

Implementation of Emotional Features on Satire Detection

Pyae Phyo Thu¹, Than Nwe Aung²

¹ University of Computer Studies, Mandalay,
Patheingyi
Mandalay 1001, Myanmar
pyaephyothu149@gmail.com

² University of Computer Studies, Mandalay,
Patheingyi
Mandalay 1001, Myanmar
mdytina@gmail.com

Abstract

Recognition of satirical language in social multimedia outlets turns out to be a trending research area in computational linguistics. Many researchers have analyzed satirical language from the various point of views: lexically, syntactically, and semantically. However, due to the ironic dimension of emotion embedded in the language, emotional study of satirical language has ever left behind. This paper proposes the emotion-based detection system for satirical figurative language processing. These emotional features are extracted using emotion lexicon: EmoLex and sentiment lexicon: VADER. Ensemble bagging technique is used to tackle the problem of ambiguous nature of emotion. Experiments are carried out on both short text and long text configurations namely news articles, Amazon product reviews, and tweets. Recognition of satirical language can aid in lessening the impact of implicit language in public opinion mining, sentiment analysis, fake news detection and cyberbullying.

Keywords: Satire Detection, Figurative Language Processing, Emotion-Based Features, Ensemble Bagging.

1. Introduction

In order to stimulate individuals to become an active reader, people use implicit attributes while expressing their ideas and feelings. Unlike literal language, figurative language is any figure of speech that uses the word deviate from its actual intentions. This kind of languages can be found extensively in various outlets: literature, television, the internet, social media, comics, and cartoons.

Satire is a form of implicit communication. It can be defined as a literary art which holds up the human vices and follies to ridicule or scorn. People often confuse concepts between satire and comics. Satire differs from comedy in that comedy evokes laughter mainly as an end in itself, while satire mocks. It uses laughter as a weapon and against the vices which exist outside of the work

itself. This goal can be an individual or a type of person, a class, an institution, a nation, or even the entire human race. However, the distinction between comedy and satire is strong only at its extreme. For example, Shakespeare's Falstaff is a comic creation primarily presented for our enjoyment; the Puritan Malvolio in Shakespeare's Twelfth Night is mostly comical but has a satirical aspect directed against the hypocritical and hypocritical type of Puritan. Due to its impact on the meaning of the text, satire becomes one of the most challenging tasks in computational linguistics, natural language processing, and social multimedia sentiment analysis.

In summary, the techniques used in language processing can be categorized as follows: (1) Signature Analysis which focused on exploring specific textual markers or signatures embedded in the language such as

the presence of adjectives, verbs, interjection, and punctuation, see Refs. 1–3, (2) Frequency Analysis which measured the frequency of rare words, ambiguity, POS tag, synonyms, sentiments, and characters, see Refs. 4–6, (3) Contextual Analysis which emphasized on any features of linguistic that contributes to the signaling of contextual presuppositions such as the contrast between author recent tweets and historic tweets, see Refs. 7–9, and (4) Supervised/Semi-supervised Techniques which mainly focused on the use of several classification algorithms in language detection, see Ref. 10. However, thanks to the implicit nature of satirical figurative language, some researchers⁶ argued that it is hard to detect from the emotional point of view.

Yet, satire is a way of criticizing people or ideas in a humorous way. The general hypothesis is that it may be embedded with the negative emotions, humor, irony or exaggeration. However, the major difficulties of recognition satire from emotional outlook is that the emotion embedded in satirical language is ambiguous. The similar emotion can be overlapped in both satirical and non-satirical data. Then the challenging problem is: how to recognize emotion embedded in the satire and how to distinguish satire from such emotions?

Most of the existing efforts in satire detection used base classifiers such as SVM and Naïve Bayes. However, implementation of base classifiers on such ambiguous data does not perform well and it can lead to the misclassification results. Apart from the base classifiers, ensemble approaches use multiple models in their decision making to make up the stability and predictive power of the model. Hence, it's effective on real-world data which depends on the classifier diversity and characteristics of the data in order to accomplish high accuracies and exhibit a low variance. Based on these issues, this work contributed as follows:

- Propose a new system that can differentiate between satirical and non-satirical data from emotional outlook
- Propose the set of emotion-based features for satire detection
- Tackle the consideration of class distribution in data which caused the over-fitting in the multi-figurative language model¹¹
- Analyze the impact of proposed emotion-based features on both base classification and ensemble classification

- A set of experiments to demonstrate that the model can achieve in both short text and long text

This paper is arranged as follows. Section 1 introduces what satire and issues in satire detection are. Section 2 discusses the existing works about satire detection. Section 3 explores how data are collected and preprocessed for further detection. Section 4 explains the detail implementation of emotion-based features implied in the research. Section 5 discusses the experimental results of the proposed emotion-based detection. And then, details analysis of each feature are discussed in section 6 and concluded in section 7.

