The Behavioral Modeling Approach for Sarcasm Detection on E-Commerce & OSN

Geeta Bhagwan Mehetre, M. B. Kalkumbe

Abstract— ***Sarcasm transforms the polarity of an apparently positive or negative affirmation into its opposite. We propose a method to construct a sarcastic Twitter message corpus in which the determination of the sarcasm of each message is made by the system. We use this reliable corpus to compare sarcastic statements in Twitter with statements that express positive or negative attitudes without sarcasm. We study the impact of lexical and pragmatic factors on the effectiveness of automatic learning to identify sarcastic utterances and we compare the performance of automatic learning techniques and human judges in this task. Perhaps it is not surprising that neither human judges nor mechanical learning techniques work very well.***

Keywords— ***Hashtags, Linguistics, Opinion Mining, Sarcasm Detection, Tweets, Open NLP.***

I. INTRODUCTION

Microblogging platforms such as Twitter, which allow users to communicate their feelings, opinions and ideas in short messages and assign labels to their own messages, have been exploited recently in the analysis of feelings and opinions [3]. Sarcasm (also known as verbal irony) is a sophisticated form of the act of speech in which speakers transmit their message implicitly. A characteristic inherent in the sarcastic act of speech is that it is sometimes difficult to recognize. The difficulty of recognizing sarcasm causes misunderstandings in daily communication and poses problems to many NLP systems such as online review synthesis systems, dialogue systems or brand surveillance systems because of the " Failure of peak feel systems to detect sarcastic comments. In this article, we experiment with a semi-supervised framework for the automatic identification of sarcastic sentences.

When it comes to feelings or emotions, no one is interested in the text, but focuses on its positive or negative expressions. People can easily express their opinions about social media services such as reviews, blogs, social networking sites because they provide a huge amount of valuable information. Today, an automated identification of feelings is made, which is beneficial for many NLP systems, such as examination synthesis systems (MOS), dialogue systems, and public media analysis systems. Existing feelings extraction systems rely mainly on the identification of polarity (eg, positive and negative evaluations), but there are many useful and comparatively unexplored feelings such as sarcasm, irony or humor . In this paper, the sarcasm of feeling was explored and its detection was done on Twitter, as a platform.

In Twitter, messages can be annotated with hashtags such as #bicycling, #happy, and #sarcasm. We use these hashtags to build a corpus marked by sarcastic, positive and negative tweets. In this article, we present an empirical study on the use of lexical and pragmatic factors to distinguish sarcasm from positive and negative feelings expressed in Twitter messages. The contributions of this article include i) the creation of a corpus that includes only sarcastic statements that have been explicitly identified as such by the composer of the message; Ii) a report on the difficulty of distinguishing sarcastic tweets from tweets that are directly positive or negative. Our results suggest that lexical characteristics alone are not sufficient to identify sarcasm and that pragmatic and contextual characteristics deserve further study.

II. RELATED WORK

Sarcasm and irony are well studied phenomena in linguistics, psychology and cognitive sciences [6]. But in the textual literature, the automatic detection of sarcasm is considered a difficult problem to solve [10] and very few studies have been done in this field. In the context where spoken dialogues are considered, the automatic detection of sarcasm is based simply on speech-related signals such as laughter and prosody [15]. The work most closely related to what we propose in this article belongs to [3], where in the author caaried the main task to identify sarcastic and non-sarcastic comments or messages in Twitter and in product reviews Amazon .

The sarcasm in written and spoken interaction may work differently [5]. In oral interaction, sarcasm is often marked by a special intonation [6] or an incongruous facial expression.Since sarcasm is more difficult to understand than a literal statement [5], it is likely that the recipients do not take sarcasm and literally interpret the statements. According to Gibbs and Izett [6], sarcasm divides its recipients into two groups; A group of people who understand sarcasm (the so-called group of wolves) and a group of people who do not understand sarcasm (the so-called group of sheep). To ensure that recipients detect sarcasm in the declaration, senders use language markers in their statements. According to Gibbs [6], these markers are clues that a writer can give to "alert a reader to the fact that a sentence is ironic" [6]. On Twitter, the hashtag '#sarcasm' is a popular marker.

Manuscript published on 30 June 2017.
***Correspondence Author(s)**
Geeta Bhagwan Mehetre, Computer Science, MSSCET, Jalna, Maharashtra, India. (e-mail: 29geeta692@gmail.com).
Prof. M. B. Kalkumbe, Computer Science, MSSCET, Jalna, Maharashtra, India.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The automatic classification of communicative constructions in short texts has become a subject widely studied in recent years. Large amounts of comments, status post and personal quotes are updated on social media web portals such as Twitter.

