Modelling Satire in English Text for Automatic Detection

Aishwarya N Reganti, Tushar Maheshwari,
Upendra Kumar, Amitava Das
IIIT, Sri City, Chittoor
India
{ aishwarya.r14, tushar.m14,
upendra.k14, amitava.das}@iiits.in

Erik Cambria
School of Computer Engineering
NTU, Singapore
cambria@ntu.edu.sg

Abstract—**According to the Merriam-Webster dictionary [1], satire is a trenchant wit, irony, or sarcasm used to expose and discredit vice or folly. Though it is an important language aspect used in everyday communication, the study of satire detection in natural text is often ignored. In this paper, we identify key value components and features for automatic satire detection. Our experiments have been carried out on three data sets, namely, tweets, product reviews and newswire articles. We examine the impact of a number of state of the art features as well as new generalized textual features. Together using these features, we outperform the state of the art by a significant 2-6% margin.**

***Keywords*—satire detection, computational creativity, figurative language, sentiment amplifiers, continuity disruption**

I. Introduction

Figurative language is language that uses words or expressions with a meaning that is different from the literal interpretation. When a writer uses literal language, he or she is simply stating the facts as they are. Figurative language is used with a meaning that is different from the basic meaning and that expresses an idea in an interesting way by using language that usually describes something else. So thus, one of the greatest challenges in computational linguistics is figurative language processing, since the words or expressions used possess a meaning that is different from the literal interpretation. Satire is one such form of figurative language that demands acute analysis and reasoning. Predictive models that can detect satire with reasonable accuracy can be beneficial in many applications involving customer review analysis, natural language user interfaces, opinion mining, automatic reply suggestion systems etc. Although there have been previous research works on satire detection, to the best of our knowledge, these works are restricted to a single domain of text like social media posts, product reviews etc. In this paper, we propose a set of generalized linguistic features which provide encouraging results for different kinds of corpora.

Generally, four types of satire can be defined in English Language.

Exaggeration - *:* To enlarge,emphasize or portray something beyond normal bounds so as to highlight faults. For example, consider this tweet that was posted by an anonymous user-

"I'm super excited today!! so much that I'd kill myself"

Incongruity - *:* To present things that are out of place or are absurd in relation to its surroundings. Consider this example which was picked out from a product review-

"The back camera of the phone is so good that I can capture every atom of a scenery"

Reversal - *:* To present the opposite of what must be actually conveyed. The below example was picked from a product review where the user initially stressed on the shortcomings of the product (camera) and later made a satirical comment-

"I'm extremely disappointed. Not as expected!. It's just amazing how the flash works !"

Parody - *:* To imitate the techniques and/or style of some person place or thing. Consider this post by a user on twitter-

"My mistress, I was truly touched by your dumbness".

The type of satire used depends on the source from which the text is retrieved. It can be observed that, generally, product reviews are either of the 2nd or 3rd type. Newswire articles are majorly of the 2nd type. while social media posts are majorly of the 1st and 4th type. There are certain outliers, however on a large scale the above trend can be generalized. Since the system must be capable of detecting satire in all kinds of corpora, linguistic features must be rightly chosen to detect all the above types of satire. In the following sections of the paper, we propose various features to detect satire, we also experiment with different classifiers to obtain optimum results.

The major contributions of this paper are : (1) We introduce a novel approach to binary classification of satire in English text. (2) We propose a list of generalized linguistic features which provide benchmarking results on different types of satire corpora. (3) We make available a standard satire corpus which was retrieved from twitter (with user generated tags such as #satire, #satirical)

The rest of the paper is structured as follows: In the Section II we elucidate the previous works carried out in this area. In section III, we report the statistics of the three corpora that have been used in the paper. In section 4, we present our set of features and classifiers for automatic detection. In section 5, we elucidate the results obtained on the three corpora using different features and classifiers. In section 6, draw inferences from our results. In the last section, we conclude with a summary and an overview of possible future work.