2. Related Work

Satire has attracted scholarly minds for a long time. Thanks to their efforts, a great deal is known about its origins, functions, motives, forms, plots, and typical devices. Although the phenomenon of literary satire has been investigated extensively, computational approach to satire detection is rarely applied. The model² which attempted to detect satire in English news-wires articles was initially discussed in 2009. This approach was implemented using standard text classification features (i.e., Binary feature weights and Bi-normal separation feature scaling), lexical features (i.e., profanity and slang), and semantic validity which queried the web using the conjunction of defined named entities, expecting that satirical conjunctions were less frequent than the ones from the non-satirical news. The results show that the combination of SVMs with BNS feature scaling achieved high precision and lower recall. Among proposed features, the “validity” feature achieved the best overall F-score.

This approach was expanded in Ref. 12 using various feature extraction strategies and weighting schemes such as BIN, TF-IDF, BNS, and TF-IDF-BNS. The combination of SVMs with TF-IDF-BNS feature scaling achieved high precision and high F-score than the combination with BNS, IDF, and Binary feature weights.

In 2015, the frequency based satire detection model was proposed in order to independent from the lexical information⁶. This model is implemented across multi-lingual languages: English, Spanish and Italian using a set of frequency based features namely frequency of rarest words, ambiguity, POS, synonyms, sentiments, and characters. Due to the lack of lexical and semantic

form, these features succeed over the word-baseline in recognition satirical tweets in multi-languages.

Furthermore, fourteen different linguistic indices have been examined with one-way multivariate analysis of variance (MANOVA) on Amazon product reviews¹³. Results show that these indices of lexical sophistication are the significant predictor for satire detection in product reviews. The model achieves 71% accuracy in leave-one-out-cross-validation (LOOCV). Later, the model which used the power of lexical, sentiment and speech act features with Ensemble Voting classifier w been proposed in Ref. 1. The model outperforms the state of the art by a 6% margin.

Common-sense inferences among texts have also considered in detection¹⁴. The system was designed using a graph-based representation of narrative (Narrative Representation Graph) which capture its participants, their actions, and utterances. Based on this NRG, the model make a set of inferences and mapping to general categories abstraction. Then the system makes a prediction by reasoning over this abstract NRG, decomposing it into paths, and later associate the paths to decision output. The model outperforms the convolutional neural network and state of art model, but the narrative representation is limited and can't capture the more nuanced information.

Unigrams, semantic, psycholinguistic, and statistical features are employed together with ensemble feature selection methods to detect satiric, sarcastic and ironic content in news and customer reviews¹⁵. LIBSVM, Logistic Regression, Naïve Bayes, Bayesian Network, Multilayer Perceptron, C4.5, Classification and Regression Trees (CART) are implied over two satiric and one ironic dataset. Among these classifiers, logistic regression and random forest yield the best classification results for any ensemble feature set. They explore that satire and irony share some common psycholinguistic characteristics like correctness, meaningfulness, imageability, and familiarity and strengthen a belief that satire is a sub-type of iron through several sets of experiments.

Some researchers mention that satirical cues are often reflected in certain paragraphs rather than the whole document¹⁶. Hence, a set of paragraph-level linguistic features are used to unveil the satire by incorporating with neural network and attention mechanism. Results prove that analyzing the data with paragraph-level linguistic features is higher than the document-level

linguistic features. In this way, almost all of the analytic works proposed the best of the best indices for satire analysis. However, due to the ironic dimension of satirical figurative language, no prior work has been done on emotional outlook. Some researchers⁶ point out that recognition of satire from emotion will be hard due to their ironic dimension.

Moreover, most of the existing efforts till now used base classifiers such as SVM and Naïve Bayes in language detection. However, implementation of base classifiers on such ambiguous data will not perform well and it can lead to the misclassification results. In recent years, ensemble methods become popular because of its stability and predictive power in detection. Many research works in various areas were considered ensembling in their tasks. See Refs. 10, and 17-18 for more details. It is effective on real-world data which depends on the classifier diversity and characteristics of the data in order to accomplish high accuracies and exhibit a low variance. There are various kinds of ensemble methods: Voting, Bagging and Boosting. Their major goal is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability/robustness over a single estimator. In this work, ensemble bagging is employed together with emotion-based features in order to tackle the problem of ambiguous emotion.

3. Dataset Collection and Text Preprocessing

One of the purposes of this study is to examine whether the model can implement in both long text and short text. Hence, three kinds of dataset namely: Amazon Product Review dataset, News-articles dataset and Twitter datasets are used.

