Sarcasm detection using machine learning algorithms in Twitter: A systematic review

Sarsam, S. M., Al-Samarraie, H., Ibrahim Alzahrani, A. & Wright, B.

DOI 10.1177/1470785320921779
ISSN 1470-7853

Publisher: SAGE Publications

Sarcasm detection using machine learning algorithms in Twitter: A systematic review

Abstract

Recognizing both literal and figurative meanings is crucial to understanding users' opinions on various topics or events in social media. Detecting the sarcastic posts on social media has received much attention recently, particularly because sarcastic comments in the form of tweets often include positive words that represent negative or undesirable characteristics. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) statement was used to understand the application of different machine learning algorithms for sarcasm detection in Twitter. Extensive database searching led to the inclusion of 31 studies classified into two groups: Adapted Machine Learning Algorithms (AMLA) and Customized Machine Learning Algorithms (CMLA). The review results revealed that Support Vector Machine (SVM) was the best and the most commonly used AMLA for sarcasm detection in Twitter. In addition, combining Convolutional Neural Network (CNN) and SVM was found to offer a high prediction accuracy. Moreover, our result showed that using lexical, pragmatic, frequency, and part-of-speech tagging can contribute to the performance of SVM, while both lexical and personal features can enhance the performance of CNN-SVM. This work also addressed the main challenges faced by prior scholars when predicting sarcastic tweets. Such knowledge can be useful for future researchers or machine learning developers to consider the major issues of classifying sarcastic posts in social media.

Keywords: sarcasm detection; machine learning algorithms; twitter, trolling

Introduction

Microblogging platforms are the main mediums for a person to express his/her views, thoughts, and opinions on various topics and events. Sarcasm is a sophisticated form of irony that is commonly found in social networks and microblogging websites, as these platforms often encourage trolling and/ or criticism of others. There is a slight difference between irony and sarcasm (Reyes, Rosso, & Buscaldi, 2012). Sarcasm, as a term, is commonly used to describe an expression of verbal irony (Colston, 2000). It is combined with certain types of irony such as jocularity, hyperbole, rhetorical questions, and understatement (Gibbs, 2000). Kumon-Nakamura, Glucksberg, and Brown (2007) referred to sarcastic irony as an opposite term to the non-sarcastic one. Gibbs Jr and Colston (2007) suggested that irony is often compared to satire and parody. Thus, to characterize sarcasm on Twitter, Parmar, Limbasiya, and Dhamecha (2018) suggested the following: 1) conflict between negative situation and positive sentiment, 2) conflict between positive situation and negative sentiment, 3) Tweet starts with an interjection word, 4) likes and dislikes contradiction, 5) tweet conflicting ubiquitous facts, 6) tweet contains positive sentiment with antonym pair, and 7) tweet conflicting facts that are time sensitive. With a large volume of content being produced on social media and the need to analyze it closely, text classification methods have been introduced to deal with this sophisticated emergent.

In text classification, sarcasm detection is an essential tool that has many implications for several areas including security, health, and sales (Jain & Hsu, 2015). With the help of sarcasm detection techniques, companies can analyze customers' feelings about their products. This provides crucial help for those companies to boost their product quality (Saha, Yadav, & Ranjan, 2017). In sentiment analysis, the sarcasm classification is an essential subtask (Cambria, Poria, Bisio, Bajpai, & Chaturvedi, 2015), especially in classifying tweets, for conveying implicit

information within the message that a person expresses or shares with others. In addition, the structure of the tweet may also be used to predict sarcasm (e.g., transforming the polarity of a positive/a negative statement into its opposite form). On Twitter, there are several issues that make sarcasm detection a difficult task. Parmar et al. (2018) listed some of the existing challenges in classifying sarcastic tweets. These challenges are: 1) the nature of the collected tweets (e.g., Twitter limits 280 characters for posting tweets which may lead to more ambiguity), 2) the collected tweets contain several uncommon words, slang, abbreviations that are of a more informal nature, and 3) there is no predefined structure for sarcasm recognition in Twitter. Consequently, previous studies have applied machine learning techniques in order to predict sarcasm in tweets (Jain & Hsu, 2015). For instance, Altrabsheh, Cocea, and Fallahkhair (2015) examined several machine learning techniques, features, and preprocessing levels to recognize sarcasm from students' feedback collected via Twitter. In order to detect the sarcasm, Altrabsheh et al. (2015) compared several classifiers that were recommended by Tian et al. (2014). The result showed that Complement Naive Bayes (CNB) had the highest recall function. Ren, Ji, and Ren (2018) proposed two different context-augmented neural models to be used for sarcasm detection. Prasad, Sanjana, Bhat, and Harish (2017) compared numerous classification algorithms in which they found that Gradient Boost had the best performance with prediction accuracy. Tungthamthiti, Shirai, and Mohd (2016) proposed a novel approach for recognizing sarcasm in tweets through combining two classification algorithms (Support Vector Machine (SVM) with N-gram featurse and SVM). Based on these, it can be said that the performance of classifiers is important for accurate prediction of sarcasm when processing expressions in the textual data. In addition, the type of classifiers seems to play a key role in sarcasm detection. However, in Twitter, limited studies have addressed the efficiency of sarcasm detection

algorithms with regards to the utilized features. Therefore, this study reviewed the major sarcasm classifiers, their classification performance, and the features contributed to such performance. This study also explored the challenges faced by prior scholars when attempting to detect sarcastic tweets. Outcomes from this review offer practical implications and recommendations for future scholars about the types of machine learning algorithms and the main features used in the detection of the sarcastic tweets.

