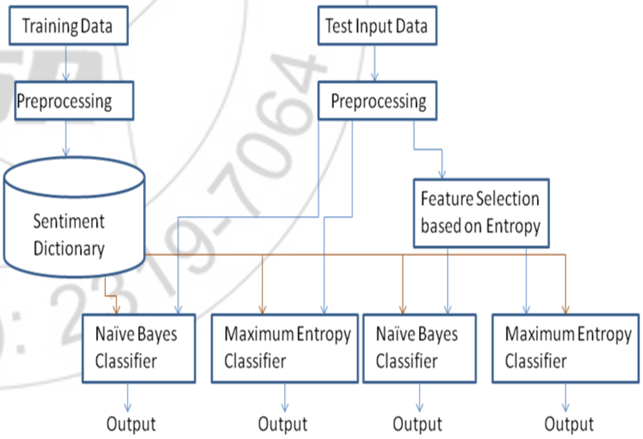

Figure 1: Proposed Architecture

The training dataset that contains 17282 tweets is first pre-processed to form Sentiment dictionary. Pre-processing involves tokenization, stop word removal stemming. The sentiment dictionary contains words and their respective positive, negative and neutral frequencies. The input test data is then inputted and pre-processing is applied on it. The pre-processed input data is a collection of features (words). The positive, negative and neutral frequencies of the features in test data is obtained using the sentiment dictionary and then using various methods the probability of the input belonging to all the classes namely positive, negative and neutral class is calculated. The input data is then considered to be classified in the class which has the maximum probability. Probability can be calculated using four methods namely

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Naïve Bayes Classifier, Maximum Entropy Classifier, Naïve Bayes with Feature Selection based on Entropy Method, Maximum Entropy with Feature Selection based on Entropy Method. The output of all the four classification methods is compared with the original classification that is already known because test data is annotated. In this way we can calculate the accuracy of various classification methods.

3. Pre-Processing of Datasets

The dataset collected from the SNS media contains a lot of opinion about the data which are expressed in different ways by individuals. The negative, positive and neutral polarity labeled dataset is highly susceptible to inconsistency and redundancy. In order to improve the quality of result, pre-processing of raw data is necessary. It helps in removing repeated words and punctuation and thus help in improving the quality of the data. Some of the pre-processing techniques are as follows:

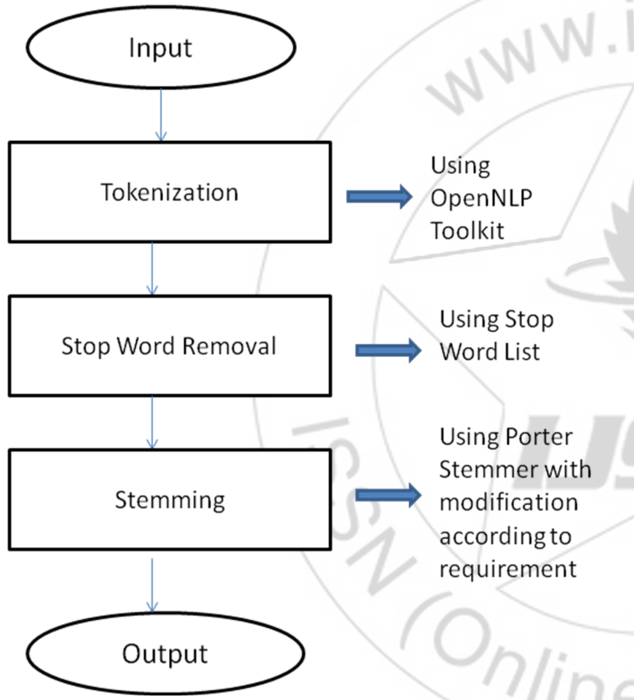

Figure 2: Pre-Processing Techniques used

3.1 Tokenization

In Lexical analysis, Tokenization is the process of breaking up a stream of text, into words, symbols or other meaningful elements called tokens. In this way, the input data on tokenization will give collection of features. For the training data, these features is used for the construction of sentiment dictionary. Features here refers to words. For test data, these features is accessed in the dictionary to obtain their frequencies in all the three classes, which is then used for the classification of the input.

3.2 Stop Word Removal

Many words will be syntactic words, i.e. they do not contribute to the semantic meaning of the sentence. Such words are required to be removed. Stop Words list is obtained and after tokenization such words that belong to the stop word list are discarded since they only lead to occupying space while not contributing for classification.