The automatic labeling process for polarity associated with text as positive or negative may reveal, aggregate or follow over time how the general public thinks of certain things. See Strapparava [14] for an overview of recent research in the analysis of opinion and opinion mining. A major obstacle to the automatic determination of the polarity of a text (short) are constructions in which the literal meaning of the text is not the intended meaning of the sender, as many polarity detection systems rely primarily on Positive and negative words as markers. The task of identifying these constructs can improve the classification of polarities and provide new perspectives on the relatively new genre of short messages and microtexts on social media. Previous work describes the classification of irony [9], sarcasm (Tsuret al., 2010), satire (Burfoot and Baldwin, 2009) and humor (Reyes et al.). The work of Reyes et al. (2012b) and Tsur et al (2010). Reyes et al. (2012b) collect a corpus of irony based on tweets that consist of the #irony hashtag to form classifiers on different types of features (signatures, unexpected, style and emotional scenarios) and try to distinguish # irony-tweets tweets containing hashtags #education, #humor or #politics, achieving scores of F1 of about 70. Tsur et al. (2010) focus on product reviews on the World Wide Web, and try to identify sarcastic sentences from them in a semi-supervised way. Training data can be collected using manual annotation for sarcastic sentences and retrieving additionally training data can be built based on phrases annotated as queries.

The sarcasm is annotated on a scale of 1 to 5. As characteristics, Tsur et al. Look at the patterns of these sentences, composed of high frequency words and content words. Their system obtained an F1 score of 79 on a product review test, after extracting and annotating a sample of 90 phrases classified as sarcastic and 90 phrases classified as non-sarcastic. In the two works described above, a system is tested in a controlled environment: Reyes et al. (2012b) compare irony with a small set of other subjects, while Tsur et al. (2010) derived from the unlabeled test a sample of product reviews with 50% of the phrases classified as sarcastic. On the other hand, we apply a sarcasm detector leads to a set of real world tests representing a realistic sample of tweets posted on an account.

III. PROPOSED METHOD

Given a set of tweets, we try to rank each according to whether it is sarcastic or not. Therefore, from every tweet, we extract a set of features, we refer to a set of learning and use automatic learning algorithms to perform the classification. The features are extracted in a way that makes use of different tweet components, and covers different types of sarcasm. All the tweets on which we perform our experiments are checked and annotated manually.

Since the existing Twitter dataset was removed from the server that contained 58,609 tweets with the tag "#sarcasm", we will create another that will be cleaned by removing noisy and irrelevant ones, as well as those where using Hashtag Fall in one of the first two uses of the three described above.

Regarding non-sarcastic tweets, we have collected tweets dealing with different topics and we are assured that they have some emotional content. The dataset will contain sarcastic and non-sarcastic tweets. The sarcastic tweets will be collected by querying the Twitter API with the #Sarcasm tag. To reduce noise, we filter non-English tweets, very short tweets (that is, those with less than 3 words) and those that contain URLs. In most cases, URLs refer to photo links. We believe that part of the sarcasm is included in the photo, so we reject them. This dataset is used during our experimentation process to optimize the parameters defined for our functionalities.

In the rest of this work, we will refer to this set as "optimization set". The set also contains sarcastic tweets, which are checked manually and classified as sarcastic and non-sarcastic. This set will serve as a test and will be used to evaluate the performance of our proposed approach. Therefore, in the rest of this work, it will be called "test set".

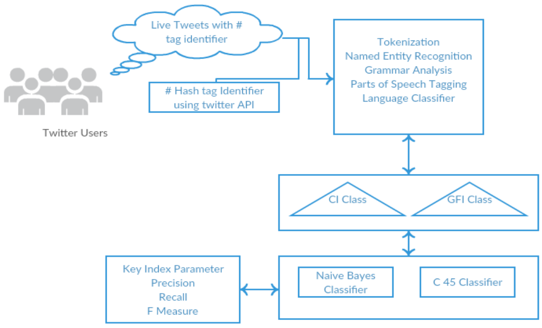

Figure 1: Architectural Diagram of Proposed System

- **Framework Sentence Analysis and Grammar Identification**

A. Tokenization

Tokenization [12] is the process of breaking a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. The tokens can be individual words, sentences or even entire sentences. In the process of tokenization, certain characters such as punctuation are rejected. The tokens become the entry for another process such as parsing and extracting text. The tokenization relies mainly on simple heuristics to separate the tokens by following a few steps:

• Tokens or words are separated by spaces, punctuations, or line breaks.

