SARCASM DETECTION IN SENTIMENT ANALYSIS

Dr. Kalpesh H. Wandra[1], Mehul Barot[2]
[1]Director (Academic Administration) Babaria Institute of Technology,
[2]Research Scholar, C.U.Shah University

Abstract
Sentiment Analysis is a technique to identify people's opinion, attitude, sentiment, and emotion towards any specific target such as individuals, events, topics, product, organizations, services etc. Sarcasm is a special kind of sentiment that comprise of words which mean the opposite of what you really want to say (especially in order to insult or wit someone, to show irritation, or to be funny). People often express it verbally through the use of heavy tonal stress and certain gestural clues like rolling of the eyes. These tonal and gestural clues are obviously not available for expressing sarcasm in text, making its detection reliant upon other factors.

Keywords: Sarcasm, humour, machine learning, twitter, tweets, Politics, Political Sarcasm

I. Introduction

Sentiment analysis is the field of study that analyses people's sentiments, attitudes, and emotions from text. It is one of the most active research areas widely studied in data mining, Web mining, and text mining. Data mining refers to extracting knowledge from large amounts of data [1]. One of the subdomain of data mining is Web Mining which extracts knowledge from the WWW.[1][2]
The web mining is divided in to three domains [1] [2] which are as follows:
- •Web Usage Mining [2]
- •Web Content Mining [2]
- •Web Structure Mining [2]

Here for Sentiment Analysis the data of interest is only the text data, so Text mining is done on the content of the web. Text Mining, refers to the process of deriving high-quality information from text [4]. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text categorization, text clustering, sentiment analysis etc. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities).Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The overarching goal is, essentially, to turn text into data for analysis, via application of natural language processing (NLP) and analytical methods.
There are many challenges in Sentiment Analysis and one of them is sarcasm detection.

Sentiment analysis can be easily misled by the presence of words that have a strong polarity but are used sarcastically, which means that the opposite polarity was intended.

Sarcasm is a form of speech act in which the speakers convey their message in an implicit way. The inherently ambiguous nature of sarcasm sometimes makes it hard even for humans to decide whether an utterance is sarcastic or not.

Unlike a simple negation, a sarcastic sentence conveys a negative opinion using only positive words or intensified positive words. The detection of sarcasm is therefore important, for the development and refinement of Sentiment Analysis. Sarcasm is "a form of ironic speech commonly used to convey implicit criticism with a particular victim as its target". "Irony" and "sarcasm" are both ways of saying one thing and meaning another but they go about it in different ways.

A statement like Great, someone stained my new dress. is ironic, while You call this a work of art? Is sarcastic.

Sarcasm is a form of speech act in which the speakers convey their message in an implicit way. The inherently ambiguous nature of sarcasm sometimes makes it hard even for humans to decide whether an utterance is sarcastic or not. In this chapter, sarcasm is discussed in detail, what are the types of sarcasm and the challenges faced in detection of sarcasm. Unlike a simple negation, a sarcastic sentence conveys a negative opinion using only positive words or intensified positive words. The detection of sarcasm is therefore important, for the development and refinement of Sentiment Analysis.

II. Related Research

The automatic classification of communicative constructs in short texts has become a widely researched subject in recent years. Large amounts of opinions, status updates and personal expressions are posted on social media platforms such as Twitter. The automatic labeling of their polarity (to what extent a text is positive or negative) can reveal, when aggregated or tracked over time, how the public in general thinks about certain things. See Montoyo et al. (2012) for an overview of recent research in sentiment analysis and opinion mining.

A major obstacle for automatically determining the polarity of a (short) text are constructs in which the literal meaning of the text is not the intended meaning of the sender, as many systems for the detection of polarity primarily lean on positive and negative words as markers. The task to identify such constructs can improve polarity classification, and provide new insights into the relatively new genre of short messages and micro texts on social media. Previous works describe the classification of irony (Reyes et al., 2012b), sarcasm (Tsur et al., 2010), satire (Burfoot and Baldwin, 2009), and humor (Reyes et al., 2012a). Most common to our research are the works by Reyes et al. (2012b) and Tsur et al. (2010). Reyes et al. (2012b) collect a training corpus of irony based on tweets that consist of the hashtag #irony in order to train classifiers on different types of features (signatures, unexpectedness, style and emotional scenarios) and try to distinguish #irony-tweets from tweets containing the hashtags #education, #humour, or #politics, achieving F1-scores of around 70. Tsur et al. (2010) focus on product reviews on the World Wide Web, and try to identify sarcastic sentences from these in a semi-supervised fashion. Training data is collected by manually annotating sarcastic sentences, and retrieving additional training data based on the annotated sentences as queries. Sarcasm is annotated on a scale from 1 to 5. As features, Tsur et al. look at the patterns in these sentences, consisting of high-frequency words and content words. Their system achieves an F1-score of 79 on a test set of product reviews, after extracting and annotating a sample of 90 sentences classified as sarcastic and 90 sentences classified as not sarcastic.