II. Related Work

As stated previously, satire is a general term referring to any form of wit, irony or sarcasm used to ridicule something/someone. Our research mainly focuses on binary classification of a given instance into satirical and non-satirical classes. It is quite cumbersome to obtain a corpus with labelled text for satire detection since manual effort is required to analyse the text as and there is a possibility that the satire is context based which makes it highly challenging for automatic systems to detect since the context is not known. Such occurrences must be manually filtered out. One such labelled dataset containing ironic product reviews, crawled from Amazon was collected by [2] in 2012. The reviews were annotated by crowd-sourcing the reviews and considering inter-annotator agreement. The corpus can be used for identifying irony on two levels: a document and a text utterance. A sarcasm corpus was created by [3]. The corpus was automatically collected by extracting statements using Google Book search, which ended with the phrase ”said sarcastically”. They also performed a regression analysis on the corpus so obtained, exploiting the number of words as well as the occurrence of adjectives, adverbs, interjections, exclamation and question marks as features. Many such approaches have been proposed to detect irony and sarcasm based on common lexical patterns and general structure. In 2010, [4] devised a semi-supervised system to detect irony in tweets and Amazon product reviews. Their work exploits features such as sentence length, punctuation marks,the total number of completely capitalized words and automatically generated patterns which are based on the occurrence frequency of different terms. in 2009, [5] devised an approach to detect irony in user generated contents using features such as emoticons, onomatopoeic expressions for laughter, heavy punctuation marks, quotation marks and positive interjections. In [6], irony detection is carried out on product reviews using various linguistic features like emoticons, punctuations, hyperbole, ellipses etc.. [7] develop a system to detect ironical tweets using pattern detection techniques and lexical features. In 2012, [8] proposed a novel approach to detect irony and humor, two major elements of figurative language. Features such as linguistic devices, ambiguity, incongruity, and meta-linguistic devices,such as polarity and emotional scenarios were used to build their predictive model. A model of irony detection assessed along two dimensions for twitter posts was proposed in [9]. The SemEval-2015 task 11 [10] was wholly dedicated to analyzing figurative language on Twitter. Three classes of figurative language were considered. (irony, sarcasm and metaphor). Participating systems were required to provide a fine-grained sentiment score on an 11-point scale. Several works have also been carried out to detect sarcasm in spoken language, for example, [11] in 2006. However, to the best of our knowledge, there has been no work so far, which specifies linguistic features which can work reasonably well for satirical instances from different sources. Most research works have restricted by domains like Social media posts/Product reviews/Discussion forums/ News articles but not all cumulatively. In our paper, we develop a framework that detects satire with good accuracy from almost all kinds of sources of text, since we test the model on three entirely different kind of corpora.

Corpus	Total	satirical	Non-satirical
Product Reviews	1254	437	817
Newswire Articles	4000	233	3767
Twitter posts	8,000	3,000	5,000

TABLE I: Corpus statistics

III. Datasets Collected and Used in this Study

In order to test the robustness of the proposed model across different domains we use product reviews crawled from Amazon, tweets and news documents in our experiments. The statistics of all the three datasets are reported in Table I.

A. *Amazon Product Reviews*

We have used the corpus created by Filatova[2] in 2012. This data set consists of 1,254 Amazon product reviews reviews, of which 437 are ironic and 817 are non-ironic. Since we started with the notion that satire is a super class of language devices including irony and sarcasm, we used this corpus to test our models. A crowd sourcing platform called Amazon Mechanical turk [12] was used in order to obtain labels for a given list of product reviews. Initially, a set of turkers were asked to submit pairs of reviews from Amazon, describing the same product, with one being ironic and the other, non-ironic. Later, a second task was hosted on Amazon mechanical turk to classify the previously submitted pairs into ironic and non-ironic. This task was done to ensure that the submitted reviews were indeed ironic and eliminate spammers' submissions. Each review was presented to 5 turkers for inter annotater agreement. Two quality control procedures were used to eliminate spam and ensure quality data. They were : simple majority voting and the data quality control algorithm that is based on computing Krippendorffs alpha coefficient [13] to distinguish between reliable annotators and unreliable annotators. These measures ensured that the labels from reliable annotators get high weight in computing the final label for a data point.

B. *Newswire Documents*

This corpus was released by [14] in 2009, This corpus contains a total of 4000 newswire documents and 233 satire news articles. The newswire documents were randomly sampled from the English Gigaword Corpus. The documents were obtained by issuing google search queries on a particular phrase and filtering all the non-newsy, irrelevant and overly-offensive documents from the top-10 documents returned from the search. All newswire and satire documents were then converted to plain text of consistent format using lynx, and all content other than the title and body of the article was manually removed (including web page menus, and header and footer data). The number of satirical documents was intentionally made lesser than the number of regular documents since it reflects a realistic picture of the web where very few satirical articles are found

Type	Total words	No. of positive words	No. of Negative words
Satirical	19	5	2
Non-satirical	18	4	1

TABLE II: Twitter Posts Corpus Statistics

C. Twitter posts

In today's world, social media platforms play a very important role in everyday life. We can indeed say that social media is good proxy of the society. Therefore, social media posts came as a natural choice for us. We chose twitter for this purpose. As of the first quarter of 2016, the micro blogging service Twitter averaged at about 236 million monthly active users, with around 6,000 tweets being posted every second. Therefore, twitter is definitely a rich source of data. The data was retrieved using the search query option of twitter4j rest API. We used "#satire", "#irony" & "#sarcasm" as the query terms. There were some time-based satirical tweets. For example, consider a tweet that was retrieved using the query #satire:

"Sreesaanth (Indian Cricketer), u jus rocked it"