• White space or punctuation marks may or may not be included if required.

• All characters in contiguous strings are part of the token. Tokens can consist of all alpha, alphanumeric, or numeric characters.

B. POS Tagging

Corpus [11] are composed of names or names, usually emerging as the subject or object of a tweet. In the case of dependency grammar, the subject opinion function has a syntactic relation of the subject-type verb (SBV) with the sentence predicate.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The object opinion characteristic has a dependency relation of the verbal object (VOB) on the predicate. In addition, it also has an object-object dependency relation (POB) on the prepositional word in the sentence.

TABLE 1: POS Tags for Words Considered as Highly Emotional

Part of Speech	Part of Speech Tag
Adjectives	"JJ","JJR","JJS"
Adverbs	"RB","RBR","RBS"
Verbs	"VB", "VBD", "VBG","VBN","VBP","VBZ"

Adjectives, verbs and adverbs have higher emotional content than names. Therefore the positive and negative words that have the associated PoS tag shown in TABLE 1 are counted again and used to create two other characteristics that we call PW and NW and represent the number of highly emotive positive and strong negative emotional words .

We add three additional features by counting the number of positive, negative and sarcastic emoticons. Sarcastic emoticons are emoticons sometimes used with sarcastic or ironic utterances (eg, "P"). These emoticons are sometimes used when the person tries to be funny or show that he is just making a joke (ie when sarcasm is used as a mind). Hashtags also have emotional content. In some cases, they are used to disambiguate the actual intention of the Twitter user transmitted in his message. For example, the hashtag used in the following tweet: "Thank you very much for being there for me #ihateyou" says that the user does not really want to thank the recipient, rather than blaming him for not being there for him. Therefore, we also count the number of positive and negative hashtags.

C. Punctuation-Related Features

Adjectives, verbs and adverbs have higher emotional content than names. Therefore the positive and negative words that have the associated PoS tag shown in TABLE 1 are counted again and used to create two other characteristics that we call PW and NW and represent the number of highly emotive positive and strong negative emotional words .

We add three additional features by considering the amount of completely positive, completely negative and completely sarcastic emoticons. Sarcastic emoticons are emoticons sometimes used with sarcastic or ironic utterances (eg, "P"). These emoticons are sometimes used when the person tries to be funny or show that he is just making a joke (ie when sarcasm is used as a mind). Hashtags also have emotional content. In some cases, they are used to disambiguate the actual intention of the Twitter user transmitted in his message. For example, the hashtag used in the following tweet that conveys, "Thank you very much for being there when i needed you a lot PS: #ihateyou" says that the user does not really want to thank the recipient, rather than blaming him for not being there for him. Therefore, we also count the number of positive and negative hashtags..

- Number of exclamation marks
- Number of question marks
- Number of dots
- Number of all-capital words
- Number of quotes

The excessive use of exclamation marks or question marks, or the repetition of a vowel, especially in an emotional word, may reflect a certain tone that the user intends to show, Tone n ' Is not always sarcastic [7]. We believe that these characteristics can be strongly correlated with the number of words in the tweet. Some very short tweets that end with many exclamation points might surprise rather than sarcasm.

D. Pattern-Related Features

The models selected in the previous subsection and called "common sarcastic expression" [9] are very common, even in spoken language. However, their number is small, they are not unique and most tweets in our workout and test sets do not contain them. However, we dig deeper and extract another set of features.

We offer more efficient and reliable models. We divide the words into two classes [12]: a first called "CI" containing words whose content is important and a second called "GFI" containing the words whose grammatical function is more important. If a word belongs to the first category, it is lemmatised; Otherwise, it is replaced by a certain expression. The expressions [13] used to replace these words are presented in TABLE 2. The classification in classes is done according to the part of voice tag of the word in the tweet.

TABLE 2: POS Tags and Expression

POS Tag	Expression
CD	CARDINAL
FW	FOREIGNWORD
UH	INTERJECTION
LS	LISTMARKER
NN, NNS, NNP, NNPS	NOUN
PRP	INTERJECTION
MD	MODAL
PB, RBR, RBS	ADVERBS

E. C45 Algorithm

J48 or C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

F. Algorithm

C4.5 creates few decision trees from the set of training data and it is likely to create in the same way as ID3, using the eventual concept of Entropy of information. The training data is a set of samples already classified. Each sample

Consists of a p-dimensional vector, where they represent attributes or characteristics of the sample, as well as the class in which it falls.