3.3 Stemming

In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. Stemming will help is reducing the size of the sentiment dictionary and also will help in getting more accurate results.

In this project OpenNLP Toolkit and Porter Stemmer has been used for pre-processing.

4. Sentiment Dictionary

Sentiment Dictionary is constructed using the annotated Training dataset that contains tweets and their annotations as they belong to positive, negative or neutral. Pre-Processing is applied to the training dataset. Sentiment Dictionary contains four columns for words and their respective frequencies in all the three classes.

5. Feature Selection based on Entropy[1]

It can be possible that dictionary may contain words (eg. I, we, us) that are not informative in case of sentiment analysis. So, in order to remove these words, we use entropy method, assuming that these words are uninformative. Let |C| be the number of classes, $n_{i,c}$ is the number of occurrences of word w_i in class c. N_i is the number of w_i in all classes. H(i) is the entropy of word w_i which is calculated as:

$$H(i) = -\frac{1}{\log |C|}\sum_c \frac{n_{i,c}}{N_i}\log\frac{n_{i,c}}{N_i} \quad (1)$$

$$0 \le H(i) \le 1$$

When the occurrence probabilities of a word w_i in respective classes are equal to each other i.e. there are no bias based on occurrence probabilities present between classes, H(i) gets highest value 1. On contrary, when a word w_i occurs in only one class i.e. its occurrence probably is causing maximum bias between classes, the value of entropy H(i) is lowest i.e. 0. So those words are used, whose entropy H(i) is equal to or lower than the threshold α, i.e.

$H(i) \le \alpha$

Thus if we set α=1, then we use words that occur only in one class and if we set α=0, all the features will be used.

6. Naïve Bayes Classifier

A Naive bayes classifier is a simple probabilistic model which is based on the Bayes rule along with a strong independence assumption. The Naïve Bayes model involves a simplifying conditional independence assumption i.e. given a class (positive, negative or neutral), the words are conditionally independent of each other. This assumption does not affect the accuracy in text classification by much but makes really fast classification algorithms applicable for the problem. In this paper a Multinomial Naïve Bayes Classifier has been used that first calculates joint probability of words and then assigns texts to the class (Positive/Negative/Neutral) they belong.

Assumptions in Multinomial Naïve Bayes Classifier:

a) In a document, the occurrence distribution of words conforms to the multinomial distribution.
b) Obtain documents from a sequence of words and occurrence of words should be independent of each other.
c) Prior Probability of a document belonging to class c is a uniform distribution.

From Training data let:

$|V|$ = Number of unique words

$V = \{w_1, w_2, \ldots, w_{|V|}\}$

$P_{i,c}$ =Occurrence Probability of a word w_i $(1 \leq i \leq |V|)$ in class c

$\Theta_c = \{P_{1,c}, P_{2,c}, \ldots, P_{|V|,c}\}$=Parameter Vector of c

$n_{i,d}$ = Number of occurrences of a word w_i in document d.

$P(d|\Theta_c)$ = Likelihood of a document d

$$P(d|\Theta_c) = \frac{(\sum_i n_{i,d})!}{\prod_i n_{i,d}!} \prod_i P_{i,c}{}^{n_{i,d}} \qquad (2)$$

Naïve Bayes Classifier outputs c* as

$$c^* = argmax_c P(d|\Theta_c) \qquad (3)$$

$$c^* = argmax_c \prod_i P_{i,c}{}^{n_{i,d}} \qquad (4)$$

Laplasian Smoothing: If the classifier encounters a word that has not been seen in the training set, the probability of both the classes would become zero and there won't be anything to compare between. This problem can be solved by Laplacian smoothing

For estimation of parameter $P_{i,c}$

$$P_{i,c} = \frac{n_{i,c} + (\alpha - 1)}{\sum_i n_{i,c} + |V|(\alpha - 1)} \qquad (5)$$

In this paper α=2 has been used which is well used as Laplacian Smoothing. So

$$c^* = argmax_c \prod_i \left(\frac{n_{i,c} + (\alpha-1)}{\sum_i n_{i,c} + |V|(\alpha-1)}\right)^{n_{i,d}} \qquad (6)$$

The main theoretical drawback of Naïve Bayes methods is that it assumes conditional independence among the linguistic features. If the main features are the tokens extracted from texts, it is evident that they cannot be considered as independent, since words co-occuring in a text are somehow linked by different types of syntactic and semantic dependencies.