This tweet was posted in 2013 when the cricketer was arrested under allegations of spot fixing. Such tweets are tricky to predict, since they are dependent on the date on which the post was made. Had the same tweet been posted in 2006 or 2007 when the cricketer was celebrated and prominent, the tweet could be classified as non-satirical. Such ambiguous tweets which were tricky to analyse, even for human beings, were filtered off, since additional learning and knowledge is required to analyse such posts. Three annotators were assigned the task of annotating the tweets which were retrieved using hash-tag search. They were asked to classify the tweets into "satire" and "non-satire", inter-annotator agreement was considered and tweets with more than two or more votes were considered to be belonging to the respective class. Finally, we retrieved 3000 satirical tweets. In order to populate the corpus with non-satirical tweets we used "#health", "#food", "#news" & "#education" and obtained non-satirical tweets. On an average, each tweet contains 18 words. To find the pattern and polarity of words used in tweets, we found the number of positive and negative words in each tweet using SentiWordNet. The average number of positive and negative words in satirical and non-satirical tweets have been reported in Table II. However, we notice that no differentiating pattern cannot be observed between satirical and non-satirical tweets. Therefore, lexical polarity alone will not be sufficient to distinguish tweets. We must also remember that the structure of tweets is quite different from that of product reviews and newswire articles since the maximum limit of a tweet is 140 characters, hence a lot of twitter users use abbreviations, phonic based spellings etc. which makes the task of satire detection in twitter even more challenging.

IV. Architecture

We model the task of satire detection as a supervised classification problem in which each instance is categorized as being satirical or non-satirical. We examine different classifiers and features that affect the accuracy of our system. We use seven sets of features to build our model. In the next subsections, we describe the features used and the set of classifiers compared. Table III provides an overview of the group of features in our model and table IV elucidates the length of the feature vector in each group.

Baseline Features

Undoubtedly, n-grams are the best task-independent features for any kind of textual classification [15]. Hence we chose n-grams as our baseline features, since task-independent features are necessary to detect satire as the difference between positive and negative classes is subtle and using only task-specific features does not yield very good accuracy. We retrieved character n-grams (bi-grams and tri-grams), word n-grams/ Bag of words(bi-grams and tri-grams) and skipgrams(bi-grams) from our corpus. We filtered out all ngrams whose frequency were less than three, in order to ensure that only essential n-grams remained. This set of features is used as our baseline.

Lexical features

Two sentiment lexicons were made use of. The NRC emotion lexicon [16] contains about fourteen thousand words. The lexicon has affect annotations for each word. Each word is tagged with either one of the 2 sentiments: negative & positive, or one of the 8 emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, trust. From these words, only words with annotations anger, anticipation, disgust, fear, joy, sadness & surprise were chosen as satirical sentences generally contain words with extreme emotions like anger or joy. SentiWordNet [17] is one of the largest sentiment lexicons with about words. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity. We calculated the net score for each word in the lexicon using the formula

$$Net\_Score = Positivity\_Score - Negativity\_Score$$

Words with a positive net score were considered to be of the positive sentiment while words with a negative net score were considered to convey negative sentiment. Words with a net score 0 were assumed to be neutral. The number of positive, negative and neutral words in the corpus were used as features.

Sentiment Amplifiers

As a general trend, it can be observed that almost all satirical utterances use one or the other form of sentiment amplifiers. Sentiment amplifiers are those elements which highlight an emotion or intensify it. Amplifiers such as exclamation marks, quotes, ellipses etc.. are used to emphasize the sentiment conveyed in the statement. Amplifiers like quotes draw attention towards a certain piece of enclosed text, since satirical statements generally express strong emotions towards someone/something, it is highly probable that amplifiers are used in satirical statements. The feature "quotes" indicates that up to two consecutive adjectives or nouns in quotation marks have a positive or negative polarity [6]. The Punctuation feature conveys the presence of an ellipses as well as multiple question or exclamation marks or a combination of the latter two. In social media texts, emoticons, slang words, acronyms and interjections act like amplifiers. The interjection feature indicates words like "heh", "oh", "wow" etc.. A list of trending acronyms in sms jargon like "LOL", "TTYL" are a part of

No.	Group	Features
1	Baseline Features(BF)	character n-grams, word n-grams, word skipgrams
2	Lexical Features(LF)	NRC Emotion lexicon, SentiWordNet
3	Literary device features(LD)	Hyperbole, Alliteration, Inversions, Imagery, Onomatopoeia
4	Sentiment Amplifiers(SA)	Brackets, Ellipses, Quotes, Question marks, Exclamation marks, Interjections, Emoticons, Slang words, Acronyms
5	Speech Act Features(SAF)	As Illustrated in Table 3
6	Sensicon Features(SE)	Sense scores for Sight, Hearing, Taste, Smell and Touch
7	Sentiment Continuity disruption features(SCD)	Count of Flips

TABLE III: Feature groups used for Satire Detection

Feature Group	BF	LF	LD	SA	SAF	SE	SCD
Feature Length	len(Ngrams)	11	5	9	11	5	1

TABLE IV: Feature Length Of Different Groups

No.	Speech Act	Example
1	Action Directive	Just fill out this application
2	Apology	Im sorry. There are no sales today
3	Appreciation	Thanks. I really appreciate that
4	Response Acknowledgment	Okay, but let me know ahead of time
5	Statement Non-Opinion	I am unique Carbon atom
6	Statement Opinion	Doctor, I feel like a pack of cards.
7	Thanking	Thank you! Ill try back later
8	Wh Question	Why dint you call me yesterday?
9	Yes Answers	Yes, I know what you mean
10	Yes-No Question	Is your phone out of order?
11	Other	Ill deal with you later

TABLE V: Types of speech acts with examples

the acronym feature list. Emoticons like “:)” (Smiling face), “:(”(sad face), etc. form the emoticon feature list. Words like “awsum” (awesome), “gr8”(great), “skul”(school) which form a part of day to day social media text were added into the slang word feature list. The presence or absence of the above mentioned sentiment amplifiers is used to form the features.