In the two works described above, a system is tested in a controlled setting: Reyes et al. (2012b) compare irony to a restricted set of other topics, while Tsur et al. (2010) took from the unlabeled test set a sample of product reviews with 50% of the sentences classified as sarcastic. In contrast, we apply a trained sarcasm detector to a real-world test set representing a realistically large sample of tweets posted on a specific day of which the vast majority is not sarcastic. Detecting sarcasm in social media is, arguably, a needle-in-a-haystack problem (of the 3.3 million tweets we gathered on a single day, 135 are explicitly marked with the hashtag #sarcasm), and it is only reasonable to test a system in the context of a typical distribution of sarcasm in tweets. Like in the research of (Reyes et al., 2012b), we train a classifier based on tweets with a specific hashtag.

Class	Features	SMO	LogR
S-P-N	*Unigrams*	***57.22***	49.00
	$LIWC^+$_F	**55.59**	55.56
	$LIWC^+$_P	**55.67**	55.59
S-NS	*Unigrams*	***65.44***	60.72
	$LIWC^+$_F	**61.22**	59.83
	$LIWC^+$_P	62.78	**63.17**
S-P	*Unigrams*	***70.94***	64.83

	LIWC+_F	66.39	**67.44**
	LIWC+_P	67.22	**67.83**
S-N	*Unigrams*	***69.17***	64.61
	LIWC+_F	**68.56**	67.83
	LIWC+_P	68.33	**68.67**
P-N	*Unigrams*	**74.67**	72.39
	LIWC+_F	74.94	***75.89***
	LIWC+_P	**75.78**	**75.78**

Table 1: Sarcasm Results

III. Review of Literature

The research in this area is still going on. Not much work has been done on this topic, there are two ways for sarcasm detection and the most used way is the Machine learning based approach.

(1) Title: SARCASM DETECTION ON TWITTER: A BEHAVIORAL MODELLING APPROACH

Publication: WSDM '15 Proceedings of the Eighth ACM International Conference on Web Search and Data Mining Pages 97-106 ACM New York, NY, USA ©2015 table of contents ISBN: 978-1-4503-3317-7

Author: Ashwin Rajadesingan, Reza Zafarani, Huan Liu

Technology/Algorithm: SCUBA: Sarcasm Classification Using a Behavioral Approach

Conclusion: Different forms of sarcasm are discussed. Based on the type of sarcasm the features of sarcasm are identified. The features help in the training of the classifier.

(2) Title: Parsing-based Sarcasm Sentiment Recognition in Twitter Data

Publication: ASONAM '15 Proceedings of the 2015 IEEE/ACM on Advances in Social Networks Analysis and Mining 2015 Pages 1373-1380 ACM New York, NY, USA ©2015

Authors: S.Kumar Bharti ,K. Sathya ,S. Kumar Jena

Technology/Algorithm: PBLGA,IWS

Conclusion: The prime focus is on the interjection words and hyperbole.

(3) Title: Sarcasm as Contrast between a Positive Sentiment and Negative Situation

Publication: EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference Association for Computational Linguistics (ACL) Pages704-714 ISBN (Print)9781937284978

Author: E.Riloff, A. Qadir, P. Surve, L.De Silva, N.Gilbert, R. Huang

Technology/Algorithm: BOOTSTRAPPING ALGORITHM

Conclusion: Bootstrapping algorithm is introduced in which the sarcasm is detected as a contrast between positive sentiment and negative situation.

(4) Title: Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches. Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches.

Publication: 28th Pacific Asia Conference on Language, Information and Computation pages 404–413

Authors: Piyoros Tungthamthiti, Kiyoaki Shirai, and Masnizah Mohd

Technology/Algorithm: Concept level Analysis

Conclusion: In this paper, new concept of coherence is introduced also concept level analysis is done with the use lexicon called ConceptNet.