Sarcasm detection advantages and implications

Sarcasm is largely used in social networks and microblogging websites, where people mock or criticize in a way that makes it difficult even for humans to tell if what is said is what is meant. The figurative nature of sarcasm makes it an often-quoted challenge for sentiment analysis (Liu, 2010; Liu et al., 2014). It has an implied negative sentiment, but a positive surface sentiment. The challenges of sarcasm and the benefit of sarcasm detection to sentiment analysis have led to an interest in automatic sarcasm detection as a research problem. Automatic sarcasm detection refers to computational approaches that predict if a given text is sarcastic (Joshi, Bhattacharyya, & Carman, 2017). This motivated several scholars to apply sarcasm detection in several important domains. For instance, sarcasm detection it can be applied in a culture-related field. A study by Joshi, Bhattacharyya, Carman, Saraswati, and Shukla (2016) explored the aspects that are influencing the prediction quality of sarcastic statements. The researcher believes that such an approach would contribute positively to judging the quality of new datasets. Another work by Kannangara (2018) applied sarcasm detection in classifying people's opinions in politics. In this context, the researchers proposed three models for socio-political opinion polarity classification of microblog posts. Also, the researchers proposed a novel sarcasm detection

model that uses ideology and fine-grained opinion as features with other linguistic features to classify sarcastic opinions. Besides the political implementation of sarcasm detection, it has a strong implementation in the industry by taking advantage of social media platform. These platforms get evolved into large ecosystems that allow users to present their opinions freely. Therefore, companies leverage this ecosystem in order to access major public opinion about aspects related to products, services, and to provide real-time customer assistance. Moreover, these companies have a strong social media presence with an active team for marketing and customer assistance purposes (Rajadesingan, Zafarani, & Liu, 2015). This produces a huge volume of information that is available on social media websites which allows such companies to rely on tools like HootSuite to perform several complicated tasks, including content management, sentiment analysis, and extraction of relevant messages for the company's customer service representatives to respond to. Unfortunately, these tools lack the sophistication to decode nuanced forms of language such as sarcasm that carry indirect messages (Rajadesingan et al., 2015). Hence, with the detection of sarcastic statements, people's emotion can be clearly recognized (Kuo, Alvarado, & Chen, 2018).

Method

This review was planned, conducted, and reported in adherence to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) statement (Moher, Liberati, Tetzlaff, & Altman, 2009). It provides a detailed guideline of the preferred reporting style for systematic reviews and meta-analyses. This review was guided by the following questions: "What are the key data mining algorithms used for sarcasm detection in Twitter?" and "What are the key sarcastic features used to detect sarcastic tweets?".

Search strategy (Identification of studies)

Electronic databases (EMBASE, PubMed, PsycINFO, Web of Science, Scopus, Cochrane Central Register of Controlled Trials (CENTRAL), and Google Scholar) were used to search and retrieve published studies on the role of machine learning algorithms in sarcasm detection. The retrieval process was handled independently by the researchers in which title and abstract screening were conducted in the initial search. Disagreements on the eligibility of the included studies were resolved through discussion between the researchers. However, in the event a resolution was not possible, a third review was consulted. Multiple studies adapted different machine learning algorithms or techniques were grouped under the Adapted Machine Learning Algorithms (AMLA) category. Other studies that developed or semi-adapted AMLA were grouped under the Customized Machine Learning Algorithms (CMLA).

Data collection and extraction

The searched terms used in this study were: [sarcasm detection in social media] or [sarcasm detection in Twitter] or [sarcasm detection]). No beginning date cutoff was used, and the last date of search was performed in November 2018. This search was supported with the entire reference lists from several published work about sarcasm detection in Twitter during the period of 2010 and 2018. The following data were extracted from each study: a) labeling approach, b) machine learning algorithm, c) evaluation metric, and d) challenges.

Inclusion criteria

The abstract of each retrieved article was sorted by study type AMLA or CMLA. Our review and coding of the abstracts led to the identification of 576 potential articles. Two readers reviewed and coded each of the 576 articles using the coding scheme. The inter-coder reliability

for each article was checked based on the coding scheme and evaluation presented in Table 1. Chance-adjusted interrater agreement for study inclusion, determined using the intraclass correlation coefficient (ICC) (Shrout & Fleiss, 1979), was 0.81.

Table 1: Article coding scheme and evaluation

Code	Classification	Description
S	Study purpose	Studies focusing on sarcasm detection
P	Platform	Twitter platform
C	Class labeling	The labeling mechanism of the target features (i.e., class) such as polarity
M	Machine learning algorithm	The type of machine learning algorithm used (supervised or unsupervised or semi-supervised)
E	Evaluation metrics	The utilized metrics to evaluate the performance of the classifier
L	Language	The language of the study and the data were in English

Figure 1 shows the selection process of previous studies based on the PRISMA guidelines. Our search of the literature resulted in 3,282 potentially relevant articles. Additional 27 studies were identified from other sources. Duplicates were removed, and abstracts from the remaining 1,641 publications were screened. Initially, non-English articles, articles with limited focus on sarcasm detection, and review articles were excluded (N =1020). The remaining 621 articles were selected for further screening based on the inclusion scheme identified above. After removing potential duplicates and assessing each article based on the inclusion criteria above, 31 articles were used to answer the research questions of this review.