Speech Act Features

A speech act in linguistics is an utterance that has performative function in language and communication [18]. In short, it is the action that lies in utterances such as apology, appreciation, promise, thanking, etc. Here, in this paper, we use 11 major (avoiding 43 fine-grained speech act classes) speech acts to classify text (as illustrated in table V)

A speech act classifier was built using the SPAAC (Speech act annotated corpus) [19]. The SVM-based speech act classifier was developed using the following features: bag-of-words (top 20% bi-grams), presence of wh words,presence of question marks, and sentiment lexica such as NRC Linguistic Database, SentiWordNet. The features used in the classifier and respective accuracies have been indicated in table VI. [20]. The classifier so built achieved an Accuracy of 70% after 10 fold cross validation. This classifier was used to obtain the speech act distribution for our satire corpora. Since speech act is determined at the sentence level, a speech act distribution was obtained for text containing more than one sentence.

$$(Speech\,Act\,Distribution)_n = \frac{(Sentences)_n}{Total\,number\,of\,Sentences} \tag{1}$$

To obtain the speech act distribution for text with more than one sentence, the above formula was used. The distribution of a speech act n was found by calculating the number of sentences that were predicted to possess the speech act and dividing the number by total number of sentences in the text. Hence, 11 new features indicating speech act distribution were used. Automatic speech act classification of social media conversations is a separate research problem altogether, and hence out of scope of the current study. However, although the speech act classifier was not highly accurate in itself, the text specific speech act distributions can be used as features for satire detection.

No	Features	Accuracy
1	Only bag-of-words(BW)	55.75%
2	BW + WH-words(wh)	57.02%
3	BW + Question mark(qm)	62.97%
4	BW + SentiWordNet(senti)	67.75%
5	BW + NRC(nrc)	63.22%
6	BW + wh + qm	64.18%
7	BW + wh + qm + senti	69.41%
8	BW + wh + qm + senti + nrc	70.33%

TABLE VI: Features used for speech act classifier

Sensicon Features

Sensicon is a sensorial lexicon that associates English words with senses[21]. It contains words with sense association scores for the five basic senses: Sight, Hearing, Taste, Smell, and Touch. For example, when the word ‘apple’ is uttered, the average human mind will visualize the appearance of an apple, stimulating the eye-sight, feel the smell and taste of the apple, making use of the nose and tongue as senses, respectively. Sensicon provides a numerical mapping which indicates the extent to which each of the five senses is used to perceive a word in the lexicon. Generally, when someone makes a satirical statement, the purpose is to express disgust/anger in a creative manner which simulate senses. Therefore, we wanted to analyse if the sense scores had any relation with satire. The cumulative sensicon scores for each instance of the corpus were used as features, therefore a total of 5 features, referring to each of the 5 senses were added as features.

Sentiment Continuity disruption features

Consider this anonymous amazon product review (on Mr. Beer Premium Edition Home Microbrewery System (Kitchen)) which was picked up from the Filatova corpus.

“I made several batches of beer with a variety of mixes, waters and techniques. All resulting in a barely drinkable malt beverage (I refuse to use the term beer). Even though I changed up the mixes and used different recipes every batch tasted the same. Ben Franklin once said ”Beer is proof that God loves us and wants us to be happy” The product produced by my Mr Beer was proof that the devil exists and

he likes to play jokes on us."

The below review was made on a video game (Nintendo DSi Matte - Black (Video Game))

"Great, buy a more expensive piece of hardware so you can download games that are locked to it. This is a great step in the direction of renting all your games. No thanks. Plus, you lose backwards compatibility of the GBA. Shorter battery life than the DS Lite! The DS lite is a cheaper and better portable system."

The above reviews were labelled as satire in the product review corpus. On close analysis, one can observe that the user initially starts off with one kind of sentiment either positive or negative, and then flips polarity somewhere in between. In the first example, the user makes a few negative statements and then flips polarity in the statement-"Beer is the proof...." where the satirical statement begins. In the second review, the user starts off with a satirical positive statement and then flips polarity in the sentence "Plus, you lose.....". As a general trend, it can be observed that in large text, consisting of more than one sentence, and satirical statements, the polarity flips at least once, either when the satirical statement ends or when the satirical sentence starts. The more the number of flips in the text, more the number of satirical sentences, and hence a stronger satire. We used the number of flips in the text as the "Sentiment Continuity disruption" feature. In order to calculate polarity of the sentence we used the TextBlob package in python. However, this feature might be of more importance only for texts with more than one sentence (Here, product reviews and newswire articles) since short texts do not show this property. It is expected that the feature will not work well for twitter posts. The below algorithm explains the procedure to obtain the Sentiment Continuity Disruption Score.