Amazon Product Review dataset was obtained from the publicly available corpus¹⁹ which contained 1,000 satirical product reviews and 1,000 serious product reviews. These reviews were collected from four most sarcastic and most serious reviews products appeared on December 9, 2015 which contained about 1,000 words. After that, it was formatted in plain text.

News-articles dataset was obtained from the publicly available corpus²⁰ which have been accumulated using various satirical textual features. These articles were collected by scraping news websites (Reuters, CNET, CBS News) and various satirical news websites (Daily Currant, Daily Mash, Dandy Goat, Empire News, The

Lapine, Lightly Braised Turnip, Mouthfrog, News Biscuit, News Thump, Satire Wire, National Report and The Spoof) ranging from 2013 to 2015. Each article contained at most 2,000 words and formatted in plain text.

In order to accumulate twitter data, an easy to use python library for accessing Twitter API: tweepy²¹ was implied. These tweets were collected from 4 true news and satirical accounts as in Ref. 6. Firstly, about 3,000 tweets were collected using the keywords “@MailOnline”, “@NewsBiscuit”, “@TheDailyMash” and “@TheTimes”. In total, about 12,000 tweets were accumulated for non-satire and satire corpus.

Furthermore, the tweets were collected using user-defined hashtag “#satire”. It’s another way of accumulating twitter data because the author of the tweet self-identified what their tweets are talking. Hence, it’s the most reliable way to accumulate the twitter. As a result, about 8,000 tweets were collected using this user-defined hashtag. The details of the dataset status used in this study can be seen as shown in Table 1.

Table 1. Corpus statistics

Dataset	Satire	Non Satire	Total
Product Reviews	1,000	1,000	2,000
News Articles	1,706	1,705	3,411
Twitter (Hashtag-Based)	4,300	4,300	8,600
Twitter (Account-Based)	5,980	6,000	11,980

Table 2. List of Twitter newspaper and satirical news accounts

Satirical	Non-Satirical
News Biscuit	The Daily Mail
The Daily Mash	The Times

4. Features Extraction

The system is implemented with six classes of features: (1) Word-Based, (2) Emotion, (3) Sentiment, (4) SenticNet, (5) Bag-of Sorted Emotion (BOSE) and (6) its TFIDF (BOSE_TFIDF). The former used as text classification baseline and the latter used as proposed emotional features in the system. The details illustration of this emotion-based satire detection system can be seen as shown in Fig. 1. Firstly, basic emotional features and sentimental features are computed using sentiment

analysis and social cognition engine²² (SÉANCE). It is a freely available text analysis tool for sentiment, emotion, social cognition, and social order analysis which contained built-in databases: GALC, EmoLex, ANEW, SENTIC, VADER, Hu-Liu, GI, and Lasswell. Based on these emotional and sentimental features, bag-of sorted emotion (BOSE) features are extracted. And then, emoBigram and emoTrigram are extracted in order to apply unsupervised term weighting (TFIDF). The details of each feature are described in the following subsections.

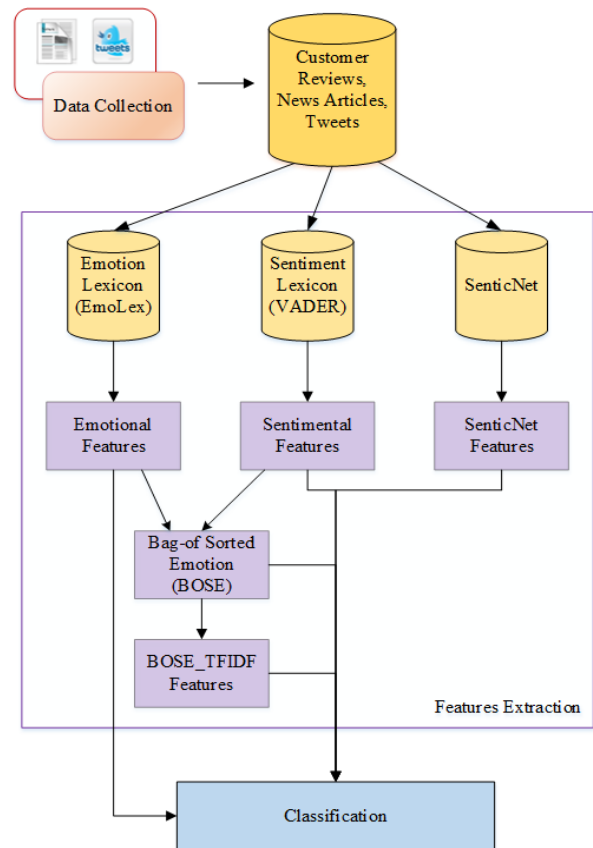


Fig. 1. Architecture of the emotion-based satire detection system.