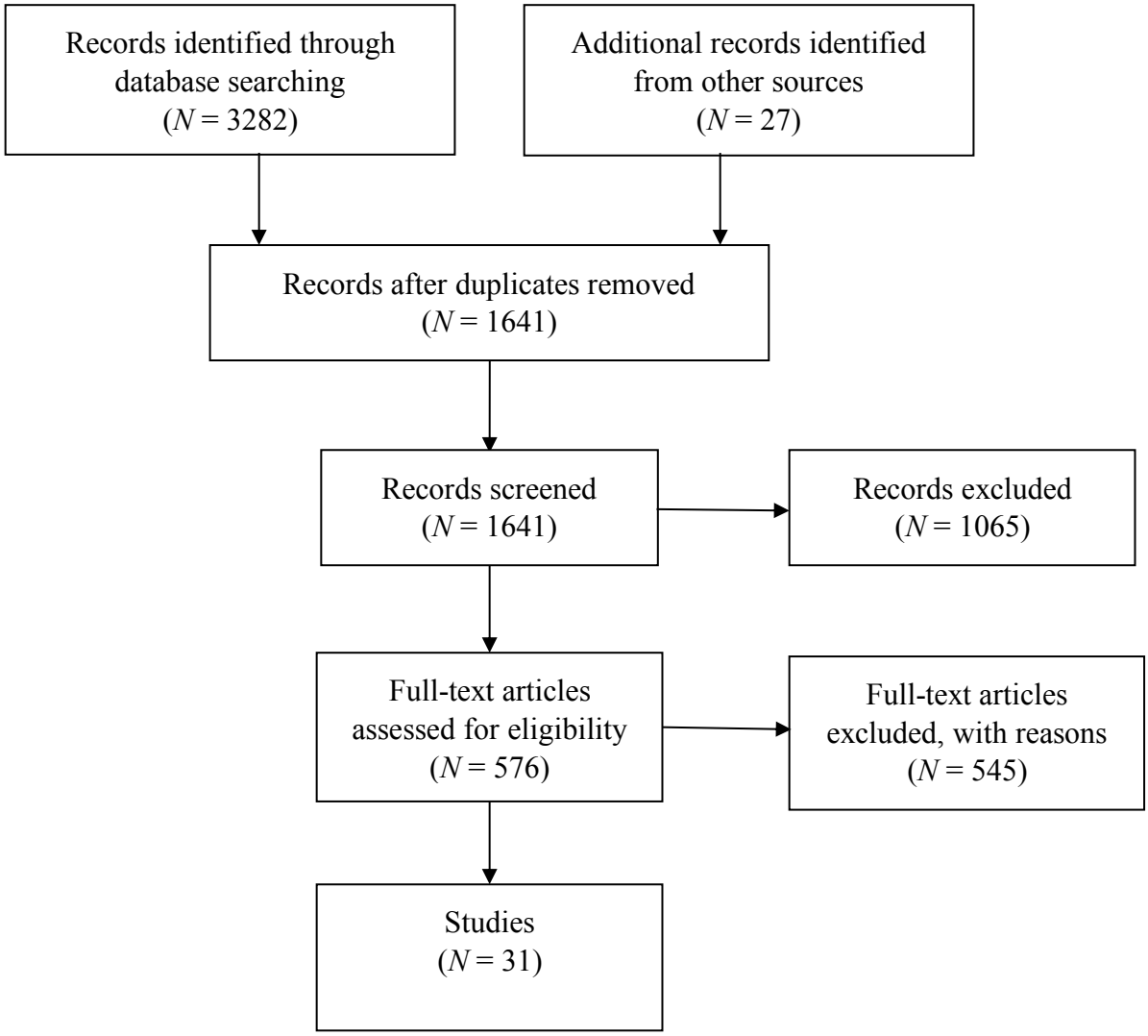

Figure 1: Studies selection process

Results

Our result of the 31studies on sarcasm detection algorithms are summarized in Table 2. This result showed that the percentage of utilizing the algorithms varied between two groups (AMLA and CMLA). In the AMLA group, Support Vector Machine (SVM) was found to be the most frequent algorithm (22.58 %) followed by Logistic Regression Method (19.35 %), Naïve Bayes (9.67 %), and Random Forest (6.45 %), respectively (see Figure 2). However, algorithms in the CMLA group were found to be less frequently used (3.22 %) for detecting the sarcastic

tweets. Hence, the AMLA group consisted of the algorithm that was most frequently used in sarcasm detection tasks, while the CMLA contains the algorithms that are less frequently used in such tasks.

