Satire Detection from Web Documents using machine Learning Methods

Tanvir Ahmad
Dept. of Computer Engineering
Jamia Millia Islamia University
Delhi, India
tahmad2@jmi.ac.in

Halima Akhtar[1], Akshay Chopra[2], Mohd Waris Akhtar[3]
Dept. of Computer Engineering
Jamia Millia Islamia University
Delhi, India
akhtar.halima@yahoo.com[1],aks.cul@gmail.com[2], warisakhtar@gmail.com[3]

***Abstract*—Satire exposes humanity's vices and foibles through the use of irony, wit, and sometimes sarcasm too. It is also frequently used in online communities. Recognition of satire can help in many NLP applications like dialogue system and review summarization. In this paper we filter online news articles as satirical or true news documents using SVM (Support Vector Machine) classification method combined with machine learning techniques. With ample training documents SVM tends to give good classification results. For obtaining promising results with SVM an understanding of its working and ways to influence its accuracy is required. We also use various feature extraction strategies and conclude that TF-IDF-BNS feature extraction gives maximum accuracy for detection of satire in web content.**

***Keywords* – Satire detection, bi-normal separation, classification, SVM, feature extraction**

I. INTRODUCTION

Satire and irony is useful form of speech act in any communicator's daily conversation. It is a kind of writing that mixes a critical attitude along with humor and wit in an order to improve mankind and human institutions.

Satirical news articles often try to mimic true news articles, incorporating sarcasm and irony to provide humorous insight. We define satirical article as one which tend to deliberately expose organizations, real world individuals and events to ridicule. The difficulty in satire detection results in misunderstanding in communication and also pose problem for NLP systems like dialogue system, online review systems and brand monitoring systems because of failure in recognizing sarcastic comments[1]. In this paper we utilize some machine learning techniques along with various feature extraction strategies to classify satirical new articles. Bag of words representation is used for news articles along with feature weighing to obtain good classification accuracy.

II. RELATED WORK

Satire classification is termed as a prominent undertaking to computational linguistics. Text classification is the task of assigning any kind of topic category to any piece of text and that could be subject categories in online data base Keeping into consideration the most widely-researched text classification task , sentiment classification (Pang and Lee, 2008) [2]. Various methods have been proposed earlier and quite a lot of them is applied directly to news classification provided that the categories are predefined and also that there exists a good documentation set for training for each predefined category[3,4].

We have used the following targeted lexical features that are the main unique characteristics of satire news documents which are Headline features, Slang, Semantic, Profanity[5]. A growing number of statistical means for feature selection or rather ,terming it as feature scaling using feature scoring metrics have been used for text categorization. That includes the two most novel text representations, TF·IDF and binary features which can be further segregated to feature scaling methods which are TF IDF, BNS, and no scaling (binary weights).We clearly demarcate the higher Accuracy , Precision , F-Score , Recall for BNS as compared to the rest as in reference to the HP Lab work[6].By contrast, for our computational work ,we use the LIBSVM perform the inductive learning for the training dataset as perhaps in accordance with the recent work Joachims (1999) who has explored the use of Support Vector Machines (SVMs) for text classification with promising and more precise results as compared to the other classification techniques[7].

III. PROPOSED ARCHITECTURE

Figure 2. describes the proposed framework needed for the classification of satirical new articles. The various modules in the framework are – *Document Pre-processing, Feature extraction, machine learning and SVM classifier.*

978-1-4673-6751-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ISCMI.2014.34

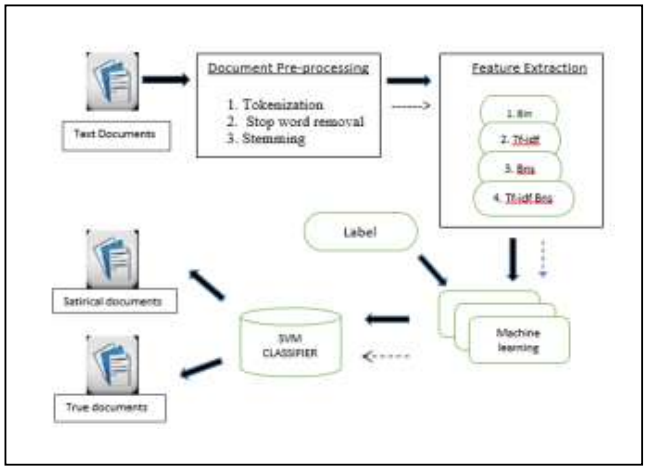

Figure 1. Proposed Architecture of Satire Detection

In figure 1. data flow of the procedure is denoted by solid arrows whereas the control flow of the process is depicted by dotted arrows.

