DOI: 10.21917/ijsc.2018.0233

SARCASM DETECTION IN ONLINE REVIEW TEXT

Srishti Sharma and Shampa Chakraverty
Division of Computer Engineering, Netaji Subhas Institute of Technology, India

Abstract

Sarcasm is a type of sentiment where people express negative sentiment using positive connotation words in text and vice-versa. In this work, we propose a cross-domain sarcasm detection framework that allows acquisition, storage and processing of tweets for detecting sarcastic content in online reviews. We conduct our experiments on Amazon product review dataset namely the Sarcasm Corpus Version1 having about 2000 reviews. We use Support Vector Machines (SVM) and Neural Networks (NN) for detecting sarcasm using lexical, pragmatic, linguistic incongruity and context incongruity features. We report the results and present a comparative evaluation of SVM and NN classifiers for single domain sarcasm detection indicating their suitability for the task. Then, we use these models for cross-domain sarcasm detection. The experimental results indicate the reliability of our approach.

Keywords:

Sarcasm, Machine Learning, Support Vector Machines, Neural Network Classifier, Amazon, Twitter

1. INTRODUCTION

Conversations in the modern age are dripping with sarcasm and thus it is a necessary skill today to comprehend when somebody is being sarcastic. Sarcasm is often confused as a synonym of irony, but it is more specific. Sarcasm is irony intended to mock or express contempt. For example, if someone says, "Woww! He's cleverer than Sherlock Holmes!" sarcastically, it conveys that the target said something simple or obvious and the speaker is mocking him. Sarcastic statements are often thought to be sincere, as they are mostly applicable to the situation on a superficial level, but are meant to be taken in the opposite way. In a verbal dialogue, presence of certain verbal or physical cues, like the speaker's facial expressions and voice intonations help in easily identifying sarcasm. On the other hand, detecting sarcasm in written text is a challenge due to the absence of these indicators. Since textual communication doesn't provide the opportunity to signal irony and provide cues one might expect that people would avoid it altogether. However, Jeff Hancock, a professor of communications at Stanford published a study suggesting that people may use sarcasm more frequently online than they do in face-to-face interaction [1]. Hancock noted that people are more inclined to pull off humor online as the general feeling is that its repercussions are less severe than when the target is not in physical proximity. Furthermore, he observed that online conversations offer more time to think vis-a-vis a face-to-face dialogue and people utilize this extra time to come up with complex, tongue-in-cheek, sarcastic statements.

Interest in Natural Language Processing (NLP) research has grown manifolds in the last decade or so and most of it is centered around analysis of social media text to solve problems like SA, co-reference resolution, metaphor detection, etc. [2]. On June 5, 2014, the BBC specified that the U.S. Secret Service was looking for a system capable of sensing sarcasm in online text [3]. The fact that even most people misconstrue irony and sarcasm is a key problem. As such, sarcasm detection is both a formidable task and yet of prime standing to innovation and improvement in the domain of artificial intelligence. Sarcasm detection is an essential pre-processing step in many NLP tasks like Sentiment Analysis (SA). Removal of sarcastic sentences filters out noisy samples before training models for NLP applications.

This paper investigates the possibility of classifying sarcasm in text reliably and in the process detecting the distinctive textual features essential for sarcasm detection.

2. RELATED WORK

Sarcasm is a perplexing issue in SA, and one which is not limited by the barrier of language. Several researchers across the globe, working on different languages have delved into the problem of identifying sarcasm from written text. Automatic detection of sarcasm is considered a challenging problem [4] [5].

Sarcasm detection approaches can broadly be classified into (1) Rule-based approaches, (2) Learning algorithms and features based approaches. Rule-based approaches make use of some rules to detect sarcasm which in turn have been derived by observing sarcastic texts. Learning algorithms and feature based approaches are dependent upon some identifying parameters used to detect sarcasm. The learning algorithms use these parameters as features.