4.1. Word-Based Features

The highest 1,000 most frequent occurrences of word n-gram model: bigram BoW and trigram BoW are used as the text classification benchmark.

4.2. Emotional Features

Eight basic emotions of emotion lexicon: EmoLex namely anger, anticipation, disgust, fear, joy, sadness,

surprise, and trust are used as the basic emotional features. These emotional features are extracted using sentiment analysis and social cognition engine (SÉANCE).

4.3. Sentimental Features

Sentiment polarity of SÉANCE's VADER: positive, negative, neutral and compound sentiment are used as the basic sentimental features.

4.4. SenticNet Features

In order to assist the performance of the system, four emotional dimensions and polarity of emotion lexicon: SenticNet are computed using SÉANCE. SenticNet is a database extension of WordNet consisting of norms for 13,000 words with regard to four emotional dimensions (sensitivity, aptitude, attention, and pleasantness) and polarity.

4.5. Bag-of Sorted Emotion (BOSE) Features

This feature is assembled from eight basic emotional features and three sentimental features. Firstly, emotional features and sentiment features are sorted out individually as in Algorithm 1 and Algorithm 2. Later, the corresponding index of each sorted features are extracted, and then, concatenate as shown in Fig. 2.

Algorithm 1: Sorting Emotion

Input Basic Emotions (BE) computed with SÉANCE's EmoLex

Output Sorted Emotion (SE)

```

1: function sortedEmotion(emotion):
2:   for i in len(BE) do
3:     srtEmoScore[i] = sort(BE[i])
4:     srtEmoIdx[i] = extract corresponding
                       index from
                       srtEmoScore[i]
5:     srtEmotion[i] = convert srtEmoIdx[i]
                           index into
                           corresponding
                           emotion label
6:   end for
7:   return srtEmotion
8: end function

```

Algorithm 2: Sorting Sentiment

Input Basic Sentiment (BS) computed with SÉANCE's VADER

Output Sorted Sentiment (SS)

```

1: function sortedSentiment(sentiment):
2:   for i in len(BS) do
3:     srtSentiScore[i] = sort(BS[i])
4:     srtSentiIdx[i] = extract
                       corresponding
                       index from
                       srtSentiScore[i]
5:     srtSentiment[i] = convert
                           srtSentiIdx[i]
                           index into
                           corresponding
                           sentiment label
6:   end for
7:   return srtSentiment
8: end function

```

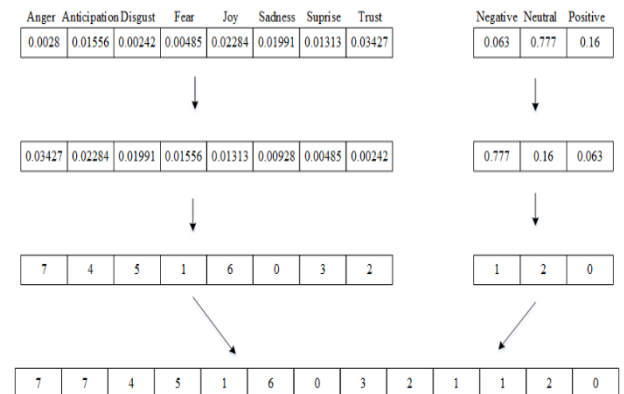


Fig. 2. Structure of bag-of sorted emotion (BOSE).

4.6. BOSE-TFIDF Features

In order to achieve BOSE-TFIDF, emotional and sentimental features are first sorted out using Algorithm 1 and Algorithm 2. Based on this sorted information (BOSE), bigram and trigram emotion are extracted as described in Algorithm 3. Later, term frequency-inverse document frequency (TFIDF) weighting scheme: Eq. (1) is computed on each emoNgram. The tfidf weight is an unsupervised weighting which composed of two parts: the first computes the normalized Term Frequency (TF) which is the number of times the term t appears in

document d , divided by the total number of terms in that document; the second term is the Inverse Document Frequency (IDF) which is computed as the logarithm of the number of the documents in the corpus D divided by the number of documents where the specific term appears.