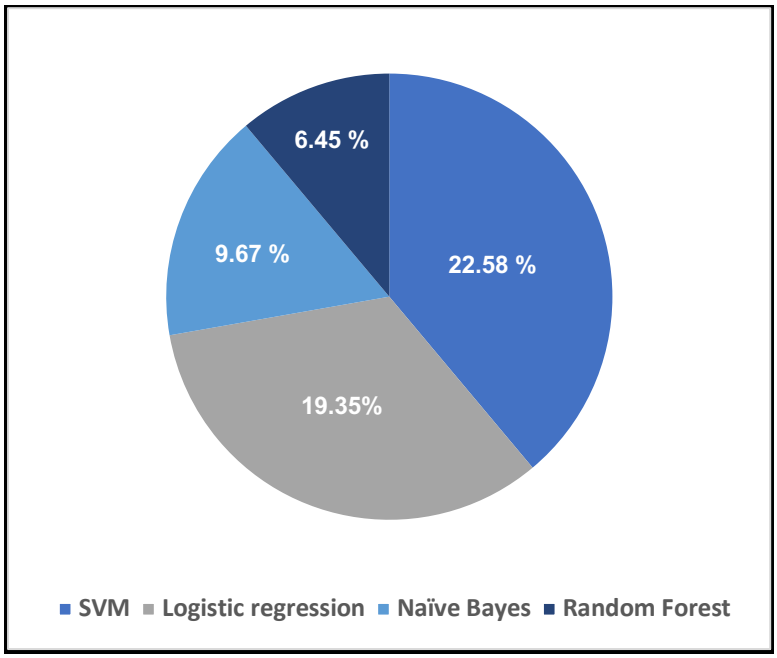

Figure 2: The frequency of AMLA's group algorithms

Table 2: Utilization frequency of machine learning algorithms

No.	Author(s)	Labeling approach	Algorithm	Category
1.	Davidov, Tsur, and Rappoport (2010)	Labeling based on discrete range of 1 to 5 where 5 indicates a clearly sarcastic sentence and 1 indicates a clear absence of sarcasm.	Semi-supervised sarcasm identification algorithm (SASI)	CMLA
2.	González-Ibánez, Muresan, and Wacholder (2011)	Sarcastic (S), positive (P) and negative (N)	SVM	AMLA
3.	Riloff et al. (2013)	Polarity: positive and negative tweets.	Bootstrapping	CMLA
4.	Kovaz, Kreuz, and Riordan (2013)	Sarcastic and non-sarcastic.	Logistic regression	AMLA
5.	Ptáček, Habernal, and Hong (2014)	Sarcastic and non-sarcastic.	Maximum Entropy (MaxEnt)	CMLA

6.	Tungthamthiti, Kiyoaki, and Mohd (2014)	Sarcastic and non-sarcastic (when a tweet contains contradiction of sentiment polarity without coherence between them, it could be regarded as non-sarcastic tweet).	SVM	AMLA
7.	Rajadesingan, Zafarani, and Liu (2015)	Polarity: positive and negative tweets (in particular sarcastic or non-sarcastic).	Sarcasm Classification Using a Behavioral modeling Approach (SCUBA)	CMLA
8.	Bouazizi and Ohtsuki (2015)	Polarity: positive and negative tweets	SVM	AMLA
9.	Joshi, Sharma, and Bhattacharyya (2015)	Sarcastic or non-sarcastic.	LibSVM with RBF kernel	CMLA
10.	Ghosh, Guo, and Muresan (2015)	Positive or negative tweets.	SVM	AMLA
11.	Cerezo-Costas and Celix-Salgado (2015)	Polarity: positive, negative, and neutral.	Logistic regression	AMLA
12.	Bamman and Smith (2015)	Sarcastic and non-sarcastic.	Logistic regression	AMLA
13.	Barbieri, Ronzano, and Saggion (2015)	Polarity: positive, negative, and neutral.	SVM	AMLA
14.	Signhaniya, Shenoy, and Kondekar (2015)	Sarcastic and non-sarcastic.	SVM	AMLA
15.	Jain and Hsu (2015)	Sarcastic and non-sarcastic.	Logistic regression	AMLA
16.	Bouazizi and Ohtsuki (2016)	Polarity: positive and negative tweets.	Random Forest	AMLA
17.	Amir, Wallace, Lyu, and Silva (2016)	Sarcastic and non-sarcastic.	Content and User Embedding Convolutional Neural Network (CUE-CNN)	CMLA
18.	Zhang, Zhang, and Fu (2016)	Sarcastic and non-sarcastic.	Gated recurrent neural network (GRNN)	CMLA
19.	Poria, Cambria, Hazarika, and Vij (2016)	Sarcastic and non-sarcastic.	CNN-SVM	CMLA
20.	Abercrombie and Hovy (2016)	Sarcastic and non-sarcastic	Logistic regression	AMLA
21.	Ghosh and Veale (2016)	Sarcastic and non-sarcastic	A combination of CNN; Long short-term memory	CMLA

			(LSTM) network; and Deep neural network (DNN)	
22.	Bali and Singh (2016)	Sarcastic and non-sarcastic.	Logistic regression	AMLA
23.	Tungthamthiti, Shirai, and Mohd (2016)	Sarcastic and non-sarcastic.	SVM	AMLA
24.	Saha et al. (2017)	Polarity: positive, negative or neutral.	Naïve Bayes	AMLA
25.	Prasad, Sanjana, Bhat, and Harish (2017)	Sarcastic or non-sarcastic.	Gradient Boost	CMLA
26.	Tay, Tuan, Hui, and Su (2018)	True and false.	Multi-dimensional Intra-Attention Recurrent Network (MIARN)	CMLA
27.	Ren, Ji, and Ren (2018)	Polarity: positive and negative tweets.	MODEL-KEY	CMLA
28.	Parmar et al. (2018)	Polarity: positive and negative tweets.	Feature based Composite Approach (FBCA)	CMLA
29.	Das, Kadam, Kalra, Nayak, and Govilkar (2018)	Labels are two types 0 and 1 indicating the sentence being not sarcastic and sarcastic respectively.	Naïve Bayes	AMLA
30.	Parde and Nielsen (2018)	Polarity: positive and negative tweets.	Naïve Bayes	AMLA
31.	Bouazizi and Ohtsuki (2018)	Polarity: positive, negative or neutral.	Random Forest	AMLA

AMLA

The major machine learning algorithms, such as SVM, Logistic Regression, Naïve Bays, and Random Forest that were utilized frequently for predicting sarcasm on the Twitter platform (see Figure 3). The following subsection describes each algorithm.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is the most commonly used algorithm found in the literature for detecting sarcasm on Twitter, particularly due to its efficiency in facilitating the process of sentiment classification (González-Ibánez et al., 2011). Several labeling methods were

also found to be used in labeling the collected tweets before passing them to the SVM classifier. For example, the polarity method which is a natural language processing (NLP) application was used to categorize text sentiment, thus making it suitable for sarcasm detection. This includes detecting potential changes in users' emotion based on the characteristics of their tweets. Bouazizi and Ohtsuki (2015) used the polarity method to label each either positive or negative type. Barbieri et al. (2015) also adopted the polarity technique and added an additional label 'neutral' to the detection process. González-Ibánez et al. (2011) applied polarity by labeling the tweets as sarcastic, positive and negative, while Ghosh, Guo, and Muresan (2015) labeled their data as sarcastic sense and literal sense from a negative and positive perspective. Other previous studies, such as Tungthamthiti, Kiyoaki, and Mohd (2014), Signhaniya, Shenoy, and Kondekar (2015), and Tungthamthiti, and Shirai, and Mohd (2016), labeled tweets either sarcastic or non-sarcastic. This generic approach was found to be useful when detecting the full statement/tweets by considering aspects related to "coherence" (e.g., the relationships across multiple sentences). The general concept behind using this approach is that the sarcastic tweets should contain expressions which clearly indicate the references to some words across the tweets. When a tweet contains contradiction of sentiment polarity without coherence between them, it could be regarded as non-sarcastic tweet. The performance results from using the SVM algorithm ranged between 50.93% and 91.8 %.

