IMPROVING THE SENTIMENT ANALYSIS OF SARCASTIC TWEETS
USING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING

A Master’s Paper
by Aayushi Darade

© 2021 Aayushi Darade

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Information Systems

May 2021
ABSTRACT

With the developing technology, the number of comments made on the internet is increasing day by day. It is difficult to make a manual sentiment analysis on these comments. Therefore, new algorithms should be developed to automatically perform sentiment analysis on these texts for companies. In this study, a sentiment analysis model using machine learning and NLP (Natural Language Processing) algorithms are compared. While developing this model, CNN (Convolutional Neural Network) methods and machine learning algorithms were used together. As a naïve method of sentiment analysis, the root of each word in a sentence takes a score from a dictionary and the final polarity score of the relevant sentence is calculated by using additive score-based models. Machine learning models and Natural Language Processing models were trained to perform accurate sentiment annotations by using features based on polarity scores of texts. This analysis was conducted on 10,000 tweets using publicly available data from Kaggle. The results showed that NLP outperforms machine learning algorithms and gives better accuracy.
# TABLE OF CONTENTS

Chapter 1. Introduction.............................................................................................................1

Chapter 2. Literature Review..................................................................................................4
  2.1 Sentiment Analysis of Twitter Data.................................................................................4
  2.2 Techniques for Sarcasm Detection...................................................................................5
    2.2.1 Natural Language Processing...................................................................................8
    2.2.2 Machine Learning...................................................................................................8

Chapter 3. Development of Hypothesis..................................................................................10

Chapter 4. Methodology........................................................................................................12
  4.1 Sample and Measures....................................................................................................12
  4.2 Procedure....................................................................................................................12
  4.3 Machine Learning Models............................................................................................13
    4.3.1 SVM (Support Vector Machine).............................................................................14
    4.3.2 Maximum Entropy Classifier...............................................................................15
    4.3.3 Naïve Bayes..........................................................................................................15
  4.4 Natural Language Processing Model...............................................................................16

Chapter 5. Data Analysis......................................................................................................18
  5.1 Cleaning of Raw Data....................................................................................................18
  5.2 Feature Engineering.......................................................................................................19
    5.2.1 Lexical Features.....................................................................................................19
    5.2.2 Pragmatic Features...............................................................................................20
    5.2.3 Incongruity Features.............................................................................................20
    5.2.4 Sentiment Features...............................................................................................22
    5.2.5 Topic Feature........................................................................................................23
  5.3 Key Performance Indicators Calculation......................................................................24

Chapter 7. Results...............................................................................................................26

Chapter 8. Discussion...........................................................................................................27
  Summary of Results...........................................................................................................27
  Limitations and Future Research.......................................................................................28

References...........................................................................................................................29
LIST OF TABLES

Table 1. Comparison of Machine Learning and Natural Language Processing Classifiers……26
LIST OF FIGURES

Figure 1. Hyperplane SVM model.................................................................14

Figure 2. CNN model...................................................................................17
ACKNOWLEDGEMENTS

I would like to thank my course instructor Dr. Katherine Hamilton for her guidance since the start of my paper. Dr. Hamilton has provided very helpful feedback in every draft of my paper. I would also like to thank Dr. Roderick Lee for helping me in deciding the topic and guiding me on that.
CHAPTER 1

Introduction

Around 4.1 billion people have access to the internet, which represents about half of the global population (Johnson, 2021). This number has been growing at an enormous rate since the year 2000, primarily due to the smartphone revolution in the telecom sector. Almost half of people create their own content, share videos, images or repost other user’s content and create content for companies to analyze their products (Gallegos, 2021). Among social media platforms, Twitter is the second largest. Twitter generates 347,222 tweets each minute and 21 million tweets per hour (Kharde & Sonawane, 2016). It is therefore no surprise that many studies are done on sentiment analysis of Twitter data using machine learning approaches (Gautam & Yadav, 2014). Though several researchers have studied soft computing techniques, only a few types of techniques have been explored. Techniques such as deep learning, evolutionary computing, optimization algorithms and hybrid approaches, including neuro-fuzzy models, have been the least explored or implemented to substantiate their influence on sentiment analysis (Kumar & Jaiswal, 2020).

Sentiment analysis is one of the hardest tasks to complete in natural language processing because even humans struggle to analyze sentiments accurately (Maynard & Greenwood, 2014). Sentiments are difficult to analyze because all utterances are uttered at some point in time, in some place, by, and to some people. All utterances are uttered in context. Analyzing sentiment without context is therefore fairly difficult. The area that is most difficult to classify in natural language processing is sarcastic text. Sarcastic text is characterized by the expression of negative sentiments using positive words. These types of expressions more easily cheat sentiment analysis
models because of the mixing of sentiments. These expressions are often incorrectly classified unless the analysis is specifically designed to take its presence into account.