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \quad (1)$$

Algorithm 3: EmoNgram Computation

Input Sorted Emotions (SEC), Sorted Sentiments (SSC), ngram

Output EmoNgram (ENC)

```

1:  function emoNgram(SEC, SSC,
    ngram):

2:      for  $i$  in  $\text{len}(\text{SEC})$  do

3:          emotion = tokenize(SEC[i])
4:          sentiment = tokenize(SSC[i])

5:          if ngram = "bigram" then
6:              emoNgram[i] = concatenate
                           the first two emotions
7:          end if

8:          if ngram = "trigram" then
9:              emoNgram[i] = concatenate
                           the first three emotions
10:         end if

11:         if ngram = "biSenti" then
12:             emoNgram[i] = concatenate
                           the first two emotions,
                           and the first sentiment
13:         end if

14:         if ngram = "triSenti" then
15:             emoNgram[i] = concatenate
                           the first three emotions,
                           and the first sentiment
16:         end if

17:     end for

18:     return emoNgram

19: end function

```

5. Experimental Results

In order to verify the effectiveness of the system, we evaluated it through a classification task. The standard classifier SVM is employed as the benchmark classifier in order to compare with the Random Forest classifier. It is one of the ensemble bagging methods for classification and regression that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Experiments are performed in both short text and long text using standard 10-fold cross-validation. The system has run upon Windows 10 (64-bit) PC with Intel Core i3 processor, 8GB RAM, 1GB graphic card and 500 GB hard disk.

Table 3. Classification results of each feature on product reviews dataset.

Features	SVM	Random Forest
BoW	0.474	0.815
Emotion	0.601	0.624
Sentiment	0.608	0.627
SenticNet	0.530	0.587
BOSE	0.474	0.596
BOSE_TFIDF	0.474	0.604
All Emotion	0.474	0.719
All Features	0.474	0.767

Table 3. shows the F measure of each feature on product reviews dataset. In all cases, random forest outperforms the SVM classifier. Meanwhile, the SVM does not perform well as in the random forest and it even gives the same results 0.474 in BOW, BOSE, BOSE_TFIDF, all emotional features, and all features. This is due to the ambiguous nature of satire dataset and its emotion.

Table 4. Distribution of emotion in each class.

Emotion	Satire	Non-Satire
Trust	1,046	1,153
Anticipation	268	273
Joy	42	92
Fear	220	118
Anger	61	31
Sadness	49	32
Surprise	11	4
Disgust	9	2

In order to visualize the fact, the number of each emotion in product reviews dataset was analyzed as

shown in Table 4. It shows that almost all of the emotion is involved in both classes equally. It is hard to distinguish satire from single emotion. Distinguishing such ambiguous data with SVM is not adequate. It causes bias error in detection. Meanwhile, ensemble bagging classifier: random forest can tackle this problem of ambiguous and ironic nature of satire and emotion. Analyzing the data with all emotional features alone achieve 0.719 F measure. When it combines with BoW, the performance of the system arise to 0.767.

Table 5. Classification results of each feature on news articles dataset.

Features	SVM	Random Forest
BoW	0.475	0.834
Emotion	0.620	0.669
Sentiment	0.581	0.622
SenticNet	0.498	0.561
BOSE	0.475	0.563
BOSE_TFIDF	0.476	0.548
All Emotion	0.475	0.703
All Features	0.475	0.840

Analysis results of each feature on news articles dataset can be seen as shown in Table 5. Unlike the reviews dataset, the combination of all features reaches 0.84. It outperforms the baseline: BoW which achieves only 0.834. Still, the standard classifier: SVM does not perform well as in product reviews dataset.

Table 6. Classification results of each feature on Twitter (Hashtag-Based) dataset.

Features	SVM	Random Forest
BoW	0.490	0.711
Emotion	0.564	0.639
Sentiment	0.561	0.633
SenticNet	0.528	0.699
BOSE	0.490	0.610
BOSE_TFIDF	0.490	0.585
All Emotion	0.490	0.737
All Features	0.490	0.777

Table 6. shows the results of each feature on short text hashtag-based twitter dataset. Because it is in short text configuration, BoW feature does not perform well as in long text datasets. It declines to 0.711 in random forest classification. Meanwhile, the proposed emotional features offer 0.737. The combination of both features gives 0.777 scores. Like in the previous long text datasets, SVM classifier does not perform well with

proposed emotional features. It is still ambiguous in classification.

Table 7. Classification results of each feature on Twitter (Account-Based) dataset.

Features	SVM	Random Forest
BoW	0.489	0.648
Emotion	0.560	0.727
Sentiment	0.537	0.681
SenticNet	0.512	0.606
BOSE	0.496	0.602
BOSE_TFIDF	0.498	0.607
All Emotion	0.491	0.743
All Features	0.491	0.758

In account-based twitter dataset, BoW significantly declines to 0.648 scores. Meanwhile, the proposed features give 0.743 scores. The combination of both offers 0.758. In summary, the proposed features perform more than 0.70 in both short text and long text. While it combines with BoW features it gives at most 0.840. Due to the ambiguous nature of satire and its emotion, SVM does not perform well as in the random forest. In conclusion, ensemble bagging classifier: the random forest is the most suitable one for emotion-based satire detection.