A nine-rule approach for sarcasm detection is proposed in [6]. In [7], the author makes use of parse trees to identify situation phrases that bear sentiment and then detects sarcasm using the rule that the occurrence of a negative phrase in a positive sentence is an indicator of sarcasm.

Collecting enough sarcastic data to train a model for sarcasm detection is itself an unsurmountable task. From 2006, Twitter has become the single most valued tool across the entire range of NLP tasks. Authors in [8] and [9] propose using tweets author annotated with #sarcasm or #irony for accumulation of sarcastic text. Authors in [8] make use of unigram, bigram and trigram features for identifying sarcasm from text using Balanced Winnow and achieve an accuracy of 75%. Relying on the insights of authors in [10], most sarcasm detection approaches treat the task primarily as a text categorization problem, and use lexical and linguistic features such as interjections, intensifiers, non-veridicality and hyperbole, that is, three positive or negative words in a row.

In [11], the authors use a Naive Bayes classifier and a Support Vector Machine, for detecting sarcasm in Indonesian Social media text, using features like negativity and number of interjections. They use negativity to capture the global sentiment value and interjection to exemplify the lexical singularities in the typescript. They discover that negativity features are not really beneficial as a large number of sarcastic texts have no global topic, and marking the text topic is not recognized. Furthermore, an enormous amount of the text available online is private or one to one communication between two parties which cannot be examined without adequate prior knowledge of the context and the parties. Authors in [12] suggest the use of pattern based and punctuation based features with Support Vector Machines for sarcasm detection.

Authors in [13] experiment with Twitter data divided into three categories (sarcastic, positive sentiment and negative sentiment), each containing 900 tweets. They use the #sarcasm and #sarcastic hashtags to identify sarcastic tweets. They use two classifiers –Support Vector Machine (SVM) with Sequential Mining Optimization (SMO) and logistic regression. They try various combinations of unigrams, dictionary-based features and pragmatic factors including positive and negative emoticons and user references and utilize emotion and psychological process words. They refer to the Linguistic Inquiry and Word Counts (LIWC) achieving the best result for sarcastic and non-sarcastic classification with the combination of SVM with SMO and unigrams. They employ 3 human judges to annotate 180 tweets, 90 sarcastic and 90 non-sarcastic. The human judges achieve Fleiss' κ = 0.586, demonstrating the difficulty of sarcasm classification. Another experiment included 50 sarcastic and 50 non-sarcastic (25 positive, 25 negative) tweets with emoticons annotated by two judges. The automatic classification and human judges achieve the accuracy of 0.71 and 0.89 respectively. The inter annotator agreement using Cohen's κ was 0.74.

In [14], the authors identify one type of sarcasm: contrast between a positive sentiment and negative situation. They use a bootstrapping algorithm to acquire lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. They propose a method which classifies tweets as sarcastic if it contains a positive predicative that precedes a negative situation phrase in close proximity. Their evaluation on a human-annotated dataset of 3000 tweets out of which 23% were sarcastic was done using the SVM classifier with unigrams and bigrams as features, achieving an F-measure of 0.48. The hybrid approach that combines the results of the SVM classifier and their contrast method achieved an F-measure of 0.51.

In [15], Tomas et al. investigate supervised machine learning methods for language independent sarcasm detection by using features such as N-grams, Patterns, POS Tags, Punctuations and emoticons. They use a large human-annotated Czech Twitter dataset containing 7,000 tweets with inter annotator agreement k = 0.54.

Authors in [8] use intensifiers and exclamations to differentiate between sarcastic and non-sarcastic tweets. They work on a set of 3.3 million Dutch tweets and their classifier correctly classified 101 tweets from a set of 135 sarcastic tweets.

In [17], the authors use unigrams, quotation marks, ellipsis, positive and negative sentiment words followed by exclamation or question marks, hyperbole, interjection and laughter expressions as features.

Davidov et al. [18] devise a semi-supervised technique to detect sarcasm in Amazon product reviews and tweets. They use interesting patterns of high frequency words and content words and punctuation-based features to build a weighted K -nearest neighbor classification model to perform sarcasm detection.

