Irony Detection in English Tweets

Amrita Sundari V[1]

Abstract— **Detection of Figurative languages like Irony, has recently been receiving an increase in attention in the computational linguistics research point of view. Recognition of such figurative languages are very crucial in many natural language processing tasks including sentiment analysis. In this study, I propose a system capable of detecting irony in twitter data. This is a challenging task considering the fact that the tweets do not necessarily follow the grammatical rules, and contains a lot of truncated words and punctuation.Also, I propose a variety of linguistic features that are easy to extract and a set of experiments to qualitatively measure the relevance of the features included in my model.**

Keywords: **Sentiment Analysis, Irony, Natural Language Processing**

I. INTRODUCTION

As a result of the increasing popularity of social media, a lot of valuable information of public opinions have moved online and it has become a primary source of data for many research disciplines including Natural Language Processing. The rich and diverse information in the user generated content, dynamically updated along with the change in recent trends, behavioural responses, social opinions and preferences, gives us a platform to analyse and recognize the implicit knowledge in recognizing the sentiment of the content.

In this project, one such form of figurative language which relies heavily on such implicit information ,Irony, has been dealt with. Even though Irony has been discussed and analysed in Linguistics and Philosophy [Uts00] [Wan13], there are no formal definition of irony. It can be generally defined as an utterance that has a literal meaning that is different from the intended meaning.This phenomenon is widespread in social media content and has proved to cause serious implication in NLP tasks like sentiment analysis, opinion mining etc due to the contradiction in the meaning inferred from such ironical utterances and thereby altering the polarity of the such sentences.

Eg. Wow. Another Rainy day. How Wonderful!!

The above sentence is an example of Irony and Regular Sentiment analysis will probably classify it as a positive statement. But the irony is noticeable because of our real world knowledge, where rainy day is considered to be an unpleasant day.This clearly contrast with the positive expression "How Wonderful".Therefore, in order to develop and refine the accuracy of the state-of-the-art models in sentiment analysis, it is essential to identify irony behind the sentence.

The Computational detection of Irony has gained recent upraise in the research perspective. Reyes et al(2013)[RRV13] proposed a method to detect irony with rich set of features and uses machine learning techniques and approaches the task as a classification problem. In similar terms Barbieri and Saggion (2014)[BS14] also proposed six sets of easily extractable features for the same. In this project, I have extracted and implemented the groups of syntactic, lexical and semantic features and have discussed the qualitative relevance of the features specific to the task of detecting irony. We mainly make use of the data from micro-blogging platform, Twitter, which allows users to post messages called tweets.

The rest of the report is organised as follows: the next section gives a brief description of the data set and the corpus that were used for the experiments. Section III describes about the preprocessing tools that were used , section IV gives us a brief overview of the methodoloy and the feature set, Section V describes the experiments and interpretation of the results and finally in Section VI the report is finished with a conclusion and future work.

II. CORPUS DESCRIPTION

The data that is used in the experiments and for the feature extraction has been prepared by Cynthia Van Hee(2013). The corpus consists of around 4000 tweets that were automatically extracted with the twitter hashtags #irony #sarcasm #not using the Twitter API. The data is annotated manually with the guidelines presented in Van Hee et al., 2016b[VLH16]. The annotation resulted in around 70% of data to be ironical. For the classification experiments, some more of tweets which were extracted randomly have been added to the corpus to make a balanced dataset.

For the experiments, I have split the balanced dataset into an 80% training corpus and 20% testing corpus resulting into a training set of around 3800 tweets.Both sets show a balanced data of 50% ironic and 50% non-ironic.

Another corpus is also incorporated in this approach to calculate the word usage frequency,American National Corpus frequency data[2] (Ide and Suderman, 2004).The corpus provides us with the number of occurances of a word in the spoken and written ANC.

[1] Amrita Sundari V is with Department of Electrical and Computer Engineering, Texas AM University, College Station, TX- 77840 `amrita95@tamu.edu`

[2] The American National Corpus (http://www.anc.org/) is, as we read in the web site, a massive electronic collection of American English words (15 million)

III. TEXT PREPROCESSING

The preprocessing of the twitter data is not a trivial task since micro blogs are not grammatically correct necessarily and most of the words are truncated with little context. For this reason, I use the function from tweet tokeniser from the nltk package. The irrelevant information such as URLs and references(name) have been removed with the help of Regular expressions and then tokenized. To process the truncated words, the function wordnetlemmatiser have been used to make the tokens into a meaningful lemma of each token.