Sarcasm occurs most often in user-generated content such as tweets (Bharti, Vachha & Pradhan, 2016). Sarcasm detection in sentiment analysis is very difficult to accomplish without having a good understanding of the context of the situation, the specific topic, and the environment. For example, there are several possible answers to the question, “Did you enjoy your shopping experience with us?” A response could be “Yeah, sure. So smooth!” This response would be classified as negative because of the presence of an exclamation mark. Another response could be “Not one, but many!”

Word ambiguity is another pitfall for sentiment analysis problems. The problem of word ambiguity is the impossibility to define polarity in advance because the polarity for some words is strongly dependent on the sentence context. The following two examples show how context affects word sentiment. One example is, “The story is unpredictable.” Another example is, “The steering wheel is unpredictable.” In the first example, the word polarity of “unpredictable” is predicted as positive. Whereas in the second example, the same word’s polarity is negative.

Given the current concerns with sentiment analysis, the purpose of the current study is to evaluate how different classifications impact the accuracy of the model and how deep learning and Convolutional neural networks (CNN) helps in detecting sarcasm and increase the accuracy and end result of the model. Through addressing this purpose, the study will make the following contributions. First, it will apply deep learning to sarcasm detection. This has been an underexplored area in the literature. Second, it will leverage user profiling, emotion, and sentiment features for detecting sarcasm in tweets to increase the accuracy of their classification. The high dimensionality of the input feature space in comparison with the relatively small
number of subjects (i.e., the curse of dimensionality; (Prakruthi, Sindhu, and Kumar (2018)) is a widespread concern, so some form of feature selection is often applied. If the features are selected properly the model is trained more accurately and accuracy increases. Third, it will concentrate on different CNN models and which is the most accurate. Convolutional Neural Networks (CNNs) are more accurate over the state-of-the-art traditional models. Genetic algorithms are usually used to construct a CNN architecture automatically and in real time application for image classification with automatically generated CNN architecture with the help of it. Embedding layers are used like a perfect launch pad. Embedding layer provides words to vector conversion and for computational purposes numbers are always easy to understand from a machine's perspective. A comparison of these approaches will provide unique information on the most accurate models.
CHAPTER 2

Literature Review

2.1 Sentiment Analysis of Twitter Data

Twitter is the second biggest social networking platform after Facebook, which generates 347,222 tweets each minute and 21 million tweets per hour (Kharde & Sonawane, 2016). It therefore creates an opportunity for data mining and sentiment analysis based on users’ tweets. Prakruthi, Sindhu, and Kumar (2018) contended that sentiment is nothing but an idea or view based on emotion. Sentiment analysis comprises analyzing emotional opinions from different groups of users and determining their attitude towards the complete contextual polarity or emotional response to a document, communication, or event. Nowadays people express their views on politics, products, and food items on social media platforms. Hence social media is generating a large volume of sentiment rich data in the form of tweets, status updates, blog posts, comments, and reviews.

People mostly depend upon user generated content over online to a great extent for decision making (Kharde & Sonawane, 2016). For instance, if someone wants to buy a product or wants to use any service, then they firstly look up its reviews online and discuss it on social media before making a decision. The amount of content generated by users is too vast for a normal user to analyze. So there is a need to automate this, hence various sentiment analysis techniques are widely used.

Both of these articles have similar ideas about what sentiment analysis is. Kharde and Sonawane (2016) claimed sentiment analysis tells users whether the information about the product is satisfactory or not before they buy it. Marketers and firms use this analysis data to understand about their products or services in such a way that it can be offered as per the user’s
requirements. Similarly, Prakruthi et al. (2018) explained how sentiment analysis allows users to know the situation of what the review is made with respect to a particular item rather than understanding the deep thoughts of the reviewer. Sentiment analysis also helps people to change their attitude about wrong belief on a product, service, or topic.

2.2 Techniques for Sarcasm Detection

Sarcasm occurs most often in user-generated content such as tweets (Bharti, Vachha & Pradhan, 2016). Sarcasm detection in sentiment analysis is very difficult to accomplish without having a good understanding of the context of the situation, the specific topic, and the environment. The author talks about how sarcasm detection is inherently challenging, the nature and style of content on Twitter further complicate the process of detecting it as Twitter is more informal in nature with an evolving vocabulary of abbreviations and slang words (Porwal & et.al, 2018). The issue of sarcasm detection is more complex in the case of social media texts written in more than one language. The author also states the point that the misspelled words, shortened word forms and stylistic text coupled with the use of dual language is commonplace in Malaysian social media texts, where it is not unusual to mix Malay and English. Detecting sarcastic contents in such bilingual situations adds an extra level of complication to the challenge of sarcasm detection (Gidhe & Ragha 2017).