IV. Methodologies for sarcasm detection

The two main approaches of Sarcasm Detection are [3]:

•Machine learning approach

The machine learning approach applicable to sentiment analysis mostly belongs to supervised classification in general and text classification techniques in particular. Thus, it is called “Supervised learning”. In a machine learning based classification, two sets of documents are required: training and a test set. A training set is used by an automatic classifier to learn the differentiating characteristics of documents, and a test set is used to validate the performance of the automatic classifier. A number of machine learning techniques have been adopted to classify the reviews. Machine learning techniques like Naive Bayes (NB), maximum entropy (ME), and support vector machines (SVM) have achieved great success in text categorization.

Support Vector Machine

Naïve Bayes

•Lexicon based approach

The Lexicon-based Approach relies on a sentiment lexicon, a collection of known and precompiled sentiment terms. The sentiment lexicon is used to score the sentences either positive or negative or neutral. This approach scores every sentence on the basis of the existence of the positive or negative words. [3] . The lexicon-based approach involves calculating orientation for a document from the semantic orientation of words or phrases in the document.

5.1 Feature Selection

There are three main types of features for training the classifier are as follows:

•Lexical features

The lexical features are obtained from the unigram, bigram and trigram.

•Hyperbole

The hyperbole features are presence of the intensified positive words(adjectives), interjections, quotes, punctuation marks.

•Pragmatic features

The pragmatic features the presence of emoticons like frowning smileys, smiling faces etc and the mentions in the comments or the replies in case of twitter re-tweets.

V. Proposed Architecture

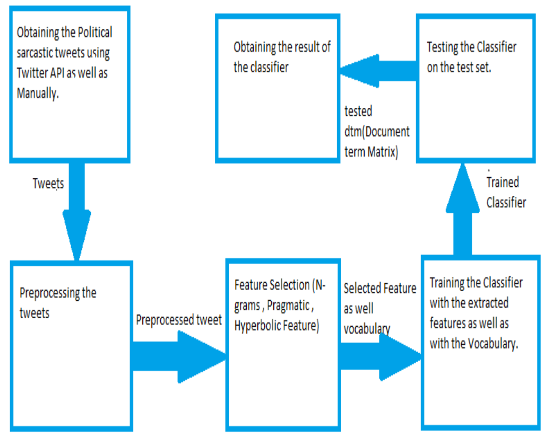


Figure 1: Proposed Architecture

The proposed Architecture follows the steps as given below:

i. The first step in the sarcasm detection is the collection of the relevant data. Here the approach chosen is the machine learning so the data needed is the test set and the training set. The data has to be pre-labelled so that the classifier can learn relevant features itself from the label. With the help of twitter API the tweets were obtained. The data of interest was sarcasm in political tweets so the tweets were obtained as per the requirement.

ii**.** After obtaining the tweets via Twitter API the tweets have to be preprocessed i.e the irrelevant data from the tweets has to be removed. The data that is irrelevant is the urls, name mentions, etc. in the tweets.

iii. After obtaining the preprocessed tweets, the features relevant for the classification are extracted. There are three main types of features they are Pragmatic Features, N-gram Features, Hyperbolic Features. The N-gram features are used here for the implementation.

iv. The obtained features along with the training set which contains preprocessed tweets are provided to the classifier. The classifier used in this implementation are Support Vector Machine and Logistic Regression.

v. Once the classifier has learned the varying characteristics through the training set as well as the dictionary provided to it the classifier is tested with the test set which is unseen to the classifier.

vi. And at last the results are obtained and the performance of the trained classifier can be obtained. The performance measure chosen for this system is Precision.

VI. Proposed Algorithm

6.1 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are the newest supervised machine learning technique .SVMs revolve around the notion of a "margin"—either side of a hyperplane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error.

If the training data is linearly separable, then a pair (w,b) exists such that

Equation 1

$w^T x_i + b \geq 1$, for all $x_i \in P$

$w^T x_i + b \leq -1$, for all $x_i \in N$ with the decision rule given by

$f_{w,b}(x) = sgn(w^T x + b)$where w is termed the weight vector and b the bias (or − b is termed the threshold). It is easy to show that, when it is possible to linearly separate two classes, an optimum separating hyperplane can be found by minimizing the squared norm of the separating hyperplane. The minimization can be set up as a convex quadratic programming (QP) problem:

Equation 2

$$Minimize_{w,b} \Phi(w) = \frac{1}{2} ||w||^2$$

Subject to $y_i(w^T x_i + b) \geq 1$, i=1,….l.