Part of Speech tagger of NLTK package, has been used to tag the tokens. The POS taggers were used to find the semantic structure of the tweets in the corpus.

IV. METHODOLOGY

The task of Irony detection is viewed as a classification problem and experiments are conducted applying various supervised learning techniques to identify the algorithm that performs well for this task in particular. I have used the implementation of various classifiers like Decision Trees, Gaussian Naive Bayes, Support vector machine and Random Forest from the scikit learn packages.

The model that has been implemented uses different set of features which will capture the nuances of Irony, like unexpectedness and polarity imbalance. There are other set of features also which detects the common patterns like the structure of the tweets such as punctuation, length etc. The following are the sets of features I have extracted from the dataset.

- Usage of rare and common words
- Structure of tweets
- Internet Laughs and Emoticons
- Sentiments
- Synonyms and ambiguity

The descriptions of each feature set are briefly discussed in the following sections. Also, it mentions all the features that has been extracted in each group and the theoretical motivation behind the extraction of each feature.

A. Usage of rare and common words

The idea behind the usage of this feature is to model the unexpectedness of a some words in a tweet. Irony is connected to the element of surprise and it can be an important factor in the usage of situational ironies. With this set of features, we try to model the inconsistencies in the word usage in a single tweet and register the frequency of the words with the use of ANC Dataset. The idea is that the surprise element, a component of ironical sentence, might be from the imbalance created by using two types of words,namely commonly used words and rare words, in the same tweet.

Three features are defined in this group which are able to model the usage of above mentioned types of words, ***imbalance, rare, averagefreq***. First,I found the frequency of all the words in the tweet from the ANC corpus.The feature *averagefreq* is nothing but the average of the frequencies of all the words of that tweet and *rare* is the frequency of the word that has the least frequency value of all words. The *imbalance* is one of the important feature of this set and it is the difference between the frequencies of the rarest word and the most common words in the tweet. Intuitively, the more the value of imbalance, the higher the probability of irony being present.

B. Structure of Tweets

This group of features covers the common lexical features that are relevant to any natural processing task related to the structure of the tweet that has been taken into consideration. This includes three features namely **char, word, wordmean**. *Char* is the count of number of characters in the tweets (a-z0-9 including speacial characters).*word* represents the number of words in a single tweet, and *wordmean* is the average number of characters in a word.

With the help of POS tagger, I was able to group each word in the tweet into the lexical categories like noun,verb etc. I have constructed five more features **noun, verb, adjective, adverb** which is nothing but the number of words in the tweet that falls in each category. Also, I have added few more features **nounrat, verbrat, adverbrat, adjectiverat** which is the ratio of the previous features(noun,verb,..) to the total number of words in the tweet.

One more important feature set in this category is the features related to punctuations. Since written form of any figurative language like irony,sarcasm, etc lacks the features like facial expression, the tone of speaker's voice that is exclusive for the spoken form, the users made use of the punctuations to express their emotional features. Therefore the following features **punctuation, ellipses** have been added to the feature set. *Punctuation* is the number of question marks(?), exclamation marks(!), commas(,), hashtags(#) etc. in the tweet and *ellipses* is the number of three consecutive periods(...) in a tweet.

C. Internet laughs and Emoticons

This set of features is one of the interesting set that have been added. There are two different features, one being the **internet laughs** and the other **emoticons**. Nowadays, *Internet laughs* are considered as another form of punctuation in social media analysis, due to its incessant usage in the internet.The most common internet laughs in usage are *haha, lol, lmao, rofl*. I have used pattern matching techniques to match all combinations of those laughs like hahahhaa, looll, lmaoo etc. using Regular expression and the number of such laughing patterns are stored in the *internet laughs* feature. The *RegEx* pattern that I have used for this purpose is given below.

pattern='([aA]*[hH][Aa]+[Hh][HhAa]*—[Oo]?[Ll]+[Oo]+[Ll]+[OolL]*—[Rr][oO]+[Ff]+[lL]+—[Ll][Mm][Aa]+[oO]+).'

Similarly, the feature *emoticon* counts the number of emojis like :), :P etc that are present in a tweet. I have used a similar technique to the internet laughs to extract the emojis. In general, it is found that the irony corpus use the least number of emoticons probably because the ironic users tend to avoid emoticons and present the words to be in central.Therefore, this turned out to be a useful feature in detecting ironies.