Kharde and Sonawane (2016) proposed various techniques used for sentiment analysis using machine learning. The first approach which is explained is about machine learning approach where supervised and unsupervised algorithms can be used for analysis. The second approach which the author explained is lexicon based. It used a sentiment dictionary which matches opinion words of users with data to determine polarity and then decide whether the tweet is positive, negative, or neutral. Machine learning approaches have better results and
accuracy than the lexicon based approach. The author gives the comparison of techniques used and which one tells us which approach gives better results. As the study shows ML approach is better, it only talks about one algorithm - Sector Vector Machine (SVM). As different algorithms work differently on the data, there is a possibility that other algorithms can perform better than SVM.

Dhawan, Singh, and Chauhan (2019) explained how sentiment analysis on Twitter data is done using Artificial Intelligence. This begins with gathering and preparing a dataset of tweets about a given product. The author then explains how the model is trained on the preparation data which is also known as training data. In the model first the authenticity of the user is checked with the help of secret key tokens. If a user is authorized then algorithms start sentimental analysis. To analyze sentiments, first features of Twitter are extracted then sentiments of Twitter are analyzed. The sentiment polarity of each tweet is then checked. The sentiment polarity is nothing but emotions of users like joy, happiness, sadness and anger. If the sentiment polarity is equal to zero then the tweet is neutral and if polarity is greater than zero then the tweet is positive otherwise, the tweet is negative.

Prakruthi, Sindhu and Kumar (2018) explained how Twitter data is classified based on sentiments. First tweets are collected by taking the input in the form of hashtags and each word in the tweet is tokenized. Tokenization refers to the act of breaking up a string sequence into pieces such as phrases, words, keywords, and symbols. Cleaning of tweets is done by removing additions of text in the tweets like URLs, numbers, and special characters which shortens the size of tweets for comparison. After that, the pre-processed tweets are compared with the available bag of words (BoW) and are classified as positive, negative, and neutral. BoW calculates how many times a word appears in a document. Fernandes and D'Souza (2016) concentrated on
providing the opinion on the particular products using the Twitter data. There are millions of reviews on a single product and it would be impossible for the customer or the organization to read each review and judge the quality of the product. This research provides opinion mining on particular products based on reviews. The work includes determination of positivity, negativity of tweets and provides overall percentage of positive, negative and neutral tweets.

Although the technique used by Prakruthi, Sindhu, and Kumar (2018) looks perfect on the surface, it has some definite shortcomings. Consider the text, “The service was horrible, but the ambiance was awesome!” Now, this sentiment is more complex than a basic algorithm and can take into account – it contains both positive and negative emotions. For such cases, this algorithm won’t work.

Fernandes and D’Souza’s (2016) idea is that the customer should automatically get suggestions about the product based on previous tweets. This paper proposed a system which is based on unsupervised learning method. Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. Only the recent reviews are collected from Twitter and the polarity of each tweet is determined. Based on the polarity the implementation identifies whether the tweet is positive, negative or neutral. Customers tweet their experience based on products like shopping facility, product features and usage of product. Millions of such tweets are generated, based on a particular product, and the system decides whether the product is good or bad. This research focuses on automatically analyzing the positive, negative and neutral opinions tweeted by the users and providing the consolidated opinion about a product to the user.
2.2.1 Natural Language Processing

Natural Language Processing is a method that is being used to detect sarcasm. The classifier for natural language processing is a tool that takes in the data-items and categorizes them into one of the k-classes. In natural language processing (NLP) various tasks are performed such as tokenization, part of speech tagging (PoS), and lemmatization. Sarcasm detection is a very narrow research field in NLP, a specific case of sentiment analysis where instead of detecting a sentiment in the whole spectrum, the focus is on sarcasm. Therefore, the task of this field is to detect if a given text is sarcastic or not (Porwal & et.al, 2018).

2.2.2 Machine Learning

Siddiqua, Ahsan, and Chy (2016) researched about emoticons and sentiment-bearing words and combined it with the weakly supervised Naive-Bayes classifier to classify the tweets using analysis tools that can perform well with minimum supervision. Shaikh and Lobo (2016) explained sentiment analysis using data mining techniques. Data Mining finds useful information from a huge data set and helps in knowledge data discovery. Data mining has many classification techniques, static as well as dynamic. Classification model mainly requires training data and testing data. Siddiqua, Ahsan, and Chy (2016) researched on a rule-based classifier, where a set of rules usually constructed to determine a certain combination of patterns, that are most likely to be related to the different classes. Each rule consists of two parts: the antecedent part and the consequent part. The antecedent part corresponds to a word pattern and the consequent part corresponds to a class label. Unsupervised rule-based classifiers cast the sentiment analysis problem as a multi-class classification problem and label each tweet as positive, negative or unknown. The authors utilized several sentiment lexicons to train the Naive-Bayes classifier and found that it is effective rather than using large training dataset. The study suggests that
classification models of supervised as well as unsupervised machine learning algorithms help in predicting sentiments of people.