In the case of linearly separable data, once the optimum separating hyperplane is found, data points that lie on its margin are known as support vector points and the solution is represented as a linear combination of only these points (see Figure below). Other data points are ignored.

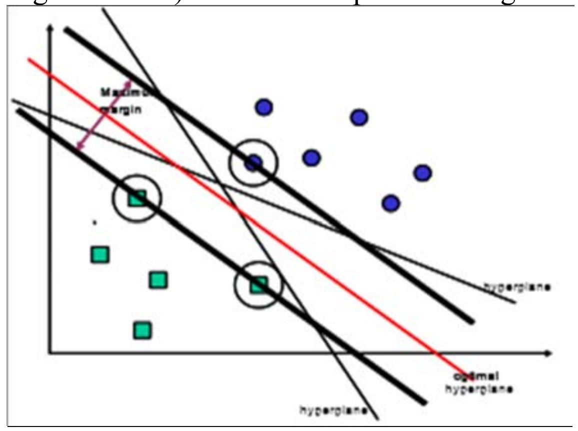


Figure 2 : Support Vector Points

Therefore, the model complexity of an SVM is unaffected by the number of features encountered in the training data (the number of support vectors selected by the SVM learning algorithm is usually small). For this reason, SVMs are well suited to deal with learning tasks where the number of features is large with respect to the number of training instances. Even though the maximum margin allows the SVM to select among multiple candidate hyperplanes, for many datasets, the SVM may not be able to find any separating hyperplane at all because the data contains misclassified instances. The problem can be addressed by using a soft margin that accepts some misclassifications of the training instances. This can be done by introducing positive slack variables ξ_i,i=1,…,N in the constraints which then become :

Equation 3

$$\begin{aligned} & w \cdot x_i - b \geq +1 - \xi \quad for \quad y_i = +1 \\ & w \cdot x_i - b \leq -1 + \xi \quad for \quad y_i = -1 \\ & \xi \geq 0, \end{aligned}$$

Thus, for an error to occur the corresponding ξ_i must exceed unity, so $\sum_i \xi_i$ is an upper bound on the number of training errors. In this case the Lagrangian is:

Equation 4

$$L_P \equiv \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i\{y_i(x_i \cdot w - b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i$$

Where the μ_i are the Lagrange multipliers introduced to enforce positivity of the ξ_i . Nevertheless, most real-world problems involve non separable data for which no hyperplane exists that successfully separates the positive from negative instances in the training set. One solution to the inseparability problem is to map the data onto a higher dimensional space and define a separating hyperplane there. This higher-dimensional space is called the transformed feature space, as opposed to the input space occupied by the training instances. With an appropriately chosen transformed feature space of sufficient dimensionality, any consistent training set can be made separable. A linear separation in transformed feature space corresponds to a non-linear separation in the original input space. Mapping the data to some other (possibly infinite dimensional) Hilbert space H as : $\Phi: R^d \longrightarrow H$ .Then the training algorithm would only depend on the data through dot products in H, i.e. on functions of the form $\Phi(x_i).\,\Phi(x_j)$. If there were a "kernel function" K such that K= $\Phi(x_i).\,\Phi(x_j)$, we would only need to use K in the training algorithm, and would never need to explicitly determine Φ. Thus, kernels are a special class of function that allow inner products to be calculated directly in feature space, without performing the mapping described above. Once a hyperplane has been created, the kernel function is used to map new points into the feature space for classification. The selection of an appropriate kernel function is important, since the kernel function defines the transformed feature space in which the training set instances will be classified. Genton (2001) described several classes of kernels, however, he did not address the question of which class is best suited to a given problem. It is common practice to estimate a range of potential settings and use cross-validation over the training set to find the best one. For this reason a limitation of SVMs is the low speed of the training. Selecting kernel settings can be regarded in a similar way to choosing the number of hidden nodes in a neural network. As long as the kernel function is legitimate, a SVM will operate correctly even if the designer does not know exactly what features of the training data are being used in the kernel-induced transformed feature space.