Bouazizi et al. [19] devise an approach to identify the degree of sarcasm found in tweets using four sets of patterns, namely sentiment-related features, punctuation related features, syntactic features and pattern features. They obtain very high precision for sentiment-related features.

Ghosh et al. [20] model sarcasm detection as a word sense disambiguation task, and use embedding's to identify whether a word is used in the sarcastic or non-sarcastic sense. Two sense vectors for every word are created: one for literal sense and one for sarcastic sense. The final sense is determined based on the similarity of these sense vectors with the sentence vector.

In [21], the authors conduct a very extensive research on the writing style of the twitter user for detecting sarcasm. This feature was claimed to have 57.5% improvement on the accuracy alone. Btw, brb are examples of this feature.

According to authors in [22], hyperbole is the key element in generating sarcastic contents and they postulate that the presence of hyperbole in the text increases the likelihood of ironic interpretation.

3. PROPOSED WORK

Usually sarcasm is very cleverly embedded in a sentence, making sarcasm detection a challenging task. Context also plays a role in determining whether sarcasm is present as a hidden sentiment or not. Hence, it is an intricate task in the domain of Natural Language Processing. Rule-based model for detecting sarcasm would have very limited performance and its application would be specific to the data. Hence, we use Machine Learning for this task. In any Machine Learning task features are of central importance. The performance of the classifier depends on the features selected. Carefully designed and chosen features play a big role in improving the results. For our cross domain sarcasm detection framework, we used a pre-compiled sarcasm dataset from Twitter [23] as the training dataset, which was compiled in a manner similar to the approach described in [8]. The features used in our model can be divided into four categories namely: lexical, pragmatic, linguistic incongruity and context incongruity. This is represented in Fig.1.

3.1 LEXICAL FEATURES

N-grams are frequently used for many NLP tasks in Machine Learning. Trigrams, bigrams and unigrams with term presence or term frequency as features are the most commonly used. For our research, we use unigrams with term frequency as features. We use unigrams in order to extract the lexical information contained in the tweets. Using the training corpus a glossary of words is created. Then, the count or the frequency of occurrence of every word in a tweet is recorded as its feature value. The glossary of words is a very large list and every tweet would contain only a few words from this glossary. As such, the feature vector corresponding to every tweet would consist of a lot of 0's corresponding to the words that are not appearing in the tweet but are present in the glossary of words. However, these can be discarded since we are looking for presence of words prevalent in sarcastic tweets which can be a potentially important indicator while absence of words conveys no information.

3.2 PRAGMATIC FEATURES

Pragmatics is a branch of linguistics that examines expressions in depth and analyzes both their inferred and literal meanings. Pragmatics studies language that is not directly spoken, using hints. Pragmatic features include,

3.2.1 Number of Capital Letters:

The number of capital letters is used a feature. Capitalization is used to lay extra emphasis on the emotion to be conveyed. Similarly, sarcastic text is also usually highlighted by the author to create an extra impact. Hence this is used as a feature.

3.2.2 Number of Emoticons:

Emoticons are commonly used across social media platforms to express sentiments. As a feature, they can be captured using UTF-8 encoding.

3.2.3 Number of Slang Expressions:

Since sarcasm is intended to have an element of humor, higher occurrence of slangs is considered potentially indicative of sarcasm. Common slang words such as 'lol', 'rofl' and 'lmao' are fairly well used. Numerous alternates of these have also been accounted for. The frequency of occurrence of these expressions is used as a feature.

3.2.4 Number of Punctuation Marks:

Punctuation marks are usually used to lay extra emphasis on the underlying emotions like surprise, shock or dismay. It has been observed that in sarcastic tweets also, a large number of punctuation marks are used especially those like '!', '?' and '...'. The amount of punctuation marks is hence used as a feature.