6. Features Analysis

This section is evaluated with two underlying goals: (a) model consistency and (b) relevance of the features. The first one is evaluated using bias-variance tradeoff and the latter is evaluated using tree-based feature selection.

6.1. Bias-Variance Tradeoff

Firstly, the consistency of the proposed system is evaluated using bias-variance tradeoff. It is an important concept when it comes to choosing a machine learning algorithm. Bias refers to the tendency to constantly learn the same wrong thing. Variance refers to the tendency to output random things irrespective of the input data. This is visualized using learning curve which is an increase of learning (vertical axis) with experience (horizontal axis).

Fig. 3 illustrates the learning curve of BoW feature on long text: news articles dataset using SVM classifier. In this figure, both training-score and validation score ends up similar to each other as the number of training examples increased. It means that the system is suffered

from high bias error. Hence, this type of classification is not suitable for emotion-based satire detection.

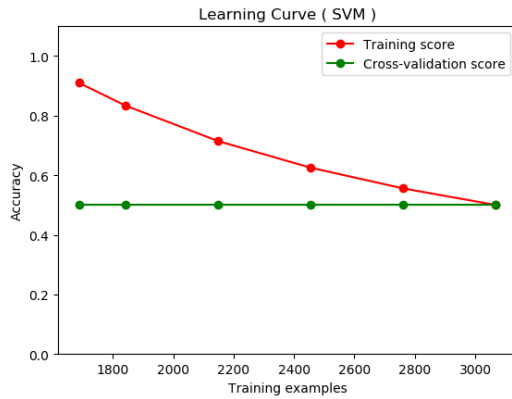


Fig. 3. Learning curve of text classification baseline (BoW) on long text (SVM classification).

Meanwhile, Fig. 4 illustrates the learning curve of this BoW feature with random forest. In this case, the system is less suffer from high bias error than in SVM. The major cause of this high bias error is that the features used are not adequate to solve the problem. Hence, classifying such ironic nature of satirical data with SVM classifier is not consistent with satire detection.

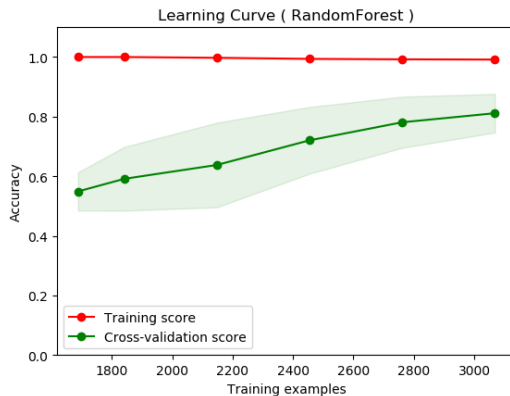


Fig. 4. Learning curve of text classification baseline (BoW) on long text (random forest classification).

Likewise, the learning curve of the proposed features with SVM suffers from high bias error. See Fig. 5. However, classification of the proposed features with random forest suffers less bias than the BoW features. This shows that even though the classification results of the BoW features outperform the proposed features, it suffers from high bias error. Meanwhile, the proposed features less suffer than that of BoW features. Hence, it works better than the BoW features.

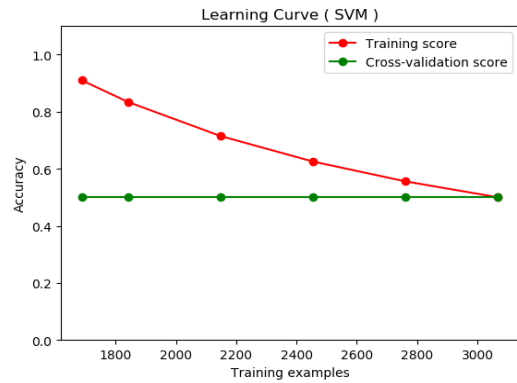


Fig. 5. Learning curve of proposed features on long text (SVM classification).



Fig. 6. Learning curve of proposed features on long text (random forest classification).

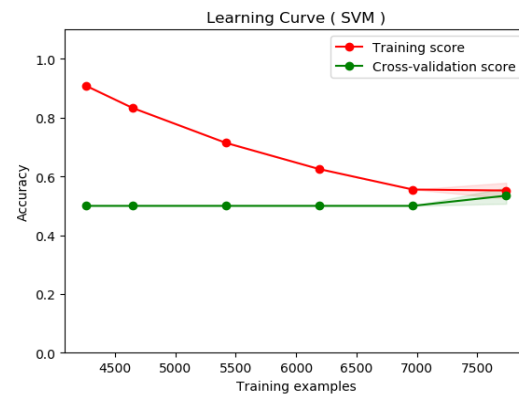


Fig. 7. Learning curve of text classification baseline (BoW) on short text (SVM classification)

Fig. 7 and Fig. 8 shows the learning curve of BoW features on short text dataset. In both classifications, the training score and the validation score end up similar. Hence, BoW features significantly suffer from high bias error in short text data.

