Optimal Feature Extraction based Machine Learning Approach for Sarcasm Type Detection in News Headlines

Vaishvi Prayag Jariwala
Delhi Publich School
Surat, Gujarat, India

ABSTRACT
Sarcasm detection has received increasing research in recent years. Detection of sarcasm is of great importance and beneficial to many NLP applications, such as sentiment analysis, opinion mining and advertising. Detection of sarcasm is of great importance and beneficial to many NLP applications, such as sentiment analysis, opinion mining and advertising. Generally, sarcasm detection task is treated as standard test classification problem. Sarcasm is the unconventional way of conveying a message which conflicts the context. It can lead to a state of ambiguity. As Sarcasm represents contrary sentiment to the literal meaning that is conveyed in the text, it is hard to identify sarcasm even for a human. Existing models mainly focus on designing effective features for improving the detection performance. In this paper, optimal features are to be selected before data passes to classification task. So data pre-processing makes the data clean so that the performance of the classifier will be enhance. Result shows the improve performance in sarcasm detection using the optimal feature sets.

Keywords
Irony, Satire, Sentiment Analysis, Sarcasm detection, SVM

1. INTRODUCTION
Social media platforms, like Twitter, have gradually now become one of the most exciting platforms for users to say their ideas and opinions on miscellaneous events, products etc.. So many companies have a eager attraction towards this data, particularly to analyze the thoughts and opinions of people concerning various kinds like political events, social events, movies, songs, product reviews etc. Many of the times when people want to buy online, that time they will conduct small survey by analyzing all the reviews and comments available online. Sentiment analysis is the opinion of the user for the particular things. The process of automating identification of sentiment in text is referred as sentiment analysis. Sentiment analysis (occasionally also termed as Opinion extraction or Opinion Mining or Subjectivity analysis) is the study of individual or groups' emotions, opinions and attitudes towards different entities like services, products, organizations, individuals, issues, events, topics, or towards their attributes. Normally, there are two ways to state sentiment analysis: 1) Implicit sentiment 2) Explicit sentiments. In implicit sentiment, a sentence which entails any opinion, that sentence depicts the existence of implicit statements. In explicit sentiment, a sentence that shows the opinion directly indicates the existence of explicit sentiment. The sentiments expressed by public are in the appearance of positive, negative or in the form of neutral polarity. Sentiment analysis is measured as a classification task, which classifies text or opinion into positive, negative or neutral polarity. Many characters influences sentiment analysis process in social media websites namely: 1) use of slang words, 2) characters limits in blog, 3) use of not-literal language, such as irony and many more. Sarcasm is often described as ironic content which is used to mock, amuse or insult. The process of recognize and classification of sarcastic contents known as sarcasm classification or detection.

The Merriam-Webster dictionary defines sarcasm as a sharp and often satirical or ironic utterance designed to cut or give pain, is not usual in online communities such as social media and e-commerce platforms. The Free dictionary defines sarcasm as a form of verbal irony that is intended to express contempt or ridicule. Sarcasm is a special type of sentiment which plays a role as an interfering factor that can change the polarity of the given text. Sarcasm detection is one of the most difficult tasks in natural language processing. The average human reader will have difficulty in recognition of sarcasm in twitter data, product review, blogs, online discussion forum, etc. The main goal of the sarcasm detection problem is to find out whether the sentences within the text are sarcastic or not. Sarcasm detection is a mostly difficult task, even for humans.

2. RELATED WORK
The ratio of emotional words is computed in [1]. Many researchers have classified words as positive, negative and neutral. The intensity of words and provided them a rating of 1-5 where 1 represents less positive or less negative and 5 represents more positive or more negative [2], [3]. Emoticons can also imitate the nature of status. The basic perception is that the orientation of sentiment words and emoticons are same when they occur in general. For orientation classification, there are two methods namely corpora and dictionary-based methods.. Conjunctions are used to join the sentences when positive conjunctions like and is used when they give rise to a positive orientation if negative conjunctions like but are used then it gives rise to opposite orientation. A Network is constructed with synonyms in wordnet [2].