A. *Document Pre-processing*

This involves filtering of the documents such that the terms without content can be removed from the documents so that efficient processing of the documents can take place. It consists of –

1. Tokenization
2. Stop Word Removal
3. Stemming

B. *Feature Extraction*

The input set of data is transformed into a set of features. Each word is regarded as one unique feature. The various weighing schemes applied to these features are-

1. Bin – Binary feature weight gives same weight to all features no matter how many times they appear in the document.
2. TF-IDF – It relates occurrences of word with the number of documents in the data set.
3. BNS – Bi-normal separation scheme allocates weights according to true positive and false positive rates. It assigns highest weights for features strongly related to positive or negative class.
4. TF-IDF-BNS – This scheme combines two weighing factors i.e. tf-idf and bns to allocate weights to features.

C. *Machine Learning*

We train a classifier using classification tool known as LIBSVM (which is SVM classifier).

D. *SVM Classifier*

It is a set of supervised learning method to analyze data set and predict the two possible classes of the input

IV. PROPOSED ALGORITHM

For the algorithm we consider a wide variety of text feature

Representations along with the targeted lexical features that a satire document may possess. For each potential feature scoring metric, we use it as a scale factor on TF feature counts and separately also as a scale factor on binary features.

A. *Benchmark Dataset*

Our data corpus consists of a total of 2624 newswire documents and 171 satire news articles, that are split into fixed training and test sets as detailed in Table 1. The newswire documents were randomly sampled from the English Gigaword Corpus and the satire data set was selected form The Orion. The 1595 test data articles from the corpus is chosen as the training data set for our work and the 1200 training data articles from the corpus is chosen as the test data respectively.

TABLE I. DATA CORPUS

	TRAINING	TEST	TOTAL
TRUE	1495	1129	2624
SATIRE	100	71	171

Satirical news articles are inclined to mimic the true newswire articles and they incorporate irony so as to provide humorous insight.

An example excerpt for satirical data from *Orion* is:

Dad way scarier when controlling temper

Contrasting this with a true newswire snippet from *Reuters* :

Gunman storm the parliament after attack.

B. *Data Preprocessing*

*P*reprocessing involves Tokenization,Stop words removal and Stemming which is done using NLTK tool kit.

1) *Tokenization*

It is the process of breaking a sentence or stream of text up into words, symbols,phrases, or other meaningful elements

Phrase: ―I will attend a tutorial .

After Tokenization: ,"I ","will" ,"attend","a", "tutorial"

2) *Stop words*

Words that are filtered out prior to, or after , text processing.

Example: Remove words such as "a", "the","I","he"

3) *Stemming*

Process to reduce inflected or derived words to their stem or root form. It is done using Potter Stemmer in nltk.

Example: Attending → attend

C. *Feature Extraction*

Feature extraction deals with the simplification of the resources that are required to describe a comparatively

large data set precisely.It is the task of transforming the input dataset into a set of defined features .

We adopted and then implemented the following weighting schemes, namely:

1. Term Frequency - Inverse Document Frequency (TF-IDF) as shown in equation (1)

$$tfidf(w) = tf.\log(\frac{N}{df(w)}) \quad (1)$$

tf(w)–term frequency (word occurrences in a document)

df(w)–document frequency (number of documents containing the word)

N–number of all documents

tfidf(w)–relative importance of the word in the document

2. Term Frequency Bi-normal separation feature scaling (TF-BNS) as shown in equation (2)

$$tf - bns(w) = tf.\left(F^{-1}(tpr) - F^{-1}(fpr)\right) \quad (2)$$

F^{-1} – inverse normal cumulative distribution function

tpr – true positive rate (P(feature|positive class))

fpr – is the false positive rate (P(feature|negative class))

TF - is term frequency .