Shaikh and Lobo (2016) explained a classification model which classifies tweets into three categories (positive, negative and neutral), and supervised learning algorithm Dynamic LM Classifier is used. The algorithm works on semantic models. This algorithm implements training and classification, used in tag-a-little, learn-a-little supervised learning without retraining epochs which makes it ideal for active learning applications. This model predicts the sales performance of the particular product for which classification is done. The positive to negative tweets ratio is considered for predicting the sales performance. This prediction model is a heuristic model. It predicts sales performance by observing historic data. Favorite count is taken into consideration for ranking the reviews after classification. The Tweet which has the greatest number of likes has the highest ranking. This ranking is useful to a customer as well as a vendor to know the favorite tweet of a user about a product or service.

Although very limited work has been done on using neural networks and NLP 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 (Socher et al., 2013), dynamic pooling networks (Kalchbrenner et al., 2014), deep convolutional networks (dos Santos & Gatti, 2014) and neural CRF (Zhangetal, 2015). This line of work gives highly competitive results, demonstrating large potentials for neural networks on sentiment analysis. One of the important reasons is the power of neural networks in automatic feature induction and this motivates researchers for working further on neural networks for detecting sarcasm.
CHAPTER 3

Development of Hypothesis

The advantages of using neural networks over machine learning is that it can detect sarcasm in the tweet more accurately. Sarcasm is a type of sentiment where people express their negative feelings using positive or intensified positive words in the text. While speaking, people often use heavy tonal stress and certain gestural clues such as, rolling of the eyes and hand movement to reveal sarcastic. When writing, these tonal and gestural clues are missing. This makes sarcasm detection very difficult for the average human (Maynard & Greenwood, 2014). Deep neural network models compared with traditional models with manual discrete features, the neural network model has two main advantages. First, it is from manual feature engineering and external resources such as POS taggers and sentiment lexicons. Second, it leverages distributed embedding inputs and recurrent neural networks to induce semantic features. The neural network model gives improved results over a state-of-the-art discrete model (Zhang & Fu, 2016). CNN outperforms machine learning algorithms in detecting sarcasm and irony in the sentences. The research shows when machine learning algorithm - Support Vector Machine (SVM) gives better accuracy amongst all and is most commonly used for sarcasm detection in Twitter. In addition, combining Convolutional Neural Network (CNN) and SVM was found to offer a high prediction accuracy (Al-Samarraie, Sarsam, & Wright, 2020; Fu & Zhang, 2016).

Companies do not get accurate feedback using machine learning algorithms every time for their products using sentiment analysis for Twitter data. Companies mostly depend upon user generated content over online to a great extent for decision making. Most of the companies come to a conclusion after seeing customer’s reaction for their products. With the sentiment data about establishment about the new products, it’s easier to estimate customer retention rate. Based on
the reviews generated through sentiment analysis in business, organizations can always adjust to the present market situation and satisfy their customers in a better way (Smailović, Grčar, Lavrač, & Žnidaršič 2013). Overall, companies can make immediate decisions with automated insights. As the amount of content generated by users is too vast for a normal user to analyze, various sentiment analysis techniques are widely used by the companies to get ideas about their products in the market. The main concern is the accuracy of this sentiment analysis. Few errors in classification can mislead companies about their products. There are more chances of misclassification when it comes to sarcasm, word ambiguity, or sentences used in other contexts as machines cannot learn context.

Based on these trends in previous research the current study will test the hypothesis that natural language processing classifiers will be more accurate at classifying sarcastic tweets than machine learning classifiers.
4.1 Sample and Measures

The data was publicly available through Kaggle.com (https://www.kaggle.com/nikhiljohnk/tweets-with-sarcasm-and-irony). The size of the data was around 10,000 records. The records included hashtags with sarcastic tweets or sarcasm and without hashtags as well. The data included twitter comments, negative, positive, neutral, company as the columns. The negative column showed whether the tweet was negative or not. Similarly, positive and neutral columns showed whether the tweet was positive or neutral. The company column showed whom the tweet was about. Out of 10,000 data points, 80,000 was trained and 2,000 was used for the test data.