Our review also revealed some key challenges when attempting to detect sarcastic tweets using the SVM classifier. For example, González-Ibánez et al. (2011) found that distinguishing sarcastic from non-sarcastic tweets to be one of the challenges. Ghosh et al. (2015) stated that identifying the sense of a target word using SVM is a hard task. This is because all the tweets L_{sent} sense are collected using sentiment hashtags (small pieces of text which usually contain

valuable information to extract the sense of a whole sentence) to which they might be lexically more similar to the sense tweets than the literal tweets (Ghosh et al., 2015).

Logistic regression

Logistic regression was also used to detect sarcasm on Twitter mainly because it is mathematically related to linear discriminant analysis (Cliche, 2014). When data or tweets labeling takes place, a straightforward labeling method such as sarcastic and non-sarcastic can be used to effectively facilitate the detection of sarcasm (Abercrombie & Hovy, 2016; Bali & Singh, 2016; Jain & Hsu; Kovaz et al., 2013). Cerezo-Costas and Celix-Salgado (2015) adopted a special method to label their tweets; this method called the "Conditional Random Fields (CRFs)". The authors utilized CRFs to obtain the scope of polarity modifiers and shifters (e.g. negation and intensification), thus labelling tweets as positive, negative, and neutral. Using this algorithm for sarcasm detection may offer a performance result ranging between 59.11 % and 80.27 %.

Naïve Bayes

Many previous studies relied on the application of Naïve Bayes for various detection purposes. In the context of this study, Naïve Bayes was commonly used to classify tweets' polarity (positive and negative) (Parde & Nielsen, 2018). Another way of labeling was established by Das, Kadam, Kalra, Nayak, and Govilkar (2018) who utilized two types of labels (0 and 1) by indicating whether the tweet is sarcastic or not sarcastic. Our review of the literature also showed that the performance of the Naïve Bayes algorithm ranged between 59 % and 67.81 %. According to Li, Fong, Zhuang, and Khoury (2016), Naïve Bayes does not perform well

when TF-IDF is applied. This can be due to the fact that Naive Bayes does not utilize search for predicting the membership probabilities for each class.

Random Forest

The Random Forest classifier is known to reduce bias due to overfitting and class imbalance between tweets. The polarity approach was the most common approach to labeling sarcastic tweets. Bouazizi and Ohtsuki (2016) used this algorithm to label their data either positive or negative. Later work by Bouazizi and Ohtsuki (2018) used the polarity method for labelling tweets into positive, negative, or neutral. The performance reported by previous studies ranged between 45.9% and 83.1%.