D. Synonyms and Ambiguity

The choice of words the ironic users uses in their tweet follows a pattern. The hidden meaning behind the one that has been conveyed in the tweet depends on the choice of the synonyms that have been used in the tweet. In order to analyse this idea, I made use of the *wordnet* corpora to find out the synonyms set(*synset*) of each word in twitter.On each of such synsets, ANC frequency corpora have been used to find the frequencies of each synonyms and sorted them in the decreasing order. With this information, I have defined four features for this group namely, **syno lower, syno lower mean, syno lower gap, syno greater gap**.

The *syno lower* is defined as the number of synonyms of word w_i with frequency lower than the frequency of the word w_i itself. It is defined with the following equation.

$$sl_{w_i} = |syn_{w_i} : f(syn_{w_i}) < f(w_i)|$$

where syn_{w_i} is the synonym of word w_i and the function $f(x)$ returns the frequency of word x from the ANC dataset. The *syno lower mean* is nothing but the average of the number of syno lower over all the words in the tweet.Also, to define the next two features, I have measured two values,*word lowest syno*,the syno lower of the word with highest number of *lower synset values* and, *word highest syno* syno higher of the word with highest *higher synset values*. It can be understood from the following equation.

$$wls_t = \max_{w_i} |syn_{w_i} : f(syn_{w_i}) < f(w_i)|$$

$$whs_t = \max_{w_i} |syn_{w_i} : f(syn_{w_i}) > f(w_i)|$$

The next feature *syno lower gap* is defined as the difference between the highest syno lower and the syno lower mean. Similarly the feature *syno higher gap* is the difference between the highest syno higher and syno higher mean. The last two features is important in measuring the imbalance.It is found that that the average of the syno lower and higher mean is greater in ironic corpora than a normal corpora, suggesting that a very common(or very rare) synonym is often used in ironic tweets.

With an intuition that a word with more synonyms is more probable to be used in ironical sentences, since it can create the ambiguity in an irony, I define three more features **synset mean, max synset and synset gap**. The first feature is the average of the number of synonyms of each word over all the words in the tweet. The *max synset* is the number of synonyms of the word with maximum number of synonyms and the *synset gap* is the difference between the previous features.

E. Sentiments

The last and the most important feature of all is the feature set which highlights the polarity of sentiments in each tweets. The widely accepted definition of irony states that a sentence with contrast in polarities of sentiments is more likely to be an irony. With this is in mind, I define four features **positive sum, negative sum, positive negative mean, positive negative gap**.

For this purpose I used the sentiwordnet corpus which gives sentiments scores along with the synonyms of the word. For each word, positive and negative scores is taken to be the average of all the scores of its synonyms.The first feature *positive sum* is the sum of all the positive scores in the tweet and similarly *negative sum* is the sum of all the negative scores in the tweet. *Positive negative mean* is the arithmetic average of the positive and negative sum and the difference between them is the *positive negative gap*.

V. Experiments and Results

For the experiments, I have divided the corpus with 3834 tweets into training set (80%) and testing set (20%).As mentioned earlier, I have conducted binary classification experiments by exploiting the above mentioned features. I have conducted two sets of experiments, one before and after feature selection. The feature set contains 29 specially extracted features as explained in the previous section along with the Bag of Words and the final number of features are around 11000.

I have used 4 classification algorithms namely Linear SVM, Gaussian NB,Decision Tree and Random forest on both sets of experiments. Since the features that I have used are all continuous, I have used Gaussian Naive Bayes instead of classical naive bayes which is better in the case of categorical features. Considering the large number of features(including the BoW features), the non linear kernel is not going to improve the accuracy and so linear kernel for SVM is used, which is as good as a non linear kernel.

The Feature selection is done by constructing a heat map of the correlation matrix as shown in the Figure 1. As we can clearly see that the group of features with high correlation are *Synonyms and Ambiguity*. With this, we leave out the recurring features identified by the high correlation and also leaving out the features with very less correlation since it becomes irrelevant to the task.The experiments are conducted before and after feature selection.