Algorithm 1 find_continuity_disruption(data)

```
$tknz\_sents \leftarrow Tokenize(data)$
$count \leftarrow 0$
$i \leftarrow 0$
$curr\_polarity \leftarrow textblob\_polarity(tknz\_sents[0])$
loop
    $prev\_polarity \leftarrow curr\_polarity$
    $curr\_polarity \leftarrow textblob\_polarity(tknz\_sents[i])$
    if $prev\_polarity \neq curr\_polarity$ then
        $count \leftarrow count + 1$
    end if
    $i \leftarrow i + 1$
    if $i \geq len(tknz\_sents)$ then
        $return\ count$
    end if
end loop
```

Literary device features

Since satire is all about expressing ones disgust/anger in a creative indirect way, we used the presence of certain literary devices that are generally used in satirical statements, as features.

Hyperbole- Hyperbole is a literary device used to over exaggerate something such that it cannot happen in the real word. For example, " I've been waiting for ages" is a statement which is not logically possible. According to [22], the feature Hyperbole indicates the occurrence of a sequence of three positive or negative words in a row.

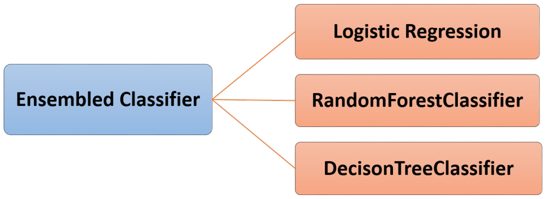

Fig. 1: Ensemble Classifier

Alliteration- The occurrence of the same letter or sound at the beginning of adjacent or closely connected words is termed as alliteration according to Google Dictionary. For example, "Bright Boy", "Dans Dog", "Fred's friends".

Inversion- Inversion is a literary device generally used in written English, where the formal structure of the sentence is inverted to stress on a specific subject. 3 kinds of inversions[23] usually used are:

1) *Adjective after Noun*- e.g: soldier strong
2) *Verb before subject*- e.g: Shouts the policeman
3) *Noun before Proposition*- e.g: worlds between

Imagery- Imagery involves the usage of words such that physical senses are triggered. For example, "It was dark and dim in the forest", here, dark and dim simulate a visual image, "He whiffed the aroma of brewed coffee" here, "whiff" and "aroma" evoke our sense of smell. A list of imagery words were collected from various sources and used as features.

Onomatopoeia- Onomatopoeia are words which create a sound effect that mimics the described topic. For example, "achoo", "thud", "bang". A list of onomatopoeia were collected from various sources and used as features.

The tables VIII, IX and X show the F-scores obtained on three corpora, using 5-fold cross validation. The Scikit-Learn package in python was used to evaluate the results. Five different classifiers have been used, Logistic Regression(LR), Random Forest(RF), Support Vector Machine(SVM), Decision Tree(DT) and an ensemble of classifiers for better performance. From table 4 it can be inferred that Logistic Regression and Random Forest outperform other classifiers by a good margin on product review corpus. Whereas on Twitter corpus performance of Logistic Regression is better than other classifiers. In general, Random Forest seems to perform poorly as compared to Logistic Regression due to possibility of overfitting on corpus. We tried yet another classifier based on ensemble of classifiers which has been found to be effective when the performance of predictive models must be improved without over- fitting. They can be used to achieve broad solution spaces by multiplying combinations of best component search spaces.

In order to select or design best component search spaces, the individual components should be independent in order to assimilate less correlated information from the data. Therefore, Pearson correlation was found between different classifiers as

	LR	RF	SVM	DT
LR	1.00	0.74	0.57	0.44
RF	–	1.00	0.63	0.38
SVM	–	–	1.00	0.71
DT	–	–	–	1.00

TABLE VII: Pearson correlation between classifier predictions

reported in the table VII. We constructed an ensemble of three classifiers : Logistic Regression(LR), Random Forest Classifier(RF) and Decision Tree Classifier(DT), based on weighted majority voting scheme(Figure 1).

$$\begin{aligned} EnsembledClassifier = 0.6 * LogisticRegression+ \\ 0.3 * RandomForestClassifier+ \\ 0.1 * DecisionTreeClassifier \end{aligned} \tag{2}$$

We selected Logistic Regression because it's performance was found to be best among all other classifiers. Random Forest Classifier and Decision Tree Classifier (which is least correlated with Logistic Regression) was selected in order to capture the non-linear signals since the correlation between these two classifiers was the least. In the ensembled classifier SVM was dropped because neither it's performance was found to be good nor it's correlation was found to be least with Logistic Regression. Our selection of such kind of ensemble is based on the expectation that a collective decision of inferior and less correlated models may help to reduce few erroneous choices made by the best predictive model. The weights were given to each component based on least cross entropy error. We explored the best combinations of weights for each of the three components in the search space by iteratively running over all combinations of $w1, w2, w3$ and choosing a value where minimum cross entropy was obtained.