At each node of the tree, C4.5 chooses the data attribute that most effectively divides its set of samples into enriched subsets in one class or the other.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The splitting criterion is the gain of standardized information (difference of entropy). The attribute with the highest standardized information gain is chosen to make the decision. The algorithm C4.5 then returns to the smaller sublists.

This algorithm has some basic cases.

• All samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree by saying to choose this class.

• None of the features provide information gain. In this case, C4.5 creates a higher decision node in the tree using the expected value of the class.

• Previously invisible class instance encountered. Again, C4.5 creates a decision node higher in the tree using the expected value.

The C5.0 algorithm is an extension of the C4.5 algorithm which is also the extension of ID3. This is the classification algorithm that applies in the large dataset. It's better than C4.5 on speed, memory and efficiency. Model C5.0 works by dividing the sample according to the field that provides the maximum information gain. Model C5.0 can divide the samples on the basis of the larger information gain field. The sample subset that is obtained from the first division will be divided thereafter. The process will continue until the subset of samples can not be split and is usually in a different field. Finally, examine the lowest level fraction, the subsets of samples that have no remarkable contribution to the model will be rejected. C5.0 easily handles the multi-value attribute and the missing attribute of the dataset [3].

G. Rule sets:

An important feature of is to generate classifiers called rule sets that consist of unordered collections of (relatively) simple if-then rules. The Rule sets option causes classifiers to be expressed as rule sets rather than decision trees.

Each rule consists of:

• A rule number - this is quite arbitrary and serves only to identify the rule.

• Statistics (n, lift x) or (n / m, lift x) that summarize the performance of the rule. Similarly to a sheet, n is the number of training cases covered by the rule and m, if it appears, shows how many of them do not belong to the class predicted by the rule. The precision of the rule is estimated by the Laplace ratio (n-m + 1) / (n + 2). The elevator x is the result of dividing the estimated accuracy of the rule by the relative frequency of the predicted class in the drive assembly.

• One or more conditions that must be met for the rule to apply.

• A class predicted by the rule.

• A value between 0 and 1 that indicates the confidence with which this prediction is made.

When a rule like this is used to classify a case, it may happen that several of the rules are applicable (ie, all of their conditions are satisfied). If the applicable rules provide for different classes, there is an implicit conflict that could be solved in several ways: for example, one might think that the rule is the greatest confidence, or one might try to aggregate the rule predictions to arrive at To a verdict. Rule sets are usually easier to understand than trees because each rule describes a specific context associated with a class. In addition, a rule set generated from a tree usually has fewer rules than the tree has leaves, another more for comprehensibility. Another advantage of rule-set classifiers is that they are often more accurate predictors than decision trees, since the rule set has an error rate of 0.5% on test cases. For very large datasets, however, generating rules with the rule set option may require much more computer time.

H. Naive Bayes Classifier

Naive Bayes is a probabilistic classifier, which means that for a document d, all classes c ∈ C the classifier returns the class c which has the posterior maximum probability given the document.

The Naive Bayes classifier is a simple probabilistic classifier that is based on the Bayes theorem with strong and naive assumptions of independence. This is one of the most basic text classification techniques with various applications in the detection of e-mail, personal message sorting, categorization of documents, detection of sexually explicit content, detection of languages and The detection of feelings. Despite the naive design and simplified assumptions that this technique uses, Naive Bayes works well in many complex real world problems.

Even though it is often surpassed by other techniques such as boosted trees, random forests, Max Entropy, Support Vector Machines etc., Naive Bayes classifier is very effective as it is less costly in computing (both CPU and memory) And it requires a small amount of training data. In addition, the training time with Naive Bayes is much lower as opposed to the alternative methods.

The Naive Bayes classifier is superior in terms of CPU and memory consumption as shown by Huang, J. (2003), and in many cases its performance is very similar to more complicated and slower techniques.

A naive bayes classifier[15] is a simple probabilistic model based on the Bayes rule with a strong hypothesis of independence. The Naïve Bayes model implies a simplified conditional independence hypothesis. This is given a class (positive or negative), the words are conditionally independent of each other. This assumption does not significantly affect the accuracy of the text classification, but makes the classification algorithms very fast applicable to the problem. In our case, the probability of maximum likelihood of a word belonging to a given class is given by the expression:

$$P(x|c) = \frac{countofxintweetsofclassc}{totalnumberofwordsinclassc}$$

Here, the xi s are the individual words of the post tweet. The classifier delivers the class with the maximum a posteriori probability. We also remove duplicate words from tweets, they do not add any additional information; This type of Naive Bayes algorithm is called Bernoulli Naïve Bayes. The inclusion of the presence of a word instead of the count has been found to improve performance marginally, when there are a large number of training examples.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Key Index Parameters for Result Classification

In information retrieval with binary classification, precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also called sensitivity) is the fraction of the relevant instances that are retrieved. Precision and recall are therefore based on understanding and measuring relevance.