3.3 LINGUISTIC INCONGRUITY

Linguistic Incongruity may be used for sarcasm detection as sarcastic text consists of positive polarity words used to describe a negative situation or vice-versa. Example, consider the expression: 'My brand new car just broke down. Yay!' The negative expression 'brand new car broke down' is incongruous to the positive situation implied by the word 'Yay' followed by an exclamation mark in order to highlight it. So, incongruity is a sign of complementary sentiments present in tweets, a symbol of sarcasm. The following features are used to examine linguistic incongruity.

3.3.1 Number of Sentiment Incongruities:

Using the SentiStrength tool [16], the polarity of each word is generated. The value generated lies the range [-5,5]. If the value is positive, it is taken as a positive word. Similarly, if the value is negative, it is taken as a negative word. A single numeric feature value which gives the count of the amount of times a negative word is followed by a positive word and vice-versa.

3.3.2 Count of Total Number of Positive and Negative Words:

Two features are generated namely the total count of the words with positive and negative polarity.

3.3.3 Largest Positive and Negative Sub-Sequence:

Size of the longest contiguous sequence of positive and negative features is used as a feature.

3.3.4 Lexical Polarity:

This is the overall polarity of the entire sentence. Owing to the theory of lexical incongruity, it is observed that a tweet which has an overall strong positive polarity has more likelihood of being sarcastic rather than a tweet with overall negative polarity. This is because in general sarcasm tends to be caustic. Using the SentiStrength tool [16], we get the polarity of each word as well as the overall sentence. For the sentence, a polarity score of -1 to -5 is considered to be negative polarity sentence, 0 is taken to be neutral while a score of +1 to +5 is taken to be of positive polarity.

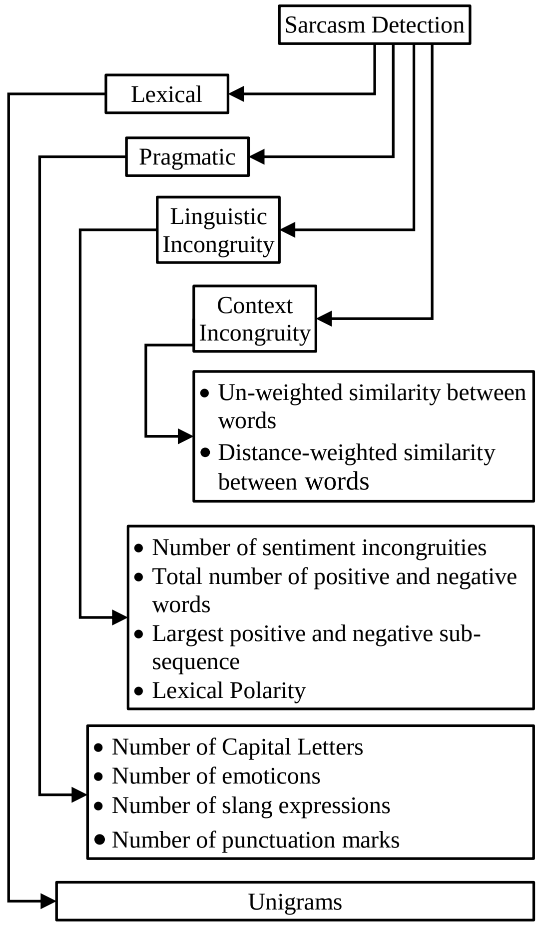

Fig.1. Different categories of text features used for Sarcasm Detection

3.4 CONTEXT INCONGRUITY

Word vector-based similarity or discordance is indicative of semantic similarity which in turn is a handle for context incongruity. Word embedding using word2vec model trained on Google News corpus are used to check for context incongruity. Gensim library is used.

- **Un-weighted similarity features (S)**: Similarity scores for all pairs of words, excluding stop words are computed. Then, four values per sentence are returned to check for Un-weighted similarity. These are Maximum and Minimum scores of most Similar as well as Dissimilar word pairs.
- **Distance-weighted similarity features (WS):** Similarity scores for all pairs of words, excluding stop-words are computed. All similarity scores are divided by the square of distance between the two words. As such, similarity between

terms physically closer to each other in a sentence is weighted higher than terms that are physically distant. Thus, for all possible word pairs, four features are recorded. These are Maximum and Minimum distance-weighted scores of most Similar and Dissimilar word pairs.