7. Maximum Entropy Classifier

The idea behind Maximum Entropy classifiers is that, given the constraints, most uniform models are most preferable Maximum Entropy models are feature based models. These features are used to find a distribution over the different classes using logistic regression.

Goal: Given a set of features, a set of their corresponding functions that measures the contribution of each feature to the model known as weighting Factor, as λ_i (i = 1, 2,… , k) and a set of constraints our aim is to find the probability distribution that maximizes the relative Entropy. By maximizing entropy, it is ensured that no biases are introduced into the system.

$$P(c|d,\lambda) = \frac{exp \sum_i \lambda_i f_i(w_i,c)}{\sum_{c'} exp \sum_i \lambda_i f_i(w_i,c')} \qquad (7)$$

$$\text{Where } f_i(w_i,c) = \begin{cases} 1, & \text{if } w_i \in c \\ 0, & \text{otherwise} \end{cases} \qquad (8)$$

The Major Advantages of using Maximum Entropy Classifier is as follows:

- Accuracy
- Consistency – This method shows consistency in results and if priors are used results also improve over a period of time.
- Performance / Efficiency - Can handle huge amounts of data
- Flexibility - The classifier is flexible of having many different typed of data in a unified platform that are classified accordingly.

8. Result

4332 tweets were classified using Naïve Bayes Classifier, out of which 3016 tweets were correctly classified. Following table gives the details of the classification:

Table 1: Classification of Tweets using naïve Bayes Classifier

	Classification using Naïve Bayes Classifier			
Original Classification	Tweets	Positive	Negative	Neutral
	Positive	1588	344	1
	Negative	637	1297	0
	Neutral	174	150	131

Accuracy is calculated using the following formula:

$$Accuracy = \frac{Total\ Correctly\ Classified\ Tweets}{Total\ Tweets} * 100 \qquad (9)$$

So Accuracy of Naïve Bayes classifier is 70%.

9. Conclusion

Naïve Bayes Classifier classified positive tweets with a good accuracy, correct classification of negative tweets is moderate while neutral tweets are poorly classified. The classification results can be improved by applying feature selection based entropy method and Maximum Entropy Classifier. Words that do not have much bias to a particular class are also used for classification. This can be the reason for many incorrect classifications. Feature Selection based on entropy method helps in identifying such words. Also, since Maximum Entropy Classifier ensures minimum bias into the system (this implies minimum assumptions) can also contribute in better accuracy.

References

[1] Shun Yoshida, Jun Kitazono, Seiichi Ozawa, Takahiro Sugawara, Tatsuya Haga, and Shogo Nakamura, “Sentiment Analysis for Various SNS Media using Naïve Bayes Classifier and its Application to Flaming Detection”, 978-1-4799-4540-5/14/ © IEEE, 2014

[2] Eric N. Forsyth, and Craig H. Martell, “Lexical and Discourse Analysis of Online Chat Dialog”, International Conference on Semantic Computing IEEE, 2007.

[3] Yang Li, Bin-Xing Fang and Li Guo, ”A Novel Online Spam Filter Based on URLs and Maximum Entropy Model”, IEEE, 2006.

[4] Madhuri A. Tayal, Dr. M M Raghuwansh and Dr. Latesh Malik, “Knowledge Representation: Predicate Logic Implementation using Sentence-Type for Natural Languages”, 2013 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2013] IEEE, 20113.

[5] Saravanan M, Divya Sundar, and Kumaresh V S, “Probing of Geospatial Stream Data to Report Disorientation”, 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2013.

Author Profile

Vishakha Mal received the B.E. degree in Computer Science & Engineering from Bhilai Institute of Technology, Durg(CG) in 2014. Now she is doing M.Tech in Computer Science & Engineering from Shri Ramdeobaba College Of Engineering & Management.

Avinash J. Agrawal received Bachelor of Engineering Degree in Computer Technology from Nagpur University, India and Master of Technology degree in Computer Technology from National Institute of Technology, Raipur, India in 1998 and 2005 respectively. He has Ph.D. in Computer Science and Engineering from Visvesvaraya National Institute of Technology, Nagpur in 2013. His research area is Natural Language Processing and Artificial Intelligence. He is having 18 years of teaching experience. Presently he is Associate Professor in Shri Ramdeobaba College of Engineering and Management, Nagpur. He is the author of more than 50 research papers in International Journal and Conferences.