Sarcasm is a complex linguistic observable fact that has long enthralled both linguists and NLP researchers. After all, a better computational understanding of this difficult speech act could potentially bring about many benefits to accessible opinion mining applications. Across the rich history of research on sarcasm, several theories such as the Situational Disparity Theory [4] and the Negation Theory [5] have appeared. In these theories, a common theme is a design that is strongly grounded in contrast, whether in sentiment, intention, situation or context. The author [6] propagates this idea forward, presenting an algorithm strongly based on the intuition that sarcasm arises from a concurrence of positive and negative situations.

The author [7] shows how linguistic style and contextual features plays a vital role in processing irony. He identifies

irony on the basis of 3types of patterns like: Opposition, Rhetorical question, and Circumlocution.

The author [8] proposed a methodology to recognize the sarcasm on twitter using Simple Vector Machine (SVM), Maximum Entropy algorithms. Initially, they collected the data and created into two datasets that are before adding the sarcastic tweets to the training data and after adding sarcastic tweets to the training data. POS tagging was performed using Penn tree bank to tag each word with the associated part of speech. The authors extracted features related to sentiment, punctuation, syntactic, and pattern etc from the training data. After extracting features, classification is done by using SVM, and Maximum Entropy algorithms. Compared to both the algorithms Maximum Entropy gives more accuracy when compared to the SVM algorithm.

The author [9] proposed a new method for sarcasm detection as NLP and Corpus-based method. The aim was to identify the intention to use the sarcastic statement in the tweets by individuals. The authors collected the tweets from the Twitter website and NLP techniques like tokenization, parts of speech (PoS), and lemmatization are performed. NLP methods on tweets are applied to fetch action words. Once the action words are found from the tweets, these are matched with the corpus of sarcasm data using semantic matching and graph-based matching which gives a score of sarcasm for the given tweet. By this score, the level of sarcasm in the given tweet is detected.

The author [10] focuses on humor and irony processing. They compare humor and irony with different genres like politics, and technology. They use different features for the identification. These consist of ambiguity, polarity, unexpectedness and emotional scenarios. Ambiguity is a combination of structural, morphosyntactic and semantic layers. Structural ambiguity can be viewed as funny situations which occur most in the text containing humor.

Although only very limited work has been done on using neural networks for sarcasm detection, neural models have seen increasing applications in sentiment analysis, which is a closely-related task. Different neural network architectures have been applied for sentiment analysis, including recursive auto-encoders [11], dynamic pooling networks [12], deep belief networks [13], deep convolution networks [14] and neural CRF [15]. This line of work gives highly competitive results, demonstrating large potentials for neural networks on sentiment analysis. One important reason is the power of neural networks in automatic feature induction, which can potentially discover subtle semantic patterns that are difficult to capture by using manual features. Sarcasm detection can benefit from such induction, and several work has already attempted for it [16] [17].

In the paper [18], it compared the performance with various language-independent features and preprocessing methods for classifying text as sarcastic and non-sarcastic. The comparison was done over three Twitter dataset in two different languages, two of these in English with a balanced and an imbalanced distribution and the third one in Czech. The feature set included n-grams, word-shape patterns, pointedness and punctuation-based features.

3. APPROACH

This paper aims to explore the support vector machine (SVM) models for News sarcasm detection. The proposed work flow in sarcastic text detection is shown in figure 1.