3. Binary feature weight (BIN)

Under this, all features are assigned the same weight, regardless of their frequency of occurrence in each article.

4. TF -BNS-IDF : as shown in equation(3) and (4)

$$tf - bns - idf(w) = tf - bns * \frac{idf(w)}{W} \quad (3)$$

$$tf - bns - idf(w) = \frac{\left(tf.\left(F^{-1}(tpr) - F^{-1}(fpr)\right) * \log\left(\frac{N}{df(w)}\right)\right)}{w} \quad (4)$$

TF-BNS–Term Frequency Bi-normal separation scaling

IDF– log*(N/ df(w)*)

W– word frequency (number of word occurrences in all documents)

D. *Inductive Learning of Classifier*

We train a classifier using a classification tool, e.g. LIBSVM which is a SVM (Support Vector Machine) classifier.In the testing phase, the effectiveness and accuracy of the model is tested on unseen instances [8].SVM is a *non-probabilistic binary linear classifier* as for each given input the standard SVM input data and predicts which of the two possible classes forms the input.

We have used LIBSVM with the options as :

`-s svm_type` – (2 -- one-class SVM)

-t kernel_type – (0 -- linear: u'*v)

E. *Classification*

In machine learning , classification basically deals with the issue of identification so as to determine to which set of categories (sub-populations) a new test data belongs, on the basis of a training set of data containing observations (or instances) whose category membership is defined .In our case we used SVM Classifier (LIBSVM) for classification of Satire Article.

V. EMPIRICAL RESULTS

For the evaluation of performance we use the two standard performance measures: accuracy for its historical standard in machine learning research and F-measure for its improved sensitivity in the common information retrieval situation where positives are rare.F-measure is the harmonic average of precision & recall.

Accuracy: It is the proportion of true results (both true positives and true negatives) in the dataset as in equation (5)

$$Accuracy = \frac{No.\ of\ True\ Positives + No.\ of\ true\ negatives}{TP+FP+FN+TN} \quad (5)$$

Precision: It is the proportion of the true positives against all the positive results (both true positives and false positives) as shown in equation (6)

$$Precision = \frac{Number\ of\ True\ Positives}{TP + FP} \quad (6)$$

Recall: It is the proportion of the true positives against all the true results (both true positives and false negatives) as shown in equation (7)

$$Recall = \frac{Number\ of\ True\ Positives}{TP + FN} \quad (7)$$

F-Score: It considers both the precision and the recall of the test to compute the score as in equation (8)

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (8)$$

Comparing the results of the feature extraction techniques,

We find that the proposed algo of TF-BNS –IDF showcase a better result over all the others.

TABLE II. ACCURACY FOR ALL

Weighing Parameter	Accuracy
TF-IDF	83.4160 %
BIN	84.0833 %
BNS	89.9167 %
TF-BNS-IDF	92.7500 %

TABLE III. COMPARATIVE STUDY

Weighing Parameter	Precision	Recall	F-score
TF-IDF	0.722	0.732	0.727
BIN	0.746	0.788	0.767
BNS	0.802	0.859	0.829
TF-BNS-IDF	0.826	0.870	0.840