Some popular kernels are the following:

Equation 5

(1) $K(x,y) = (x \cdot y + 1)^p$,

(2) $K(x,y) = e^{-\|x-y\|^2/2\sigma^2}$,

(3) $K(x,y) = \tanh(\kappa x \cdot y - \delta)^p$

Training the SVM is done by solving N^{th} dimensional QP problem, where N is the number of samples in the training dataset. Solving this problem in standard QP methods involves large matrix operations, as well as time-consuming numerical computations, and is mostly very slow and impractical for large problems. Sequential Minimal Optimization (SMO) is a simple algorithm that can, relatively quickly, solve the SVM QP problem without any extra matrix storage and without using numerical QP optimization steps at all (Platt, 1999). SMO decomposes the overall QP problem into QP sub-problems. Keerthi and Gilbert (2002) suggested two modified versions of SMO that are significantly faster than the original SMO in most situations. Finally, the training optimization problem of the SVM necessarily reaches a global minimum, and avoids ending in a local minimum, which may happen in other search algorithms such as neural networks. However, the SVM methods are binary, thus in the case of multi-class problem one must reduce the problem to a set of multiple binary classification problems. Discrete data presents another problem, although with suitable rescaling good results can be obtained.

6.2 Logistic Regression

Logistic regression is another technique borrowed by machine learning from the field of statistics.It is the go-to method for binary classification problems (problems with two class values). In this post you will discover the logistic regression algorithm for machine learning.

Logistic Function

Logistic regression is named for the function used at the core of the method, the logistic function.The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

Equation 6

1 / (1 + e^-value)

Where e is the base of the natural logarithms (Euler's number or the EXP() function in your spreadsheet) and value is the actual numerical value that you want to transform. Logistic regression uses an equation as the representation, very much like linear regression.Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

Equation 7

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data. The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's). The coefficients (Beta values b) of the logistic regression algorithm must be estimated from your training data. This is done using maximum-likelihood estimation. Maximum-likelihood estimation is a common learning algorithm used by a variety of machine learning algorithms, although it does make assumptions about the distribution of your data (more on this when we talk about preparing your data).

The best coefficients would result in a model that would predict a value very close to 1 (e.g. male) for the default class and a value very close to 0 (e.g. female) for the other class. The intuition for maximum-likelihood for logistic regression is that a search procedure seeks values for the coefficients (Beta values) that minimize the error in the probabilities predicted by the model to those in the data (e.g. probability of 1 if the data is the primary class).We are not going to go into the math of maximum likelihood. It is enough to say that a minimization algorithm is used to optimize the best values for the coefficients for your training data. This is often implemented in practice using efficient numerical optimization algorithm (like the Quasi-newton method).

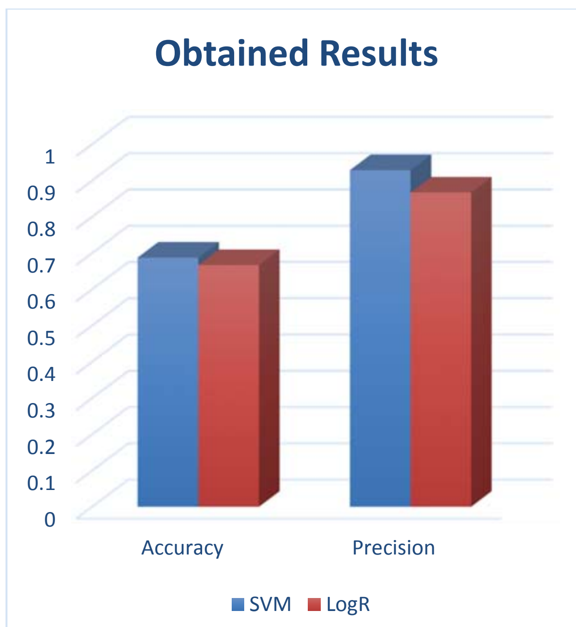
VII. Dataset

The dataset has been scraped manually also, it is in csv format. The tweets are labelled as sarcasm and regular. The manual labelling was important as it would provide proper training to the classifier. At this point the data is not as per the requirement but this dataset is working as the initial seed data and once the dynamism is incorporated after the front end development the size of the data will keep on increasing and the classifier will be trained as per the requirement.

VIII. Results

	CURRENT	CURRENT	EXISTING	EXISTING
ALGORITHM	SVM	LogR	SVM	LogR
Accuracy	0.6875	0.66667	0.654	0.607
Precision	0.8667	0.92857	Not Available	Not Available

Table 2: Sarcasm Results



IX. Conclusion and Future Scope

The Sarcasm Detection in Sentiment Analysis is still one of the most widely studied challenge in sentiment analysis. The research is still going on in this area. In this Dissertation I have tried to implement this challenge and at present I have successfully created the model. The model uses N-gram features as well as a vocabulary of 7000 words. The model has faired pretty well in terms of precision giving the values 0.866, 0.928 for Logistic Regression as well as Support Vector Machine. In future I will try to implement the model with a few thousand tweets also in place of static vocabulary I will try to incorporate a dynamic model in which as per the user feedback the model will be able to train itself so that its future performance came be improved. In order to incorporate dynamism in the model, the front end as well as backend has to be carefully developed. The front end will contain a radio button which will record the user feedback about the result given by the model. If the user gives a negative feedback the given input will be saved and added to the dataset with the appropriate label and passed to the model for the training as per the new data. This will make the system more reliable in terms of the results.