4.2 Procedure

First tweets were fetched from Twitter then the action words were fetched from tweets. Natural Language Processing, or NLP, is a field of study at the intersection of computer science, artificial intelligence, and linguistics. Through NLP, computers are able to extract meaning from natural human language. There are several Natural Language Processing tasks that focus on dissecting and extracting meaning from a particular language attribute. Some examples of NLP tasks include: separating text into sentences, words and morphemes, tagging parts of speech, finding the meaning of each word within a given context, translating text from one human language to another, converting database information into human readable language, answering questions asked in human readable language, analyzing words for sentiment, and converting spoken language into written text. The Natural Language API breaks up text into its constituent words and punctuation (called tokens) and then provides information on each part.
The API can be used to perform the following tasks on a chunk of text. Syntax analysis helps in identifying parts of speech. Entity recognition helps in label entities by type (person, location, and event). Sentiment analysis gets the overall sentiment of a block of text. Content classification classifies documents into predefined categories. Once the action words were pulled from tweets, they were compared with a corpus of sarcasm data using semantic matching and graph-based matching. This provided a score of sarcasm for the given tweet. Using this score, I was able to detect the level of sarcasm in the given tweet. Using the algorithm and the emotion word ontology I processed two types of input data. First, the text data was processed, which was directly given as an input. The input for text was taken from newspaper articles, documents, and files. Second, the tweets were extracted from Twitter using a twitter handler and the tweets were given as an input.

4.3 Machine Learning Models

Four different machine learning (ML) methods which are Naïve Bayesian (NB), Random Forest (RF), Support Vector Machine (SVM) and k-Nearest Neighbor (kNN) algorithms were used in the study. The text was classified into classes by the machine learning based approach using classification techniques. The two broader categorizations of these machine learning techniques are unsupervised and supervised learning. With unsupervised learning, there was no category involved and the targets were not provided by them at all. Thus, clustering was considered to be an important factor here. With supervised learning, the labeled dataset was used to develop this method. When the classification approach was to be designed, the labels were provided to the model. The determination and extraction of particular sets of features such that the sentiments can be detected is the success of both of these learning techniques. Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector machines (SVM) are few amongst the
widely used machine learning techniques for sentiment classification. When having an initial set of labeled opinions is unrealistic for training the classifier, the semi supervised and unsupervised techniques are designed.

4.3.1 SVM (Support Vector Machine)

The support vector machine algorithm is used to find a hyperplane in an N-dimensional space (where N is the number of features) that distinctly classifies the data points.

Figure 1. Hyperplane in SVM.

Huge margin is used for classification through SVM classifier. A hyper plane is used to differentiate the tweets. A discriminative function is utilized by SVM as:

\[ gX = w^T \varphi(X) + b \]

The feature vector denoted in the above equation by "X", weights vector by "w" and the bias vector by "b". The non-linear mapping transformed the information space to high dimensional feature space is denoted by \( \varphi() \). On the training set, w" and "b" were recognized automatically. A linear kernel was applied for classification in this approach (Suanmali, Salim, & Binwahlan)
4.3.2 Maximum Entropy Classifier

With respect to the relationship amongst features, no assumptions were considered in the maximum entropy classifier. The conditional distribution of class labels was estimated by maximizing the entropy of the system through this classifier. The mathematical representation of conditional distribution is:

\[ P(y|X) = \frac{1}{Z(X)} \exp \left( \sum_i \lambda_i f_i(X, y) \right) \]

Here, the feature vector is represented by "X" and the class label by "y". The normalization factor is represented by Z(X) and the weight coefficient by \( \lambda_i \). For classification within our feature vector, the relationships amongst part of speech tag, emotional keyword and negation are utilized.

4.3.3 Naive Bayes

The considerable numbers of features were utilized in the feature vector through Naïve Bayes classifier. Since these features were independent equally, analyzing them exclusively was important. The mathematical representation of conditional probability for Naïve Bayes is given as (Suanmali, Salim, & Binwahlan):

\[ P(X|y_j) = \prod_{i=1}^{m} P(x_i|y_j) \]

A feature vector denoted by "X" is included here which is defined by \( X=\{x_1,x_2,\ldots,x_m\} \). The class label is represented by \( y_j \). The classification of different types of independent features such as positive and negative keywords, emoticons and emotional keywords is done efficiently using Naïve Bayes. The relationships amongst features are not considered in Naïve Bayes classifier. Thus, the relationships which exist among emotional keywords, negation words and speech tags are not utilized in it.
4.4 Natural Language Processing Model

CNNs are effective at modeling hierarchy of local features to learn more global features, which is essential to learn context. Sentences are represented using word vectors (embeddings) and provided as input. Google’s word2vec vectors are employed as input. Non-static representations are used, therefore, parameters for these word vectors are learned during the training phase. Max pooling is then applied to the feature maps to generate features. A fully connected layer is applied followed by a SoftMax layer for outputting the final prediction.