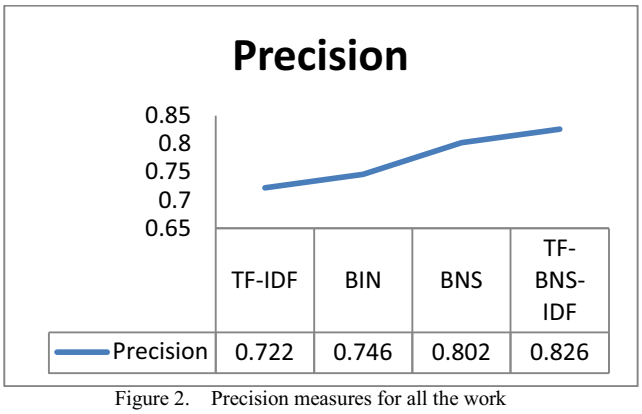

Figure 2. Precision measures for all the work

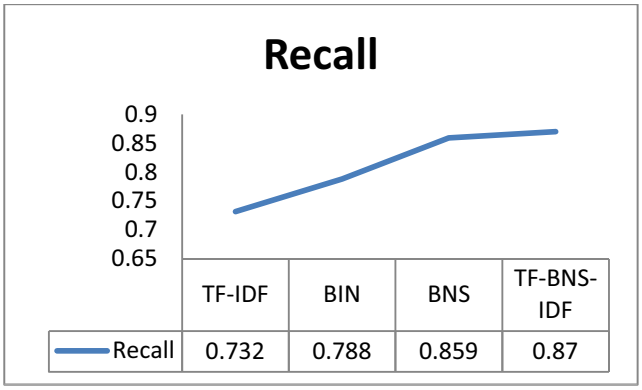

Figure 3. Recall measures for all the work

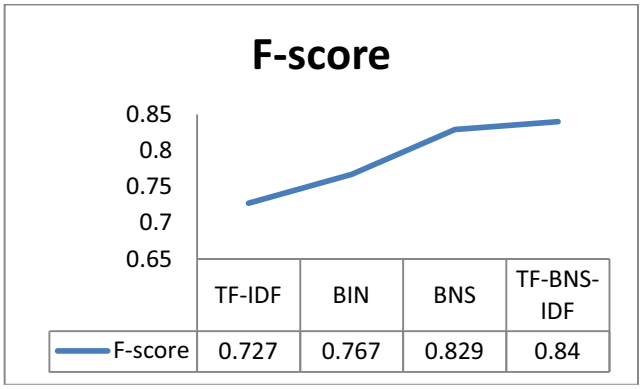

Figure 4. F-Score measures for all.

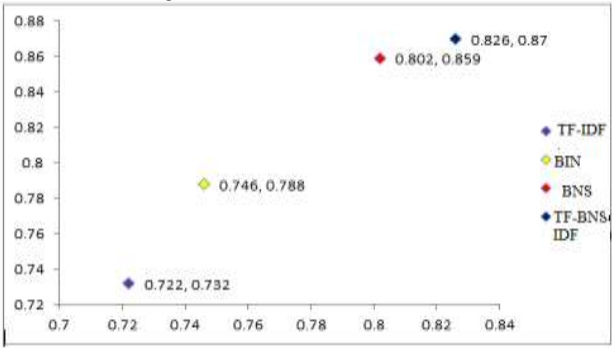

Figure 5. Shows the precision vs recall measures for all.

VI. CONCLUSION

This research project has performed the task of computational linguistics: determining whether a newswire article is ―true‖ or satirical. We found that the combination of SVMs with TF-BNS-IDF feature scaling achieves high precision and high F-Score than the combination with IDF , BNS , BINARY WEIGHTS. Our classification using various weighing technique provided varied result.

Controlling the parameters of the SVM Classifier also helped in getting better result and we arrived at the best result by exhaustive experiments and trials with the SVM classifier.

ACKNOWLEDGMENT

This work benefited from discusions with our mentors who were guiding us.We appreciate the discussion work and review made available by HP Labs Utility Datacenter.

REFERENCES

[1] Olga Nikitina , “Automatic detection of satire and sarcasm - Computational Approaches to Creative Language”, in SS 2010, 22 June .

[2] Pang and Lee ,”Opinion mining and sentiment analysis”, in Foundations and trends in Information Retrieval, Vol. 2, 2008

[3] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. “Inductive learning algorithms and representations for text categorization”, in Proceedings of the 7th international conference on Information and knowledge management, 1998, pages. 148–155

[4] Y. Yang and X. Liu, “A re-examination of text categorization methods”, in 22nd Annual International SIGIR, 1999, pages 42

[5] George Forman, “BNS Feature Scaling: An Improved Representation over TF·IDF for SVM Text Classification”, in ACM 17th Conference on Information and Knowledge Management, October 26-30, 2008

[6] Joachims , “Learning to classify using support vector machines “, 1999,page 188.

[7] C. Burfoot and T. Baldwin, “Automatic Satire Detection: Are You Having a Laugh?”, in Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 161–164.

[8] Chee-Hong Chan, Aixin Sun, EE-Peng Lim, “Automated Online News Classification with Personalization”, in Proceedings 4th International Conference of Asian Digital Library, 2001, pages 320-329.