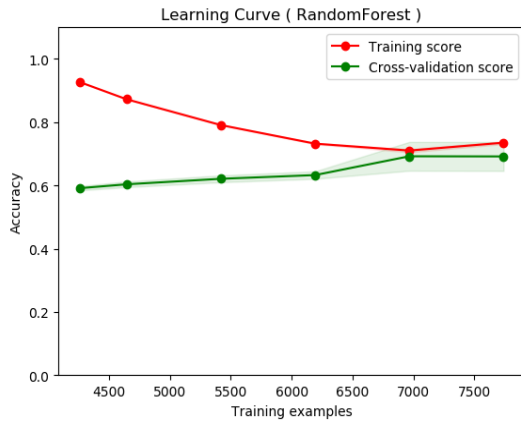


Fig. 8. Learning curve of text classification baseline (BoW) on short text (random forest classification).

Meanwhile, the curves in Fig. 9 and Fig. 10 show the learning curves of proposed features on short text data. It shows that the proposed features suffer high bias only in SVM classification. It does not suffer high bias in random forest classification.

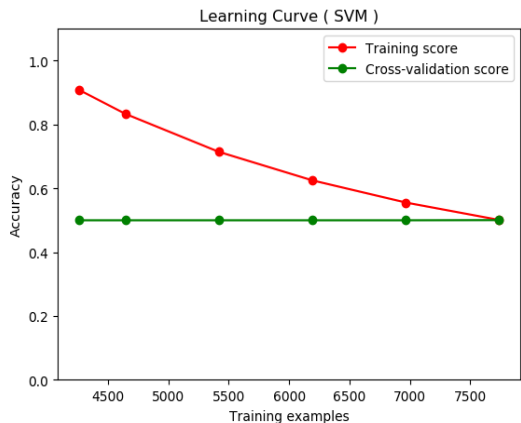


Fig. 9. Learning curve of proposed features on short text (SVM classification).

Hence, it shows that even though the classification results of the text classification baseline BoW offers higher results, it does not actually tackle the problem of ambiguous nature of satire and its emotion. Only the proposed features with random forest classification performs better and it does not suffer any kind of bias-variance error in both short text and long text configurations. This proves that proposed emotion-based system is consistent with satire detection. Random forest classifier is the most suitable one for emotion-based detection.

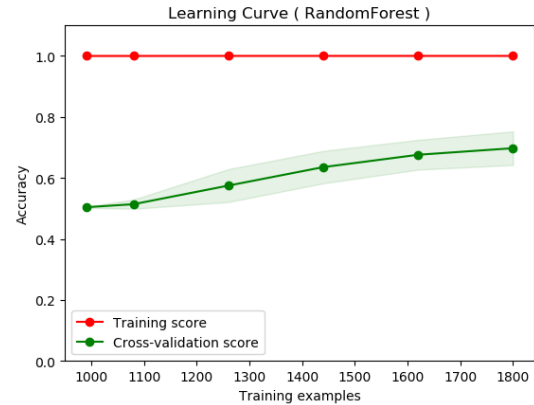


Fig. 10. Learning curve of proposed features on short text (random forest classification).

6.2. Feature Ranking

Finally, in order to have a clear understanding of the contribution of each feature in satire detection, the importance of each feature has been studied using tree-based feature selection. It uses tree-based estimators to compute the importance of features.

Table 8. The 5 most discriminating features regarding tree-based feature selection results.

Dataset	Features
Product Reviews	Senti_compound (0.0418)
	Senti_positive (0.0340)
	Senti_neutral (0.0238)
	SN_apitute (0.0205)
	SN_pleasantness (0.0200)
News Articles	BoW_story continued (0.0227)
	Senti_negative (0.0222)
	Senti_compound (0.0194)
	BoW_said statement (0.0152)
	Senti_neutral (0.0146)
Twitter (Hashtag)	SN_sensitivity (0.0682)
	SN_attention (0.0641)
	SN_apitute (0.0633)
	SN_pleasantness (0.0609)
	Senti_compound (0.0432)
Twitter (Account)	SN_pleasantness (0.0627)
	SN_apitute (0.0618)
	SN_attention (0.0611)
	SN_sensitivity (0.0588)
	Senti_neutral (0.0454)

As shown in Table 8, the most promising one in almost all datasets are sentimental features and SenticNet features. BoW features become higher only in long text news articles dataset. It proves that the proposed

emotional features are more relevant for satire detection than that of BoW features.