Features	F-score	Accuracy	Precision	Recall
RareCommon	0.534	0.512	0.509	0.562
Structure	0.632	0.636	0.638	0.627
Laughs&Emoticons	0.629	0.639	0.644	0.614
Synonyms	0.444	0.542	0.562	0.367
Sentiments	0.627	0.639	0.646	0.609

TABLE I

Scores with only the specified features along with BoW

The Table I shows the experimental results obtained with each feature set seperately. As we expected from the figure 1

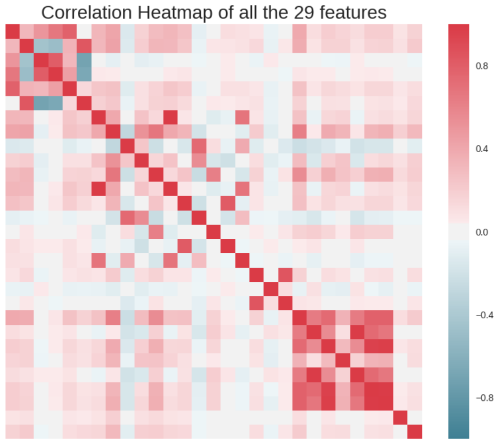

Fig. 1. Correlation matrix

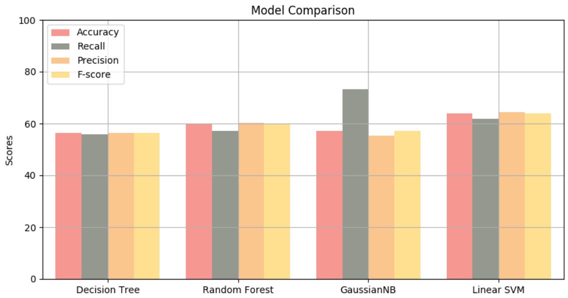

Fig. 2. Scores of the system with different classifiers

, the scores are comparitively low when only RareCommon features and Synonyms features are taken into account. This is because, the correlation is very high among the feature set indicating that the features are mostly redundant.The lexical feature set,*Structures* and the Sentiment features have the highest scores indicating that they are the most important features in irony detection task. An explanation for this could be because of the nature of Twitter data since the length of the tweet is limited, users tend to use lexical clues more explicitly while expressing irony. As expected Sentiment features are one of the important features of the detecting irony since polarity contrast is a part of the definition of irony. As per our intuition, laughs and emoticons features also provides enough information to classify ironies and the scores are comparable to the other feature sets.

Experiment	F-score	Accuracy	Precision	Recall
Without Feature Selection	0.54	0.505	0.503	0.599
With Feature Selection	0.63	0.639	0.643	0.6185

TABLE II
RESULTS OF THE EXPERIMENTS

The Table II compares the two experiments: before feature selection and after feature selection with linearSVM as the classification algorithm

The Figure 2 shows the bar plot of the scores of different classifiers that we have used for the experiment. We can see that Linear SVM performs the best with F-score of 63% with Correlation-based feature selection. The accuracy of the system increases significantly when the irrelevant and highly correlated(redundant) features are removed from the feature set.

VI. CONCLUSIONS

In this project, I have applied classification algorithms on the corpus of twitter data and have developed some easily extractable feature sets that are relevant to the task of irony detection. The F-score obtained by the classifier system after applying correlation based feature selection is 63% and the Linear SVM algorithm is used for classification. I have compared and contrasted the information gain of each feature set on this task with the help of correlation matrix. Also, I have applied different ML classifiers for this problem and it turns out Linear SVM performs the best among all.

A Quantitative analysis of the features revealed that the Sentiment features set are more important. Also, we show that lexical features provide relevant information even though it is assumed insufficient for decent irony recognition.The sentiment features performed best when there is an explicit polarity contrasts in the tweet. Future research should focus on evaluating implicit sentiments with the help of world knowledge and common sense. This makes identifying polarity contrasts which are implicit also.

REFERENCES

[Uts00] Akira Utsumi. “Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony”. In: *Journal of Pragmatics* 32.12 (2000), pp. 1777–1806.

[RRV13] Antonio Reyes, Paolo Rosso, and Tony Veale. “A multidimensional approach for detecting irony in twitter”. In: *Language resources and evaluation* 47.1 (2013), pp. 239–268.

[Wan13] Po-Ya Angela Wang. “# Irony or# Sarcasm—A Quantitative and Qualitative Study Based on Twitter”. In: (2013).

[BS14] Francesco Barbieri and Horacio Saggion. “Modelling Irony in Twitter.” In: *EACL*. 2014, pp. 56–64.

[VLH16] Cynthia Van Hee, Els Lefever, and Véronique Hoste. *Guidelines for Annotating Irony in Social Media Text*. Tech. rep. version 2.0. Technical Report 16-01, LT3, Language and Translation Technology Team–Ghent University, 2016.