To obtain the other features—sentiment (S), emotion (E), and personality (P)—CNN models are pre-trained and used to extract features from the sarcasm datasets. Different training datasets were used to train each model. (Refer to paper for more details) Two classifiers are tested—a pure CNN classifier (CNN) and CNN-extracted features fed to an SVM classifier (CNN-SVM). A separate baseline classifier (B)—consisting of only the CNN model without the incorporation of the other models (e.g., emotion and sentiment)—is trained as well.
Figure 2. CNN model.
CHAPTER 5

Data Analysis

5.1 Cleaning of Raw Data

Data usually comes from a variety of different sources and is often in a variety of different formats. Hence cleaning raw data is an essential part of preparing a dataset. As, cleaning is not a simple process, as text data often contain redundant and/or repetitive words. This is especially true in Twitter sentiment analysis, so processing text data is the first step towards our solution. The fundamental steps involved in text processing are cleaning of raw data, tokenization, stemming. Some of the standard cleaning steps are lowering case, removal of mentions, removal of special characters, removal of stop words, removal of hyperlinks, removal of numbers, removal of whitespaces. Lowering the case of text is essential because the words, ‘Tweet’, ‘TWEET’, and ‘tweet’ all add the same value to a sentence. Lowering the case of all the words helps to reduce the dimensions by decreasing the size of the vocabulary.

Mentions are very common in tweets. However, as they don’t add value for interpreting the sentiment of a tweet, we can remove them. Mentions always come in the form of ‘@mention’, so we can remove strings that start with ‘@’.

All punctuation marks were also removed. This text processing technique helped to treat words like ‘hurray’ and ‘hurray!’ in the same way.

Stop words are commonly occurring words in a language, such as ‘the’, ‘a’, ‘an’, ‘is’ etc. They were removed because they did not provide any valuable information for the Twitter data analysis.
Tokenization is the process of splitting text into smaller chunks, called tokens. Each token is an input to the machine learning algorithm as a feature. NLTK (Natural Language Toolkit) provides a utility function for tokenizing data.

Stemming is the process of removing and replacing suffixes from a token to obtain the root or base form of the word. This is called a ‘stem’. For example, the stem for the words, ‘satisfied’, ‘satisfaction’, and ‘satisfying’ is ‘satisfy’ and all of these imply the same feeling. Porter stemmer is a widely used stemming technique. nltk. stem provides the utility function to stem ‘PorterStemmer’.

Once the data was preprocessed and cleaned various algorithms were applied to it.

5.2 Feature Engineering

Various features that characterize sarcastic text were extracted and used for classification. When traditional Convolutional Neural Networks (CNN) are used for sarcasm detection, even though lexical features are considered, other kinds of features like sentiment features and incongruity features are ignored. That is why, hand engineering of features are relevant in this scenario. Lexical and pragmatic features as the features used for modeling the baseline system. In addition to those, the proposed system includes two types of incongruity features, sentiment feature and topic feature.

5.2.1 Lexical Features

N-grams are used as a lexical feature and Chi-Squared test is used as the feature selection method for selecting the relevant N-grams. Each tweet is tokenized, stemmed and uncapitalized for extraction unigrams and bigrams from tweets. Then each n-gram is used as a binary feature, where the feature value 1 indicates the presence of a particular n-gram.
An insight obtained from the analysis of the dataset is that the sarcastic text contains a greater number of adverbs and adjectives. Presence of a greater number of adverbs and adjectives shows the richness of text and the chance of it being a highly emotional content. In order to capture this property of sarcastic text, the tweets are Parts-of-speech (POS) tagged using the POS tagger available in Natural Language Toolkit (NLTK) and counted the number of occurrences of adverbs and adjectives. The extracted POS count is used as a numeric feature for classification.

5.2.2 Pragmatic Features

From the statistical analysis of the dataset, it became evident that the sarcastic text contains more number of capital letters than non-sarcastic text. To exploit this characteristics for classification, a numeric feature indicating the presence of capitalization is used as a pragmatic feature. If the number of capital letters in a text is more than the minimum threshold count, then the capitalization feature's value is set as 1, else as 0. Similarly, emoticons are also found more in number in sarcastic tweets compared to non-sarcastic tweets. So a numeric feature indicating the presence of emoticons can also be used for classification.

5.2.3 Incongruity Features

Incongruity arising due to a positive sentiment referring to a negative situation or negative sentiment referring to a positive situation is referred to as context incongruity. This incongruity present in the sarcastic text can be used as a relevant feature since it is the main reason for sarcastic nature of the text. The proposed system exploited two different kinds of context incongruity, explicit and implicit. Explicit incongruity is often expressed by explicitly providing sentiment words of both polarities in the text. Thwarted expectations, a major challenge in sentiment analysis, can also give rise to explicit incongruity. Consider this example
for thwarted expectation: “This phone is good. It is lightweight and affordable. But it’s bad battery life ruins it”. Even though the first two sentences give a very good impression about the phone, the last sentence totally destroys it. Such kind of thwarting in sentiment can be the cause of the sarcastic nature of tweets.