$$\begin{aligned} S :& \{(w1, w2, w3) | w1 + w2 + w3 = 1.0 \\ & where\, w1, w2, w3 \in \{0.1, 0.2, ..., 0.9\}\}. \\ & w1\, is\, weight\, of\, LR, \\ & w2\, is\, weight\, value\, of\, RF \\ & w3\, is\, weight\, value\, of\, DT \end{aligned} \tag{3}$$

V. Evaluation

The cross entropy results are reported in Figure 2. We observe that the minima exists at $w1 = 0.6, w2 = 0.3, w3 = 0.1$. Considering these values, the weights for the 3 classifiers were assigned. This simple ensemble based learning boosts the performance of our satire predictive model significantly. These results are also closer to our general intuition that multiple predictive models should collectively perform better than a single predictive model and are consistent with our expectations.

A. *Product Review Corpus*

No	Features	LR	RF	SVM	DT	Ensemble
1	BF	70.33 %	66.57 %	65.25 %	66.71 %	73.91%
2	BF+ LF	71.83 %	67.09 %	65. 66 %	66.79 %	73.82%
3	BF+ LD	69.89 %	66.82 %	66.02 %	66.68 %	73.22%
4	BF + SA	71.33 %	66.93 %	65.23 %	66.92 %	73.15%
5	BF + SAF	73.42 %	68.02 %	65.99%	65.22 %	75.66%
6	BF + SE	71.22 %	65.60 %	65.33 %	66.03 %	73.33 %
7	BF + SCD	72.01 %	66.61 %	65.56 %	67.05 %	74.88 %
8	All Features	75.30 %	68.93 %	66.63 %	67.11 %	77.96 %

TABLE VIII: F-Scores for product review corpus

No	Features	LR	RF	SVM	DT	Ensemble
1	BF	73.23 %	69.16 %	72.89 %	68.85 %	74.99 %
2	BF+ LF	73.32 %	69.27 %	72.76 %	68.82 %	74.82 %
3	BF+ LD	73.24 %	70.48 %	72.99 %	68.63 %	74.11 %
4	BF +SA	75.26 %	71.22%	73.91 %	70.13 %	76.91 %
5	BF + SAF	74.32 %	70.48 %	72.10%	68.89 %	74.86 %
6	BF +SE	73.02 %	70.02 %	74.03 %	69.66 %	74.58 %
7	BF +SCD	73.24 %	70.71 %	72.56 %	68.01 %	74.43 %
8	All features	76.89 %	71.06 %	74.03 %	68.11 %	78.16 %

TABLE IX: F-Scores for Twitter posts Corpus

No	Features	LR	RF	SVM	DT	Ensemble
1	BF	70.12 %	62.12 %	68.11 %	63.16 %	69.04 %
2	BF+ LF	72.77 %	61.23 %	68.45 %	63.55 %	75.32 %
3	BF+ LD	71.33 %	63.33 %	68.18 %	65.11 %	75.11 %
4	BF +SA	70.16 %	62.19 %	69.67 %	63.44 %	74.66 %
5	BF + SAF	73.24 %	61.90 %	68.99%	63.77 %	76.98 %
6	BF +SE	69.08 %	62.88 %	68.78 %	62.68 %	74.77 %
7	BF +SCD	71.88 %	63.22 %	69.12 %	63.33 %	75.77 %
8	All features	75.88 %	63.89 %	69.34 %	63.22 %	79.02 %

TABLE X: F-Scores for Newswire Corpus

From table VIII, we observe that the best F-score is obtained is 77.96% (Using Ensemble Classifier) which outperforms the the state-of-the-art for this corpus as proposed by [6]. It must also be noted that the star features, (indicating the number of stars that the user has rated the product), which provided a major boost to the F-Score have not been used by us, since we wanted to propose generalised features for all kind of corpora. We observe that speech act features performed the best on an average, while literary devices performed the least.

B. *Twitter posts corpus*

Table IX summarizes the experiments performed over the Twitter corpus. We observe that the best F-Score was again obtained by using the Ensemble Classifier. In this corpus however, the contribution of features displays a different trend. It can be noticed that on an average, sentiment amplifiers perform the best, this major boost in the performance can be due to the fact that users on social media use a lot of emoticons, acronyms, slang words etc., as compared to product reviews or newswire articles. As expected, the performance of Continuity disruption is not very good due to nature of text in social media. The maximum F-score obtained was 78.16% using ensemble classifier.