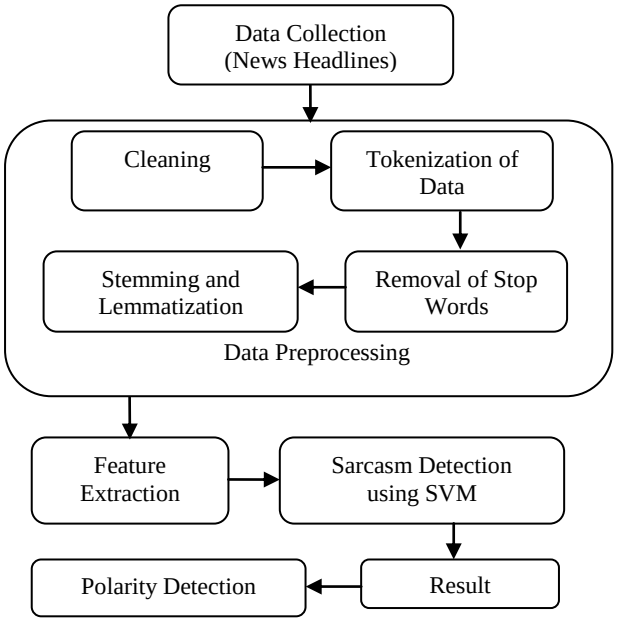


Fig. 1: Proposed approach for Sarcasm detection using SVM

3.1 Data Collection

Data is collected from the kaggle website for the News headline. This dataset was collected from The Onion and HuffPost. It contains sarcastic and regular news headlines. The sarcastic news headlines were gathered from The Onion and normal headlines from HuffPost. As it is not written by the general population, the chances of spelling mistakes and informal usage are low. It contains 27K of headlines, as of it 11.7K are sarcastic and 14.9K are non-sarcastic. The dataset consists of three attributes, is Sarcastic, Headline and link. The Sarcastic points out whether the instance of headline is sarcastic or not, Headline attribute contains the headline of the article and link attribute contains the line of the news article. The perceive sarcastic data is greater.

3.2 Data Preprocessing

Data preprocessing of the dataset is mainly done in three steps.

3.2.1 Tokenization of data

Tokenization is the process by which big quantity of text is divided into smaller parts called tokens**.** In this process, after the data is retrieved from the dataset. The dataset in which every data which is taken is in the form of sentences and phrases. Now these sentences and phrases are tokenized into words, so that is easily understandable**.** These tokens are very useful for finding such patterns as well as are considered as a base step for stemming and lemmatization.

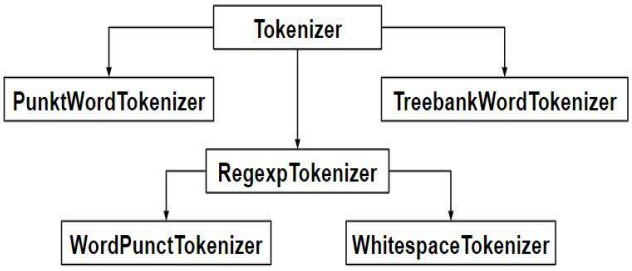


Fig.2: Tokenization process

3.2.2 Removal of stop words

Stop words are the most common words in any natural language. Stop words are the words that the search engine has programmed to ignore when both indexing and retrieving of entries. Articles are the best examples of stop words. For tasks like text classification, where the text is to be classified into several categories, stop words are removed from the given text so that more aim can be given to those words which describe the denotation of the text. Stop word removal has several advantages like: On removing stop words, dataset size reduces and the running time to train the model also reduces, removing stop words can potentially help out to enhance the performance as there are less and only significant tokens left. Thus, it could boost classification accuracy; even search engines like Google remove stop words for speedy and relevant retrieval of data from the database.

3.2.3 Stemming and Lemmatization

Stemming and lemmatization is the process of converting the words into their root words so that they can be examined as a single thing. Stemming and Lemmatization are Text Normalization (or sometimes called Word Normalization) techniques in the field of Natural Language Processing that are used to prepare text, words, and documents for further processing.

Stemming is certainly the simpler of the two methods. With stemming, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root; it just is an equal to or smaller form of the word. Stemming algorithms are typically rule-based. For example, we may have a suffix rule that, based on a list of known suffixes, cuts them off. In the English language, we have suffixes like "-ed" and "-ing" which may be useful to cut off in order to map the words "book," "booking," and "booked" all to the same stem of "book." lemmatization is a more calculated process. It involves resolving words to their dictionary form. Lemmatization usually refers to the morphological analysis of words, which aims to remove inflectional endings. Lemmatization does not simply chop off inflections, but instead relies on a lexical knowledge base like WordNet to obtain the correct base forms of words.

3.3 Feature Extraction

Feature extraction has a vast task in determining the outcome of any machine learning job. The quality of classification, both qualitatively and quantitatively, depends on the features selected. In this paper, it focuses on extracting the features from news headlines that can be categorized into various types, namely, lexical, hyperbolic, pragmatic, sentiment, and contradiction. Lexical features include n-gram, bigram, and unigram which are combination of words that are extracted from the news headlines to aid in tokenization. Intensifiers are also identified as they might help in the sarcasm detection process. The proposed system extracts a total of 17 features: noun and verb count, positive intensifier, negative intensifier, bigram, trigram, skip gram, unigram, sentiment score, interjections, punctuators, exclamations, question mark, uppercase, repeat words count, positive word frequency, negative word frequency.