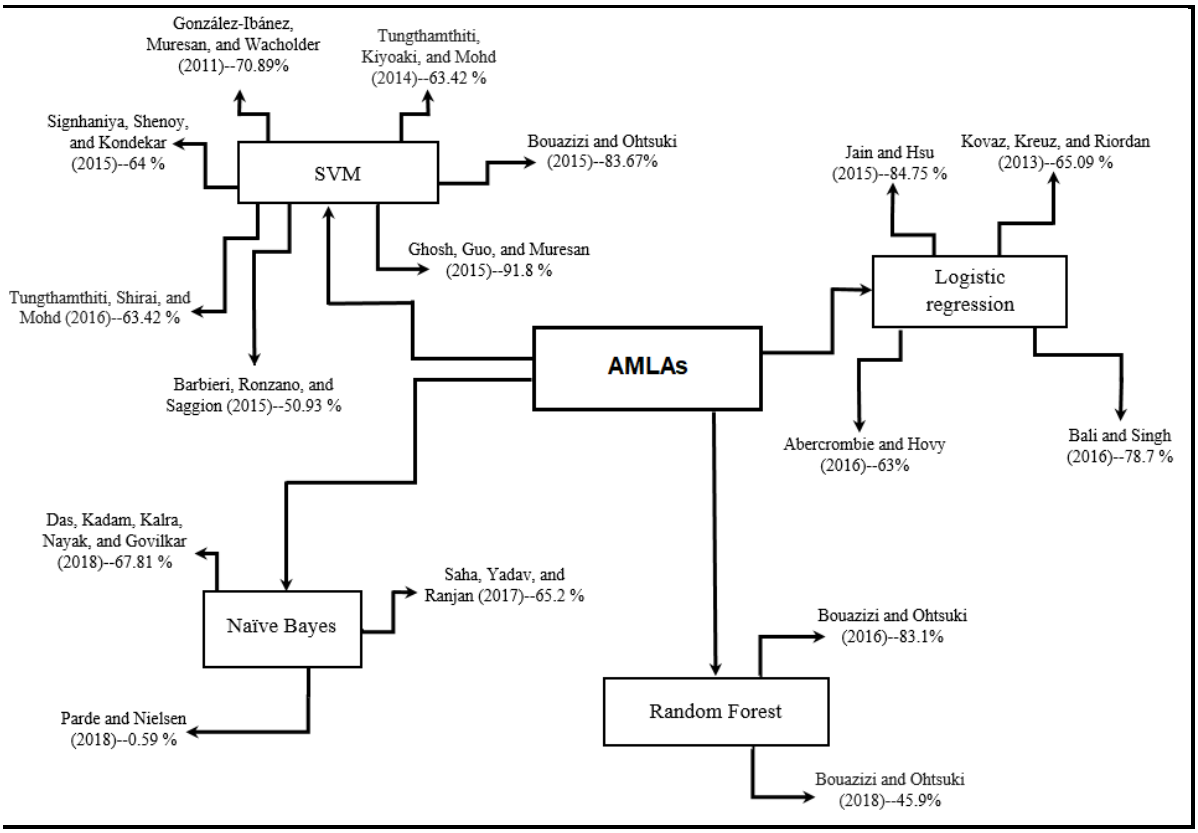

Figure 3: The classification of AMLAs for sarcasm detection

CMLA

A number of scholars have made some extensions to the standard machine learning algorithms in an attempt to provide a customized classification of sarcasm (see Figure 4). For example, Davidov, Tsur, and Rappoport (2010) used a Semi-supervised Algorithm for Sarcasm Identification (SASI) to detect sarcasm on Twitter. The algorithm consisted of two modules: semi-supervised pattern acquisition for identifying sarcastic patterns that serve as features for a classifier, and a classifier that classifies each sentence to a sarcastic class. The process of labeling the data was based on a discrete range of 1 to 5 where 5 indicates a clearly sarcastic sentence and 1 indicates a clear absence of sarcasm. At the classification stage, SASIs demonstrated excellent performance of 94.7%.

Riloff et al. (2013) developed a sarcasm recognizer approach by presenting a novel bootstrapping algorithm. This method automatically processes the list of positive and negative phrases from sarcastic tweets. The proposed algorithm was able to identify just one type of sarcasm that is common in tweets, i.e., contrast between a positive sentiment and negative situation. This is because a common form of sarcastic tweets contains a positive sentiment contrasted with a negative situation. Riloff et al. (2013) also showed that identifying contrasting contexts using the phrases learned through bootstrapping may potentially improve the overall classification performance to 78%.

Maximum Entropy (MaxEnt) was used by Ptáček et al. (2014) to characterize sarcasm in two languages (English and Czech). Sarcastic and non-sarcastic were the main labels used by Ptáček et al. (2014) to perform the sarcasm detection process. The performance result of MaxEnt was 94.7 %. This high performance can be attributed to its structure which consists of character n-gram, skip-bigram, and pointedness.

In order to provide automatic sarcasm detection, Rajadesingan et al. (2015) used lexical and linguistic cues to address the difficulty of the sarcasm classification task on Twitter through leveraging certain behavioral traits that are perceived to be intrinsic to users expressing sarcasm. Using this method, the authors were able to recognize some traits by comparing between tweets posted by the user through Sarcasm Classification Using a Behavioral modeling Approach (SCUBA). The SCUBA considers psychological and behavioral aspects of sarcasm and leveraging users' historical information to know whether tweets are sarcastic or not expressing sarcasm. The evaluation of the detection process exhibited that SCUBA was able to achieve a high-performance result of 92.94%.

Another study by Joshi et al. (2015) presented a computational system that harnesses context incongruity to perform sarcasm detection on Twitter. They used LibSVM with Radial Basis Functions (RBF) kernel in order to label their tweets as sarcastic and non-sarcastic. The authors reported that LibSVM achieved high performance (88 %). However, they addressed some errors when applying LibSVM; these were subjective polarity, system incongruity is expressed outside the text, incongruity due to numbers, dataset granularity, and implicit incongruity.

Amir et al. (2016) introduced a deep neural network technique to accomplish an automated sarcasm detection task. They proposed a method to automatically perform a learning process and then exploit user embeddings with lexical signals to recognize sarcasm. The authors labeled the data as sarcastic and non-sarcastic when Content and User Embedding Convolutional Neural Network (CUE-CNN) algorithm was used. The performance result was 87 %, particularly due to the efficiency of CUE-CNN in inducing vector lexical representations using a convolutional layer.

Zhang et al. (2016) investigated the use of neural network for tweet sarcasm detection. Sarcastic and non-sarcastic were used as the main labels. Bi-directional Gated Recurrent Neural Network (GRNN) was invoked to detect the sarcasm with a performance result of 90.74 %. This can be due to the nature of the utilized approach, which leveraged the distributed embedding inputs and recurrent neural networks to induce semantic features.

Poria et al. (2016) developed several models based on a pre-trained convolutional neural network in order to extract sentiment, emotion and personality features for the purpose of sarcasm detection. The developed algorithm combines Convolutional Neural Networks (CNN) and SVM (CNN-SVM). The results showed extremely high performance for CNN-SVM (97.71 %).

Ghosh and Veale (2016) proposed a neural network semantic model for sarcasm detection as an attempt to solve grammatical inaccuracy problems. The proposed neural network model of Convolution Neural Network (CNN), Long Short-Term Memory (LSTM), and Deep Neural Network (DNN) showed a performance result of 92%.

It is also worth mentioning that sarcastic tweets can mislead data mining activities and result in wrong classification. Prasad et al. (2017) compared several classification algorithms, such as Random Forest, Gradient Boosting, Decision Tree, Adaptive Boost, Logistic Regression and Gaussian Naïve Bayes, in order to classify sarcastic tweets. The dataset used two labels (sarcastic and non-sarcastic), and the result showed that the best performing classifier was the Gradient boosting classifier (81.82%). It was assumed that such result can be due to the ability of Gradient boosting to optimize the cost function which aids in the classification of the dataset by continuously assessing the features and modifies the classifier to avoid wrong classification.