In simple terms, high accuracy means that an algorithm returns significantly more relevant than irrelevant results, while a high recall means that an algorithm has yielded the most relevant results.

The most important category measurements for binary categories are:

Precision	$P = TP/(TP + FP)$
Recall	$R = TP/(TP + FN)$
F Measure	$tp + tn/tp + tn + fp + fn$

IV. CONCLUSION

The detection of sarcasms is a really fascinating subject to deal with. It evaluates different types of characteristics for extracting feelings, including feelings, words, patterns and n-grams, confirming that each type of characteristic contributes to the classification of frame feelings.

In this work, we propose a new hybrid method to detect sarcasm in Twitter. The proposed method uses the different components of the tweet. Our approach uses Partof-Speech-tags to extract models characterizing the sarcasm level of tweets.

In the future these methods can be applied to the automated grouping of feelings and sensation dependency rules and can be developed to detect other non-literary forms of feelings such as humor.

ACKNOWLEDGMENT

I would like to thank Principal. Dr. S.K. Biradar, MSSCET and HOD. Prof. G.P. Chakote, Computer Science, MSSCET, Jalna, Maharashtra for the support & guidance throughout the project.

REFERENCES

1. Carvalho, P., Sarmento, S., Silva, M. J., and de Oliveira, E. 2009. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In Proceeding of the 1st international CIKM workshop on Topicsentiment analysis for mass opinion (TSA '09). ACM, New York, NY, USA, 53-56
2. Clark, H. and Gerrig, R. 1984. On the pretence theory of irony. Journal of Experimental Psychology: General, 113:121–126. D.C.
3. Davidov, D., Tsur, O., and Rappoport, A. 2010. SemiSupervised Recognition of Sarcastic Sentences in Twitter and Amazon, Dmitry Proceeding of Computational Natural Language Learning (ACL-CoNLL).
4. Derks, D., Bos, A. E. R., and Grumbkow, J. V. 2008. Emoticons and Online Message Interpretation. Soc. Sci. Comput. Rev., 26(3), 379-388.
5. Gibbs, R. 1986. On the psycholinguistics of sarcasm. Journal of Experimental Psychology: General, 105:3–15.
6. Gibbs, R. W. and Colston H. L. eds. 2007. Irony in Language and Thought. Routledge (Taylor and Francis), New York.
7. Kreuz, R. J. and Glucksberg, S. 1989. How to be sarcastic: The echoic reminder theory of verbal irony. Journal of Experimental Psychology: General, 118:374-386.
8. Kreuz, R. J. and Caucci, G. M. 2007. Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on Computational Approaches to Figurative Language (pp. 1-4). Rochester, New York: Association for Computational. LIWC Inc. 2007.
9. The LIWC application. Retrieved May 10, 2010, from http://www.liwc.net/liwcdescription.php.
10. Nigam, K. and Hurst, M. 2006. Towards a Robust Metric of Polarity. In Computing Attitude and Affect in Text: Theory and Applications (pp. 265-279). Retrieved February 22, 2010, from http://dx.doi.org/10.1007/1-4020-4102-0_20.
11. Pak, A. and Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining, in 'Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)' , European Language Resources Association (ELRA), Valletta, Malta
12. Pang, B. and Lee, L. 2008. Opinion Mining and Sentiment Analysis. Now Publishers Inc, July.
13. Pennebaker, J.W., Francis, M.E., & Booth, R.J. (2001). Linguistic Inquiry and Word Count (LIWC): LIWC2001 (this includes the manual only). Mahwah, NJ: Erlbaum Publishers
14. Strapparava, C. and Valitutti, A. 2004. Wordnet-affect: an affective extension of wordnet. In Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon.
15. Tepperman, J., Traum, D., and Narayanan, S. 2006. Yeah right: Sarcasm recognition for spoken dialogue systems. In InterSpeech ICSLP, Pittsburgh, PA.
16. Utsumi, A. 2000. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. Journal of Pragmatics, 32(12):1777– 1806.

Geeta Bhagwan Mehetre currently pursuing M.E in MSSCET, Jalna, Maharashtra. Area of research is Opinion Mining, Natural Language Processing.

Prof M. B. Kalkumbe currently professor at MSSCET, Jalna, Maharashtra.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)