Using these features on the training dataset of tweets, we train our model based on SVM Classifier and Neural Network classifier. Then, we use this to test on dataset of Amazon product reviews.

4. EXPERIMENTS AND RESULTS

First, we test our sarcasm detection framework in single domain, dividing the Twitter Sarcasm Corpus Version1 available to us [23] into training and test datasets, with 95% training data and 5% test data. We carry out this step to ascertain the applicability of the proposed model for sarcasm detection. For this, we use Support Vector Machines (SVM) and Neural Networks (NN) and train using the features explained in Section 3. After training both the SVM and NN models using tweets and testing them in single domain, we use these models for cross-domain sarcasm detection, testing on 1995 Amazon product reviews, taken from the Sarcasm Corpus v1.

5. EVALUATION METRICS

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). This is represented in Eq.(1).

$$Pr = \frac{TP}{TP + FP} \tag{1}$$

where, TP: True Positive, TN: True Negative, FP: False Positive and FN: False Negative.

True positives indicate the number of positive files which are rightly classified as positive. Similarly True negatives indicate the number of negative files which are correctly classified as negative. False positives denote the number of negative files which are misclassified as positive. False negatives indicate the number of positive files that are misclassified as negative.

- **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). This is represented in Eq.(2).

$$Re = \frac{TP}{TP + FN} \tag{2}$$

- **F1- Score**: This is the harmonic mean of Precision and Recall. F1 score is calculated as shown in Eq.(3).

$$F_1 = 2 \times \frac{Pos\ Pr \times Pos\ Re}{Pos\ Pr + Pos\ Re} \tag{3}$$

6. RESULTS

For single domain sarcasm detection task, the results of sarcasm detection on the Twitter Sarcasm Corpus Version 1 using SVM and NN are as listed in Table.1. Both SVM as well as NN classifier perform well, with comparable F1 scores of 0.81 and 0.80 respectively, indicating their suitability for the sarcasm detection task. SVM reports higher Recall than NN but NN classifier gives more Precision.

Table.1. Experimental Results of SVM and NN Classifiers on Corpus of Tweets

	Precision	Recall	F1 Score
SVM	0.72	0.93	0.81
NN	0.76	0.83	0.80

We also report the F1 scores obtained by using different feature categories and the same are listed in Table.2. The Fig.2 represents the results diagrammatically. As we can see, combination of all the four feature categories reports the best performance. Lexical features alone are not suitable for sarcasm detection. Pragmatic, Linguistic and Context Incongruity features all individually also are suitable for sarcasm detection and the combination of these feature categories with one another and with lexical features keeps on improving the performance of the sarcasm detection model.

The results obtained for cross-domain sarcasm detection task are listed in Table.2. Both SVM and NN give comparable F1 scores of 0.66 and 0.65 respectively. The Fig.2 indicates that Recall values for both the classifiers are even better in cross-domain than in single domain also indicating the proposed approach's efficiency in detecting sarcastic utterances out of the total sarcastic utterances in cross-domain.

These results establish that models trained on tweets can be used to detect sarcasm on product reviews as well. This is of importance in several NLP applications like SA where there is a need to filter out sarcastic content and there may not be enough annotated sarcastic data for that domain and that level of granularity.