7. Conclusion

Due to the widespread usage of satirical figurative language, handling satire becomes one of the most challenging tasks in computational linguistics. This system analyzes both long text and short text data to determine whether it is satire or not using emotional features. Even though ironic dimension of the language lead difficulty in satire detection from emotion, a set of emotion-based features are proposed by using Emotion Lexicon: EmoLex and Sentiment Lexicon: VADER. It shows that only the ensemble bagging classifier can tackle the problem of ambiguous emotion. Otherwise, the system will suffer from high bias error. Moreover, the proposed features offer the promising result which is greater than 0.70. In the best case, it achieves at most 0.84. However, the system only considers the emotion of the major text. It does not consider the contextual emotion between current and previous text. In future, we will plan to explore the contextual emotion between current and historic text.

References

1. AN. Reganti, T. Maheshwari, U. Kumar, A. Das and R. Bajpai. *Modeling Satire in English Text for Automatic Detection*. In Data Mining Workshops ICDMW (2016), pp. 970-977.
2. C. Burfoot and T. Baldwin. *Automatic satire detection: Are you having a laugh?* In Proceedings of the ACL-IJCNLP 2009 conference short papers (2009 Aug 4), pp. 161-164.
3. RJ. Kreuz and GM. Caucci. *Lexical influences on the perception of sarcasm*. In Proceedings of the Workshop on computational approaches to Figurative Language (2007 Apr 26), pp. 1-4.
4. F. Barbieri and H. Saggion. *Automatic detection of irony and humour in twitter*. In Proceedings of the International Conference on Computational Creativity (2014), pp. 155-162.
5. F. Barbieri, H. Saggion and F. Ronzano. *Modelling sarcasm in twitter, a novel approach*. In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (2014), pp. 50-58.
6. F. Barbieri, F. Ronzano and H. Saggion. *Do We Criticise (and Laugh) in the Same Way? Automatic Detection of Multi-Lingual Satirical News in Twitter*. In IJCAI (2015 Jun 23), pp. 1215-1221.
7. D. Bamman and NA. Smith. *Contextualized Sarcasm Detection on Twitter*. In ICWSM (2015 May), pp. 574-577.
8. A. Joshi, V. Sharma, and P. Bhattacharyya. *Harnessing Context Incongruity for Sarcasm Detection*. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers 2015), pp. 757-762.
9. A. Khattri, A. Joshi, P. Bhattacharyya and MJ. Carman. *Your Sentiment Precedes You: Using an author's historical tweets to predict sarcasm*. In 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (2015), pp. 25-30.
10. E. Fersini, FA. Pozzi and E. Messina. *Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers*. In Data Science and Advanced Analytics (2015), pp. 1-8.
11. PP. Thu and N. Nwe. *Impact analysis of emotion in figurative language*. In IEEE/ACIS 16th International Conference (2017 May 24), pp. 209-214.
12. T. Ahmad, H. Akhtar, A. Chopra and MW. Akhtar. *Satire detection from web documents using machine learning methods*. In Soft Computing and Machine Intelligence (2014), pp. 102-105.
13. S. Skalicky and S. Crossley. *A statistical analysis of satirical Amazon. com product reviews*. The European Journal of Humour Research 2, no. 3 (2015), pp. 66-85.
14. D. Goldwasser and X. Zhang. *Understanding Satirical Articles Using Common-Sense*. Transactions of the Association of Computational Linguistics 4, no.1 (2016), pp. 537-49.
15. K. Ravi and V. Ravi. *A novel automatic satire and irony detection using ensembled feature selection and data mining*. Knowledge-Based Systems (2017), pp. 15-33.
16. F. Yang, A. Mukherjee and E. Dragut. *Satirical News Detection and Analysis using Attention Mechanism and Linguistic Features*. preprint arXiv:1709.01189. (2017 Sep 4).
17. S. Singh. *An Ensemble Approach for Cyber Attack Detection System: A Generic Framework*. International Journal of Networked and Distributed Computing, Vol. 2, No. 2 (April 2014), pp.78-90
18. H. Nomiya, S. Sakaue and T. Hochin. *Recognition and Intensity Estimation of Facial Expression Using Ensemble Classifiers*. International Journal of Networked and Distributed Computing, Vol. 4, No. 4 (October 2016), pp. 203-211
19. https://rpubs.com/pm0kjp/satire_serious_reviews
20. A. Frain and S. Wubben. *SatiricLR: a Language Resource of Satirical News Articles*. In LREC (2016).
21. <http://www.tweepy.org>
22. SA. Crossley, K. Kyle and DS. McNamara. *Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis*. Behavior research methods (49, no. 3 2017), pp. 803-821.