REFERENCES

[1] Data Mining Concepts and Techniques, J. Han M. Kamber

[2] Web Content Mining: Its Techniques and Uses IJARCSSE Volume 3 Issue 11 2013 G.Upadhyay K.Dhingra

[3] Sentiment analysis algorithms and applications: A survey Ain Shams Engineering Journal Volume 5 Issue 6 December 2014

[4]] Sarcasm as Contrast between a Positive Sentiment and Negative Situation E.Riloff, A. Qadir, P. Surve, L.De Silva, N.Gilbert, R. Huang EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference Association for Computational Linguistics (ACL) Pages704-714 ISBN (Print)9781937284978

[5] Parsing-based Sarcasm Sentiment Recognition in Twitter Data S.Kumar Bharti K. Sathya S. Kumar Jena ASONAM '15 Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 Pages 1373-1380 ACM New York, NY, USA ©2015

[6] Identifying Sarcasm in Twitter: A Closer Look R. González-Ibáñez S. Muresan N. Wacholder HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Volume 2 581-586 Association for Computational Linguistics Stroudsburg, PA, USA ©2011

[7] The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods Y. Tausczik and J. Pennebaker Journal of Language and Social Psychology 29(1) 24–54

[8] SARCASM DETECTION ON TWITTER: A BEHAVIORAL MODELLING APPROACH WSDM '15 Proceedings of the Eighth ACM International Conference on Web Search and Data Mining 97-106 ACM New York, NY, USA ©2015 ISBN: 978-1-4503-3317-7 Ashwin Rajadesingan Reza Zafarani Huan Liu

[9] Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches P. Tungthamthiti K. Shirai M. Mohd

[10] Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon D. Davidov, O. Tsur, A. Rappoport Institute of Computer Science The Hebrew University Jerusalem, Israel Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 107–116, Uppsala, Sweden, 15-16 July 2010.

[11] The perfect solution for detecting sarcasm in tweets #not C. Liebrecht F. Kunneman Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 29–37, Atlanta, Georgia, 14 June 2013.

[12] Contextualized Sarcasm Detection on Twitter D. Bamman, N. Smith Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org)

[13] Signaling sarcasm: From hyperbole to hashtag Florian Kunneman, Christine Liebrecht, Margot van Mulken, Antal van den Boscha

Information Processing and Management (2014), 2014 Elsevier Ltd. All rights reserved

[14] The CLSA Model: A novel framework for concept-level sentiment analysis E. Cambria, S. Poria, F. Bisio, R. Bajpai, and I. Chaturvedi

[15] From Humor Recognition to Irony Detection: The Figurative Language of Social Media A. Reyesa, P. Rossoa, D. Buscaldib Data and Knowledge Engineering. 74:112 Elsevier

[16] Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis D. Maynard, A. Greenwood IREC

[17] Sentiment Analysis: A Review and Comparative Analysis of Web Services Serrano-Guerrero A. Olivas, P. Romero, E. Herrera-Viedm Elsevier February 20, 2015

[18] Sentiment Analysis: Capturing Favorability Using Natural Language Processing Tetsuya Nasukawa, Jeonghee Yi K-CAP'03, October 23–25, 2003, Sanibel Island, Florida, USA. Copyright 2003 ACM 1-58113-583-1/03/0010

[19] Tweet Sarcasm: Mechanism of Sarcasm Detection in Twitter Komalpreet Kaur Bindra et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 2016, 215-217

[20] Supervised Machine Learning: A Review of Classification Techniques

[21] CS838-1 Advanced NLP: Text Categorization with Logistic Regression Xiao jin Zhu

[22]http://machinelearningmastery.com/logistic-regression-for-machine-learning/

[23] Text Classification in Information Retrieval using Winnow P.P.T.M. van Mun Department of Computing Science, Catholic University of Nijmegen Toernooiveld 1, NL-6525 ED, Nijmegen, The Netherlands

[24] An Introduction to Statistical Learning Gareth James Daniela Witten Trevor Hastie Robert Tibshirani

[25] http://www.thesarcasmdetector.com/about/