Table.2. F1 Scores by Different Feature Categories

Features	SVM	NN
Lexical (L)	0.37	0.38
Pragmatic (P)	0.64	0.61
Linguistic Incongruity (LI)	0.62	0.61
Context Incongruity (CI)	0.58	0.56
L + P	0.66	0.64
L + LI	0.63	0.63
L + CI	0.59	0.58
P + LI	0.74	0.72
P + CI	0.69	0.67
LI + CI	0.67	0.66
L + P + LI	0.74	0.75
L + P + CI	0.70	0.71

L + LI + CI	0.69	0.69
P + LI + CI	0.78	0.75
L + P + LI + CI	0.81	0.80

Table.3. Experimental Results of SVM and NN Classifiers on Amazon Product Review Corpus

	Precision	**Recall**	**F1 Score**
SVM	0.50	0.95	0.66
NN	0.51	0.88	0.65

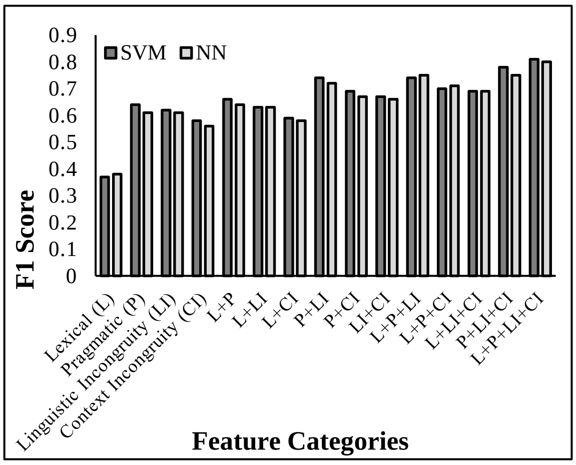

Fig.2. F1 scores using different categories of features for Single Domain Sarcasm Detection using SVM and NN

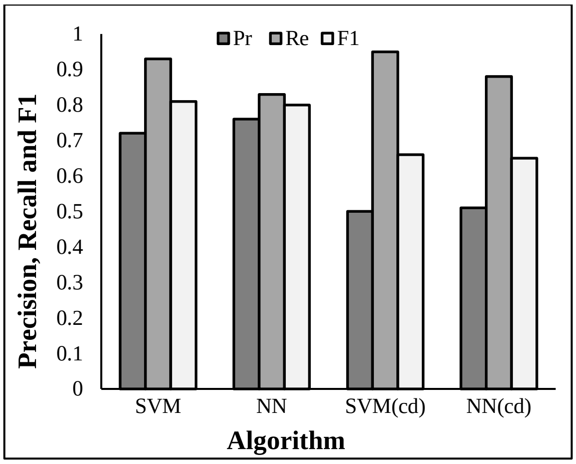

Fig.3. Precision, Recall and F1 values for Single and Cross-Domain Sarcasm Detection using SVM and NN

7. CONCLUSIONS AND FUTURE WORK

Through this work, we proposed a cross-domain sarcasm detection framework for detecting sarcastic content in online reviews using models trained on tweet data. This is of specific interest in SA systems, particularly in SA of reviews where there is a need to detect sarcasm but there is lack of trained data. On the other hand, there is always author hash tagged and explicitly identifiable sarcastic content on Twitter. We conducted our experiments on Amazon product review dataset namely the Sarcasm Corpus Version 1 having about 2000 reviews and showed that models trained on Twitter data can be used to detect sarcasm. We used Support Vector Machines (SVM) and Neural Networks (NN) for detecting sarcasm using lexical, pragmatic, linguistic incongruity and context incongruity features. The experimental results established that models trained on tweets can be used to detect sarcasm on product reviews as well. In the future, we will add more features like more lexical features, different word embedding based features-GLOVE etc. to increase the robustness of the proposed model. We will also test on a larger dataset of reviews.

REFERENCES

[1] J.T. Hancock, “Verbal Irony use in Face-to-Face and Computer-Mediated Conversations”, *Journal of Language and Social Psychology*, Vol. 23, No. 4, pp. 447-463, 2004.

[2] I. Habernal and J. Hong, “Sarcasm Detection on Czech and English Twitter”, *Proceedings of 25th International Conference on Computational Linguistics*, pp. 121-127, 2014.

[3] Us Secret Service Seeks Twitter Sarcasm Detector, Available at: http://www.bbc.com/news/technology-27711109, Accesssed on 2014.