Tay et al. (2018) proposed an attention-based neural model to explicitly model contrast and incongruity. The proposed technique called Multi-dimensional Intra-Attention Recurrent Network (MIARN) was designed based on the intuition of compositional learning through leveraging intra-sentence relationships. The result of the MIARN achieved high performance of 86.47%.

Ren, Ji, and Ren (2018) pointed out that existing detection techniques have two limitations when detecting sarcasm on Twitter: they rely on discrete models and require a large number of manual features. The authors explored the use of neural network models for classifying the sarcastic tweets using Model-Key (local). This technique depends on the convolutional neural network using two self-developed context-augmented neural models for the sarcasm detection task. Three labels (negative, sarcastic, and positive) were applied and the result showed that the proposed context-augmented neural models can successfully decode sarcastic clues from contextual information and provide a relative improvement in the detection performance (63.28 %).

Sarcasm sentiment tweets are used for taunting, insulting or to make fun of someone. This led Parmar et al. (2018) to consider the potential of live tweets, using a hybrid approach, in processing lexical and hyperbole features, as well as improving the overall performance result. The proposed algorithm was called Feature-based Composite Approach (FBCA). Two lexical and hyperbole features of composite and mapreduce were used to reduce the execution time. The classification result of FBCA achieved a high performance (90%) when predicting whether the live tweets were sarcastic or not.

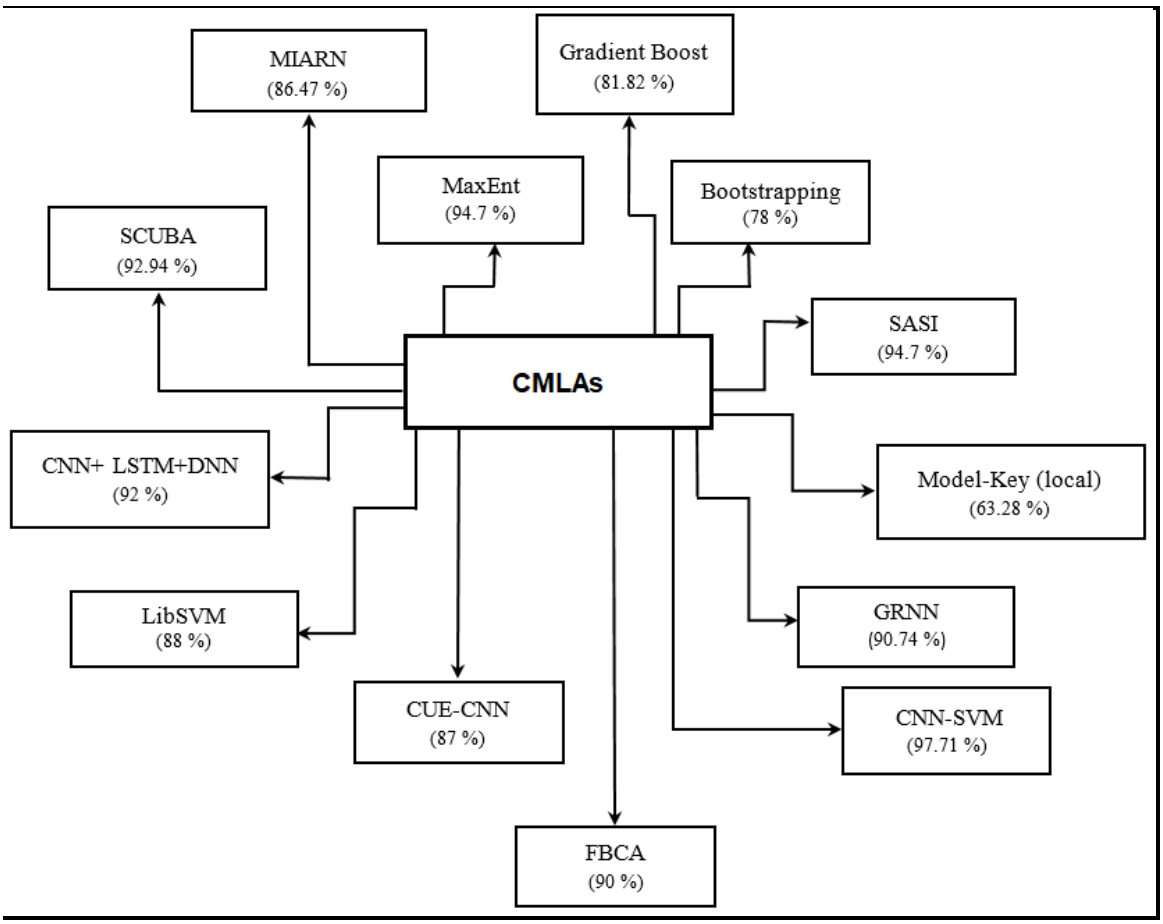

Figure 4: The classification of CMLAs for sarcasm detection

Common features of sarcasm detection

In text mining, extracting the features of the data is a critical process for the classification algorithm in order to form the final decision. During the classification process of text messages, certain features from social media posts can be used as an important factor for detecting sarcasm. Therefore, designing a dataset with the relevant features would significantly contribute to the overall machine learning performance. Applying different text mining techniques may result in different features. For each classification algorithm, the main features used to detect the sarcastic tweets are summarized in Table 3.