Various sentiment-based features are extracted from the news headlines like positive words frequency—total number of positive words; negative words frequency—total number of negative words; positive intensifiers—negative intensifiers—n grams—set of consecutively occurring "n" words(n = 1(unigram); n = 2(bigram), etc.); and skip gram—n grams with an additional factor called skip distance; passive aggressive count gives the indirect expression of hostile intention; sentiment score gives the sentiment value in which a "−1" indicates negative sentiment and a "+1" indicates a positive sentiment; the co-occurring terms need not be consecutive as the tokens can be skipped based on the skip distance value; noun and verb counts can be obtained from POS tagging of a news headline; POS Tagging is a way to tag each word present in the news headline with its appropriate parts of headline; exclamations and question marks are most meaningful among the various punctuators for detecting sarcasm. Uppercase words are extracted as features because sometimes people use capital lettered words to stress on the things that they want to convey strongly. These are the prominent set of features which will be useful for sarcasm detection. Once these features are extracted, a numerical value for the features is obtained. These extracted features are categorized into different groups such as linguistic, sentiment based, and contradiction based feature sets. Once the features are extracted, using the rule based approach, where the optimal set of features for detecting sarcasm will be identified.

3.4 Sarcasm detection using the support vector machine (SVM)

A machine learning approach based on the feature vectors generated from the news headlines dataset was used to train a classifier. The classification algorithm used is support vector machine (SVM) due to its simplicity and effectiveness in binary classification. In this paper, it uses the linear kernel to perform the classification task. Each data item is plotted as a point in n-dimensional area, where 'n' is number of features, with the value of every feature being the value of a specific coordinate. Support Vector Machines can resolve the problem of dealing with nonlinear boundaries by transforming the instance space using a nonlinear transformation, a process referred to as the kernel mapping technique. In the transformed space, a linear model can be constructed that corresponds to a nonlinear model in the original instance space. Al-though this transformation would appear to introduce a great number of new dimensions to the problem, SVM functions only depend on the training and test data through the kernel mapping, which is typically far less complex. Even in the nonlinear case they are therefore able to avoid the curse of dimensionality excessive parameters which lead to over-fitting and intractable complexity.

4. EXPERIMENTS AND RESULTS

The dataset used in the experiment was collected from the kaggle site of news headlines. The following well-known metrics were used for evaluation of the sarcasm detection task:

Precision: Precision (*Pr*) is the number of items correctly labeled as belonging to the positive class, (True Positives) divided by the total number of elements labeled as belonging to the positive class (the sum of True Positives and False Positives).

$$Pr = \frac{TP}{TP+FP} \quad (1)$$

Recall: Recall, *Re*, is defined as the number of True Positives divided by the total number of elements that actually belong to the positive class (the sum of True Positives and False Negatives).

$$Re = \frac{TP}{TP+FN} \quad (2)$$

F_1- Score: This is the harmonic mean of Precision and Recall.

$$F_1 = 2\ X\ \frac{Pos\ Pr\ X\ Pos\ Re}{Pos\ Pr + Pos\ Re} \quad (3)$$

Table 1: Performance of support vector machine (SVM) on News Headlines dataset without feature selection

Evaluation Parameter	Performance
Precision	64.20%
Recall	65.45%
F_1 Score	68.74%
Accuracy	68.12%