C. *Newswire articles corpus*

Table X displays the results obtained for the newswire corpus. The highest F-Score is obtained using the ensemble classifier is 79.02% which almost equals the state-of-the-art for this corpus, proposed by [14]. We can notice that speech act features work very well for the corpus. Since newswire articles are lengthy, speech act distribution can be discerned fitly. We also observe that sentiment amplifiers do not work very well, since newswire documents are composed formally, without the usage of slang or exuberant punctuation. Continuity disruption features do not boost the performance much, probably because a major number of satirical newswire articles

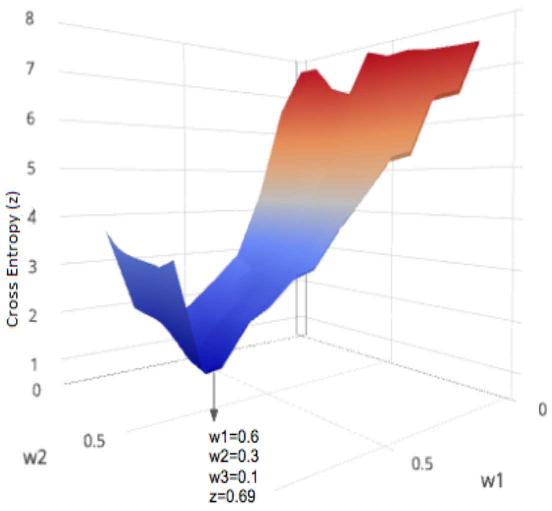

Fig. 2: Plot displaying minima of cross entropy values over weight space

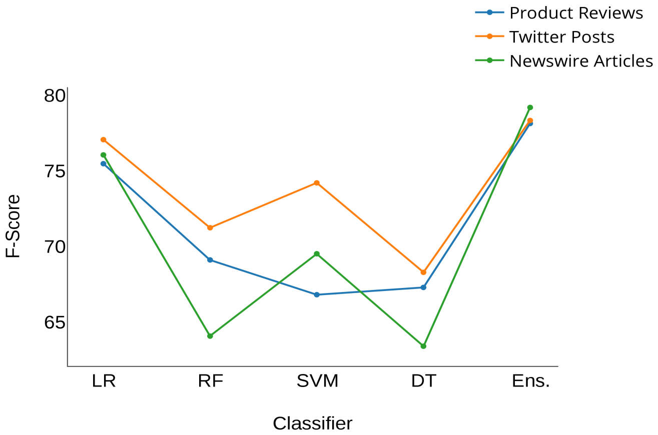

Fig. 3: Performance over different classifiers

are entirely satirical, as compared to product reviews where a few sentences are satirical and the rest are true sentences.

VI. Discussion

From the obtained results (The F-Scores of all the classifiers over the three corpora are displayed in figure 3), we observe that the features proposed work reasonably well for all corpora since our system outperforms the state-of-the-art for product review corpus and equals the state-of-the-art for newswire corpus. Since the twitter post corpus was created by us, we cannot draw any comparative analysis on it.

We observe that certain features are source-dependent, i.e., they work differently for different corpora. For example, the sentiment continuity disruption feature. This feature works well particularly for the product review corpus, because of the suitable expected structure in the product review corpus. Task-independent features like n-grams work equally on all corpora, since they draw characteristics of the corpus itself. We can notice that the speech-act displays a proficient performance in all corpora which suggests that speech-act of a sentence determines its possibility of being satirical. However, Sensicon features do not perform very will in either of the corpora, which disproves our intuition that satirical statements are sense simulating.

A crucial observation worth mentioning is the performance of the Ensemble classifier. It can be observed that in all three corpora, the Ensemble classifier leads, by a large margin. Our choice of classifiers for the ensemble, based on the cross entropy calculations proved to be worthwhile

VII. Conclusion and Future work

In this paper we have presented an approach to classify text from various sources into satirical and non-satirical. We examined the impact of a wide range of features and classifiers to obtain optimum performance. To the best of our knowledge, this is one of the first attempts to classify text from different kinds of sources using the same set of features. Our model beats the benchmark F-Score obtained for product review corpus. The performance obtained on social media posts is encouraging as well. We observe that n-grams work as good task independent features and hence are suitable for any text classification task. On an average, we notice that the Ensemble classifier boosts the performance by a good margin which proves that our intuition was true. The ensemble model works well over other individual predictive models such as SVM, RandomForest etc. because the classifiers used in the ensemble were chosen in such a way that the shortcomings of one classifier were compensated by the other.

There is however still scope for improvement, although the performance of the system is good on all the three corpora, task independent features contribute to the boost in performance. Future research should focus on the finding out new approaches by analyzing the vocabulary used in the text more extensively. We expect that a major number of satirical statements use words and phrases which are non-typical for the specific domain. Such occurrences can be detected with text similarity methods. The confidence of satire detection can be further improved if the personality of the user is determined, therefore embedding personality detection systems can help boost the performance especially in social media platforms where plenty of information about the user is accessible. The users previous posts, friend list, topics of interest, etc can help detect if posts made by him/her are satirical. A few satirical posts on social media platforms are time based as mentioned before, our system might not perform very well, therefore in our future work, we would like to develop a system which can compare the sentiment polarity of the topics in the post, with their polarity as perceived by the outside world. This can be achieved by retrieving the polarity of the extracted topics of the post from the World Wide Web. Satirical posts can be differentiated by the fact that they possess polarity opposed to general perception. Better ensemble classifiers [24] can be constructed, by blending/stacking methods with a single-layer logistic regression model is used as the combiner. We would also like to experiment with convolutional neural networks[25][26] which can automatically learn useful features for further modelling.