As stated in the work of Cambria et al., polarity lexicons can be used as a baseline approach for counting the positive and negative words for detecting polar words. This approach is used for the detection of explicit incongruity in tweets. In order to capture the thwarted expectation or presence of explicit context incongruity, the system uses a set of features based on polarity lexicon SentiWordNet. They are:

1. Number of Negative and Positive words
2. The length of the Longest continuous positive subsequence
3. The length of the Longest continuous negative subsequence
4. Number of times a negative word is followed by a positive word
5. Number of times a positive word is followed by a negative word

Implicit Incongruity is expressed through phrases of implied sentiment, as opposed to explicitly using polar words (eg: “getting late for work”). In order to find out these phrases a modified version of bootstrapping algorithm proposed by Riloff et al. is used. Mostly a sarcastic text contains a positive word followed by a negative situation. This incongruity is the reason behind the sarcastic property of the text. This important assumption about sarcastic text is the basis of the bootstrapping algorithm. The algorithm proposed by Riloff et al. takes an input seed word, which is a positive polar word and extracts the negative situation phrases which follows the positive word. These extracted negative situation phrases are then given as input to the algorithm and it extracts the positive words and predicates which precedes them. Again, the
negative situation phrases following the extracted positive words and predicates are extracted and fed to the algorithm to generate more positive words and predicates. This whole bootstrapping process is repeated until no new phrases are learned.

But an important point to note is that, sarcasm in a text can also be caused by the incongruity which arises due to a negative phrase referring to a positive situation. So instead of only extracting positive verb phrases and negative situation phrases as proposed by Riloff et al., both the polarities are extracted as proposed by Joshi et al. After extracting the polar verb phrases and situations, they are filtered to make sure that they follow a certain syntactic structure. That is, the phrases that have a predefined structure were selected and to do that POS tagging is used. The phrases of the form infinitive VP, V + PP, V + NP, infinite VP + [ADJ/PRO/N] etc. are selected from the extracted phrases and the ones that do not have a proper phrase structure are filtered. Separate files were maintained to store the extracted phrases of each category. Finally, for an input sentence we check whether it has implicit incongruity with help of these phrases. That is, if a sentence has a positive word followed by a negative situation phrase it is considered as a sarcastic sentence and a binary feature is added to the feature set to use for classification.

5.2.4 Sentiment Features

Incongruity feature considered the sentiment words and phrases in the tweet. But the sentiment of the whole text can also be used as a feature. A major hypothesis is that sarcastic tweets might be extremely negative than non-sarcastic tweets or vice versa. Furthermore, there is frequently a big contrast of sentiment polarity in sarcastic tweets. The tweets often begin with an extreme positive sentiment and concludes with an extreme negative sentiment. For example: “I love being cheated on” starts with a positive polarity and ends in negative polarity. The converse
is also quite frequent. The proposed system uses SentiWordNet, SenticNet and Vader tool in NLTK for sentiment analysis. SenticNet is a lexical resource made by using sentic computing and it contains polarity scores of Uni-grams, Bi-grams, Tri-grams and Four-grams unlike SentiWordNet which have only Unigram polarities.

The polarity of the whole text and its sub-parts are calculated to find out if there is any contrast in polarity. First step is to calculate the polarity of the whole text and the resulting polarity is used as the first sentiment feature. It is a string feature stating whether the overall polarity is positive, negative or neutral. Then the text is split into two parts and sentiment analysis is done for both parts to find their polarity. A binary feature is added to the feature set that takes the value 1, if there is a polarity inversion from positive to negative or negative to positive between the two sub-parts. The same procedure is repeated for three-part splitting and checked whether there is an inversion of polarity between any two subparts. This whole process is done by using SentiWordNet, SenticNet and Vader tool and the majority result among these three is used as the value of features. Apart from explicit incongruity feature, sentiment feature also helps to detect any contrast in polarity, since it considers the sentiment of subparts along with the overall sentiment of the text.

5.2.5 Topic Feature

Some words are frequently grouped together in the same tweets (Eg: Saturday, party, friends, night etc). These groups of words are called topics. Certain kinds of topics are more associated with sarcasm in a domain and learning them and making use of them as a feature in supervised learning is simpler and more precise. Topic modeling is used to find the topics associated with tweets. The system uses the Python library gensim which implements topic modeling using latent Dirichlet allocation (LDA). The topic modeler is capable of learning the
important topics from the input tweets. From the training set of tweets, the topic modeler learns a group of topics and keeps it in the topic dictionary with a unique number for each topic. Later the input text is broken down into a sum of topics and the corresponding topic numbers are picked from the topic dictionary and they are added as numeric features into the feature set.

### 5.3 Key Performance Indicators Calculation

Key performance indicators (KPIs) were used to evaluate the classifiers. These KPIs included accuracy, precision, recall, and the F1 score, which are the most common ratings used in classification problems. Each of these KPIs are described in greater detail below.