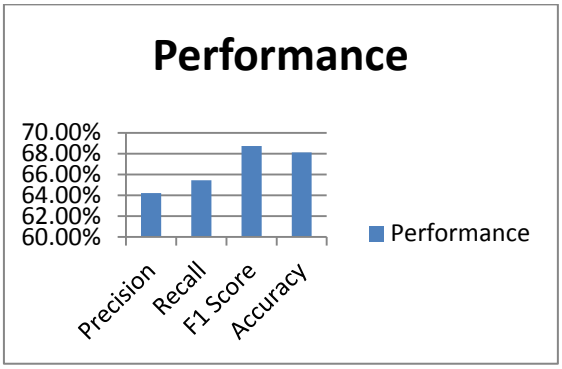
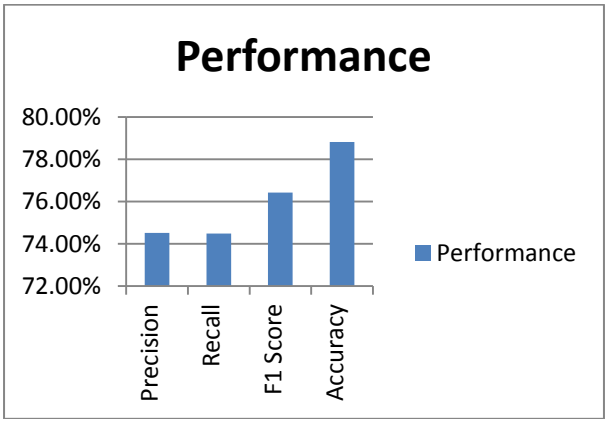


Table 2: Performance of support vector machine (SVM) on News Headlines dataset with optimal feature selection

Evaluation Parameter	Performance
Precision	74.52%
Recall	74.48%
F1 Score	76.42%
Accuracy	78.82%



5. CONCLUSION

With the increase in number of people using social media to express their views, tasks like opinion mining and sentiment analysis have gained a lot of importance. And using sarcasm in these social media texts make these tasks much more challenging. In this paper, we use the news headlines dataset for the sarcasm detection. This paper explained the methods used for collecting and annotating these news headlines both headlines level for presence of sarcasm as well as at token level for language. Paper also presented machine learning approach like support vector machine (SVM) as a classification task using the same dataset. Result shows that SVM perform better when the optimal feature set is given as an input. The provided classification system can be improved further by using various other features such as word embeddings, POS tags and other language based features.

6. REFERENCES

[1] Mondher Bouazizi and Tomoaki Otsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," IEEE Access Volume 4, 2016. pp. 5477- 5488.

[2] Shuigui Huang, Wenwen Han, Xirong Que and Wendong Wang, "Polarity Identification of Sentiment Words based on Emoticons," International Conference on Computational Intelligence and Security, 2017. Pp 134–138.

[3] S. Homoceanu, M. Loster, C. Lo_, and W.-T. Balke, ``Will I like it? Providing product overviews based on opinion excerpts," in Proc. IEEE CEC, Sep. 2011, pp. 26_33.

[4] Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? Lingua 116(10):1722–1743.

[5] Rachel Giora. 1995. On irony and negation. Discourse processes 19(2):239–264.

[6] Riloff, Ellen, et al. "Sarcasm as contrast between a positive sentiment and negative situation." *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013.

[7] Utsumi,A.(2004).Stylistic and contextual effects in irony processing. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, pages 1369–1374.

[8] Sana Parveen, Sachin N. Deshmukh, "Opinion Mining in Twitter – Sarcasm Detection" International Research Journal of Engineering and Technology (IRJET), volume 04, issue 10, pages 201-204, October 2017.

[9] Manoj Y. Manohar, PallaviKulkarni, "Improvement Sarcasm Analysis using NLP and Corpus based Approach", International Conference on Intelligence Computing and Control Systems (ICICCS), IEEE, 2017.

[10] Reyes, A., Rosso, P., and Buscaldi, D.(2012). From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

[11] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the EMNLP.

[12] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modeling sentences. In Proceedings of the 52nd ACL, pages 655–665. Association for Computational Linguistics.

[13] Shusen Zhou, Qingcai Chen, Xiaolong Wang, and Xiaoling Li. 2014. Hybrid deep belief networks for semisupervised sentiment classification. In Proceedings of COLING 2014, pages 1341–1349. Dublin City

University and Association for Computational Linguistics.

[14] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, pages 69–78.

[15] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In Proceedings of IJCAI, BueNos Aires, Argentina, August.

[16] Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mario J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In CONLL 2016.

[17] Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In Proceedings of the 7th WASSA, pages 161–169.

[18] Tom´as Pt´acek, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on Czech and English Twitter. In COLING, pages 213–223.