[4] K. Nigam and M. Hurst, “Towards a Robust Metric of Polarity”, *Proceedings of International Conference on Computing Attitude and Affect in Text: Theory and Applications*, pp. 265-279, 2010.

[5] B. Pang and L. Lee, “*Opinion Mining and Sentiment Analysis*”, Now Publishers, 2008.

[6] Tony Veale and Yanfen Hao, “Detecting Ironic Intent in Creative Comparisons”, *Proceedings of European Conference on Artificial Intelligence*, pp. 765-770, 2010.

[7] S.K. Bharti, K.S. Babu and S.K. Jena, “Parsing-based Sarcasm Sentiment Recognition in Twitter Data”, *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 1373-1380, 2015.

[8] C. Liebrecht, F. Kunneman and A. Van Den Bosch, “The Perfect Solution for Detecting Sarcasm in Tweets #Not”, *Proceedings of 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 29-37, 2013.

[9] P. Reyes and T. Veale, “A Multidimensional Approach for Detecting Irony in Twitter”, *Proceedings of International Conference on Language Resources and Evaluation*, pp. 1-30, 2012.

[10] R.J. Kreuz and G.M. Caucci, “Lexical Influences on the Perception of Sarcasm”, *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pp. 1-4, 2007.

[11] E. Lunando and A. Purwarianti, “Indonesian Social Media Sentiment Analysis with Sarcasm Detection”, *Proceedings of International Conference on Advanced Computer Science and Information Systems*, pp. 212-215, 2013.

[12] Oren Tsur, Dmitry Davidov, and Ari Rappoport, “Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews”, *Proceedings of 4th International Conference on Weblogs and Social Media*, pp. 23-26, 2010.

[13] Roberto Gonzalez-Ibanez, Smaranda Muresan and Nina Wacholder, “Identifying Sarcasm in Twitter: a Closer

Look", *Proceedings of 49th Annual Meeting of the Association for Computational Linguistics*, pp. 581-586, 2011.

[14] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert and R. Huang, "Sarcasm as Contrast between a Positive Sentiment and Negative Situation", *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pp 704-714, 2013.

[15] Tomas Ptacek, Ivan Habernal and Jun Hong, "Sarcasm Detection on Czech and English Twitter", *Proceedings of 25th International Conference on Computational Linguistics*, pp. 213-223, 2014.

[16] Senti-Strength Tool, Available at: http://sentistrength.wlv.ac.uk/

[17] Konstantin Buschmeier, Philipp Cimiano and Roman Klinger, "An Impact Analysis of Features in a Classification Approach to Irony Detection in Product Reviews", *Proceedings of 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 42-49, 2014.

[18] D. Davidov, O. Tsur and A. Rappoport, "Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon", *Proceedings of 14th Conference on Computational Natural Language Learning*, pp. 107-116, 2010.

[19] Mondher Bouazizi and Tomoaki Ohtsuki, "Sarcasm Detection in Twitter", *Proceedings of International Conference in Global Communications*, pp. 331-334, 2015.

[20] Debanjan Ghosh, Weiwei Guo and Smaranda Muresan, "Sarcastic or Not: Word Embedding to Predict the Literal or Sarcastic Meaning of Words", *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp 1003-1012, 2015.

[21] Ashwin Rajadesingan, Reza Zafarani and Huan Liu, "Sarcasm Detection on Twitter: A Behavioral Modeling Approach", *Proceedings of 8th ACM International Conference on Web Search and Data Mining*, pp. 97-101, 2015.

[22] Data Mining Project focused on Twitter Sarcasm Measurement, Available at: https://github.com/dmitryvinn/twitter-sarcasm-measurement, Accessed on 2016.

[23] R.J. Kreuz and R.M. Roberts, "Two Cues for Verbal Irony: Hyperbole and the Ironic Tone of Voice", *Metaphor and Symbol*, Vol. 10, No. 1, pp. 21-31, 1995.