Accuracy shows the overall accuracy of the instances which are correctly classified to the total number of the instances. It is calculated using the following formula:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

In this formula, TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative.

Precision shows the percentage of relevant searched sarcastic tweets. It measures the number of tweets categorized as sarcasm against the total amount of tweets classified as sarcasm. It is calculated using the following formula:

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

Recall represents the percentage of relevant sarcastic tweets that have been searched. It is calculated as the ratio of the number of true positives (TP) and the sum of true positives and negative positives (TN). The following formula was used to calculate this metric:

\[
\text{Recall} = \frac{TP}{TP + FN}
\]
Finally, the F1 score is a measure of accuracy that can be interpreted as a weighted average of accuracy and recall. It is calculated using the following formula:

\[
F1 \text{ score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]
CHAPTER 6

Results

Table 1 provides a summary of the results of the KPIs for the three classifiers utilized.

The results showed that ML Random Forest obtained 80 percent accuracy and the F1 score was 78 percent. ML SVM presented 77.99 percent precision and a low F1 score of 68 percent. CNN can detect sarcasm with high precision, and can use output to narrow down sentiment analysis, but the actual results of all sarcastic tweets can be very different. The NLP classifiers presented the highest accuracy and F1 scores as compared to SVM and Random forest classifiers.

Table 1. Comparison of Machine Learning and Natural Language Processing Classifiers.

<table>
<thead>
<tr>
<th></th>
<th>ML(SVM)</th>
<th>ML(Random Forest)</th>
<th>NLP(CNN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>74</td>
<td>80</td>
<td>85</td>
</tr>
<tr>
<td>Precision</td>
<td>77.99</td>
<td>78</td>
<td>80</td>
</tr>
<tr>
<td>Recall</td>
<td>63</td>
<td>72</td>
<td>76</td>
</tr>
<tr>
<td>F-1 score</td>
<td>68</td>
<td>78</td>
<td>83</td>
</tr>
</tbody>
</table>

Note. Values in the table represent percentages.
CHAPTER 7

Discussion

Sarcasm detection is known as the ‘Achilles heel of sentiment analysis. It is the most difficult area in sentiment analysis. There are three main challenges in sarcasm detection. First, it is an easier task to detect sarcasm from speech when it is compared to the sarcasm detection from text. Because, the use of a certain tone of speech, body language, and facial expression can be useful while identifying sarcasm from speech. Second, the quality of the data set is also a crucial factor in sarcasm detection. The general nature of sarcastic sentences are ambiguous and doubtful. The presence of hashtags which indicates the sarcasm solves this ambiguity. But without hashtags, sarcastic sentences are complicated to understand. Third, feature selection is another important task in sarcasm detection. So, introducing new features and using them with already existing features can increase the accuracy of sarcasm detection. Selecting an appropriate new feature should involve deeper study about semantic, punctuation-based and hyperbole features.

7.1 Summary of Results

In this work, a system was proposed which detects sarcasm on tweets on Twitter and compared the accuracy of Machine Learning algorithms and Natural Language Processing algorithms. Sarcasm was very dependent and highly contextual; therefore, sentiment and other contextual clues are needed to help detect the sarcasm in text. The current study used sarcastic tweets, 10,000 tweets containing #sarcasm, and #not dataset. The system used the SVM, CNN, and Random forest classifier. In support of the hypothesis, the results showed that CNN had more accuracy than other classifiers, that is NLP outperformed Machine learning in accurately classifying sarcastic texts.


7.2 Limitations and Future Research

Even though this study has multiple strengths it also has some limitations. These limitations provide opportunities for future research. A limitation of this study was that all patterns for sarcastic detection are not covered in the extracted patterns. A recommendation for future research is to combine Neural Network, Genetic Algorithm and Pattern-based approaches for more accuracy.

Another limitation is that the complexity of sarcasm makes it challenging to detect. This, like most of the previous sarcasm detection research, has been done in the English language. Future work should therefore explore how to detect sarcasm in other languages.

Another recommendation for future research is to improve the accuracy and evaluation of these classifiers by including new data sets and new feature sets. Such future work should use different deep learning methods and inspect more conceptual based features. The aim would be to increase the performance of the sarcasm detection model based on f-score, accuracy, recall, and precision. Researchers can also improve accuracy by focusing on hyperbole features and explore syntactic dependencies in text. Researchers should also go beyond text and use multimodal sarcasm detection with images and audio too to improve accuracy. Training the models on data from TV shows like The Big Bang Theory and Friends provide rich sources of data for this analysis. This type of evaluation is still in the early stages but considering the multimodality of sarcasm, it looks promising because many times sarcasm is not in the text, but is more easily detected in the intonation or face expression of the speaker.
REFERENCES


https://doi.org/10.1109/ICISET.2016.7856499