.

References

[1] M. Webster, “Merriam-webster online dictionary,” 2006.

[2] E. Filatova, “Irony and sarcasm: Corpus generation and analysis using crowdsourcing,” in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012*, pp. 392–398, 2012.

[3] R. J. Kreuz and G. M. Caucci, “Lexical influences on the perception of sarcasm,” in *Proceedings of the Workshop on Computational Approaches to Figurative Language*, FigLanguages ’07, (Stroudsburg, PA, USA), pp. 1–4, Association for Computational Linguistics, 2007.

[4] D. Davidov, O. Tsur, and A. Rappoport, “Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon,” in *Proceeding of the 23rd international conference on Computational Linguistics (COLING)*, July 2010.

[5] P. Carvalho, L. Sarmento, M. J. Silva, and E. de Oliveira, “Clues for detecting irony in user-generated contents: Oh...!! it’s ”so easy” ;-),” in *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA ’09, (New York, NY, USA), pp. 53–56, ACM, 2009.

[6] K. Buschmeier, P. Cimiano, and R. Klinger, “An impact analysis of features in a classification approach to irony detection in product reviews,” in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, (Baltimore, Maryland), pp. 42–49, Association for Computational Linguistics, June 2014.

[7] A. A. Vanin, L. A. Freitas, R. Vieira, and M. Bochernitsan, “Some clues on irony detection in tweets,” in *Proceedings of the 22Nd International Conference on World Wide Web*, WWW ’13 Companion, (New York, NY, USA), pp. 635–636, ACM, 2013.

[8] A. Reyes, P. Rosso, and D. Buscaldi, “From humor recognition to irony detection: The figurative language of social media,” *Data Knowl. Eng.*, vol. 74, pp. 1–12, Apr. 2012.

[9] A. Reyes, P. Rosso, and T. Veale, “A multidimensional approach for detecting irony in twitter,” *Language resources and evaluation*, vol. 47, no. 1, pp. 239–268, 2013.

[10] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes, “Semeval-2015 task 11: Sentiment analysis of figurative language in twitter,” in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 470–478, 2015.

[11] J. Tepperman, D. Traum, and S. Narayanan, “”Yeah Right”: Sarcasm Recognition for Spoken Dialogue Systems,” in *Interspeech 2006*, (Pittsburgh, PA), Sept. 2006.

[12] M. Buhrmester, T. Kwang, and S. D. Gosling, “Amazon’s mechanical turk a new source of inexpensive, yet high-quality, data?,” *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[13] A. F. Hayes and K. Krippendorff, “Answering the call for a standard reliability measure for coding data,” *Communication methods and measures*, vol. 1, no. 1, pp. 77–89, 2007.

[14] C. Burfoot and T. Baldwin, “Automatic satire detection: Are you having a laugh?,” in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort ’09, (Stroudsburg, PA, USA), pp. 161–164, Association for Computational Linguistics, 2009.

[15] J. Fürnkranz, “A study using n-gram features for text categorization,” *Austrian Research Institute for Artifical Intelligence*, vol. 3, no. 1998, pp. 1–10, 1998.

[16] S. M. Mohammad and P. D. Turney, “Crowdsourcing a word–emotion association lexicon,” *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

[17] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” in *Proceedings of LREC*, vol. 6, pp. 417–422, Citeseer, 2006.

[18] R. E. Sanders, “Dan sperber and deirdre wilson, relevance: Communication and cognition, oxford: Basil blackwell, 1986. pp. 265.,” *Language in Society*, vol. 17, no. 04, pp. 604–609, 1988.

[19] G. Leech and M. Weisser, “Generic speech act annotation for task-oriented dialogues,” in *Procs. of the 2003 Corpus Linguistics Conference, pp. 441Y446. Centre for Computer Corpus Research on Language Technical Papers, Lancaster University*, 2003.

[20] P. G. Georgiou, O. Lemon, J. Henderson, and J. D. Moore, “Automatic annotation of context and speech acts for dialogue corpora,” *Natural Language Engineering*, vol. 15, no. 3, pp. 315–353, 2009.

[21] S. S. Tekiroğlu, G. Özbal, and C. Strapparava, “Sensicon: An automatically constructed sensorial lexicon,” 2014.

[22] R. Gibbs and H. Colston, *Irony in Language and Thought: A Cognitive Science Reader*. Lawrence Erlbaum Associates, 2007.

[23] T. Eagleton, *Literary theory: An introduction*. U of Minnesota Press, 1996.

[24] T. G. Dietterich, “Ensemble learning,” *The handbook of brain theory and neural networks*, vol. 2, pp. 110–125, 2002.

[25] A. Severyn and A. Moschitti, “Unitn: Training deep convolutional neural network for twitter sentiment classification,” in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pp. 464–469, 2015.

[26] A. Severyn and A. Moschitti, “Twitter sentiment analysis with deep convolutional neural networks,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 959–962, ACM, 2015.