Lexical features

Lexical features are the most common feature type in text mining. The lexical features consist of unique words, phrases, noun phrases, or named entities that are associated with a score to show its degree of polarity. Using these features for emotion-mining purposes may help in determining the degree of emotions in the text (Yadollahi, Shahraki, & Zaiane, 2017). The lexical features can be categorized into unigrams- and dictionary-based features (González-Ibánez et al., 2011). According to (Ghosh et al., 2015; González-Ibánez et al., 2011), dictionary-based features are derived from a dictionary-sampling method that consists of four general classes: (a) Linguistic processes (e.g., adverbs and pronouns); (b) Psychological processes such as positive and negative emotions; (c) Personal concerns such as work and achievement; and (d) Spoken categories (e.g., assent and non-fluencies). Moreover, most dictionary-based features are derived from a list of interjections (e.g., ah, oh, and yeah) and punctuation, (e.g., ! and ?). Based on these, it can be observed that the lexical features have more discriminative clues that can be linked with user polarity scores. Thus, checking the polarity of words requires using different methods in order to estimate its degree with a high precision level.

In the sarcasm detection task, several approaches are usually applied to determine the polarity of words. For example, Tungthamthiti et al. (2014) used SentiStrength and SenticNet in order to precisely determine the polarity of words. SentiStrength is a sentiment lexicon that relies on linguistic information (e.g., name, label, and comment) and rules to estimate the sentiment strength of English text. SenticNet is an additional method for opinion mining, which aims at producing a collection of common-sense concepts with positive and negative sentiment scores. Nevertheless, Signhaniya, Shenoy, and Kondekar (2015) found that using the lexical structure

from a dependency tree can help improve the performance of the mining algorithm. In sum, the lexical features are essential elements for solving text mining problems. Lexical features can be used to recognize the polarity of a particular statement, thus making it possible to detect sarcastic tweets.

Stemmed features

Stemming is formulated based on the idea that words with the same stem are close in meaning. The stemming process aims at reducing the words in the world list effectively (Rani, Ramesh, Anusha, & Sathiaseelan, 2015). Stemmers can be used to consolidate terms which reduce the size of indexing files as well as enhancing the retrieval performance (Nayak, Chandavekar, & Balasubramani). According to Rani et al. (2015), the stemming process involves using an affix removal algorithm which removes prefixes and suffixes of the word in the document. To accomplish this process accurately, different types of algorithms are used. In general, the stemming algorithms are classified into three groups: truncating methods (e.g., porters stemmer); statistical methods (e.g., hidden Markov model stemmer); and mixed methods (e.g., Krovetz stemmer). Stemming allows applying these algorithms in the data pre-processing stage in order to extract the stemmed features that belongs to the task model. For this reason, Signhaniya, Shenoy, and Kondekar (2015 used the snowball stemmer in the data pre-processing stage in order to extract the stemmed features relevant to the sarcasm task. Saha et al. (2017) used the stem Porter operator to generate the stemmed words to perform polarity testing for the sarcasm detection task. Consequently, generating stemmed features during the pre-processing task may simplify the sarcasm detecting process. This is because the stemmed features can reduce inflectional data forms and derivationally-related forms of a word, thus influencing the performance of the machine learning algorithm.

Pragmatic features

Symbolic and figurative texts are referred to as pragmatic features (e.g., smiles and other emoticons). These features are frequently used in tweets, mainly due to the limitations in the tweet length. The pragmatic features are a powerful indicator for identifying sarcasm in Twitter. Therefore, in sarcasm classification, many researchers have extracted pragmatic features and used them in the classification process. For example, González-Ibánez et al. (2011) used three pragmatic features: (a) positive emoticons (e.g., smileys); (b) negative emoticons (e.g., frowning faces); and iii) ToUser, which is used to determine if a tweet is a reply to another tweet or not. Bali and Singh (2016) used pragmatic features with machine learning algorithms to identify the number of emoticons and expressions in the processed text messages. Parde and Nielsen (2018), on the other hand, used pragmatic features by considering the percentages of strongly subjective positive words, strongly subjective negative words, weakly subjective positive words, and weakly subjective negative words in the instance. Based on these, one can observe the importance of extracting pragmatic features from social media posts. Those features are associated with users' feeling towards a particular topic. This led prior studies to extract the pragmatic features from sarcasm-related tweets and link them with the latent emotion embedded in these tweets to build a rich dictionary that consists of users' emotions in association with their opinions.

Frequency-related features

Frequency-related features are commonly used in a document or a corpus. It reflects the importance of a word in a document or a corpus. Extracting the frequency-related features is a critical task; thus, it can be applied in sarcasm classification in different ways. For instance,

Barbieri et al. (2015) used two frequency corpora (the American National Corpus and the VU Amsterdam Metaphors Corpus) in order to extract three main features: rarest word frequency, frequency mean, and frequency gap. Researchers computed these features by considering only nouns, verbs, adjectives, and adverbs. Bouazizi and Ohtsuki (2016) divided words into two classes: the first class contained words of which the content is important, while the other class contained words of which the grammatical function is more important. Davidov et al. (2010) applied word frequency-related features to classify words into high-frequency words (HFWs) and content words (CWs). A word whose corpus frequency is more than F_H was considered to be a HFW, and vice versa. In conclusion, previous works showed the potential of frequency-related features in the classification of documents as well as the overall prediction process in sarcasms detection task.

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a numerical statistic that represents the importance of a word (term) to a document within the corpus. In TF-IDF, the frequency of a word in a document needs to be compared with its number of occurrences in other documents (Cong, Chan, & Ragan, 2016). TF-IDF is typically used to stop filtering words in text summarization and categorization application. It is also used to increase proportionally to the number of times that a word appears in a document. However, TF-IDF is generally offset by the frequency of the word in the corpus; therefore, such a technique helps to control the fact that some words are more common than others (Christian, Agus, & Suhartono, 2016). Because of this advantage, previous studies that explored sarcasm detection, adopted TF-IDF to extract features linked to sarcasm. Zhang et al. (2016) used TF-IDF values in order to sort the words in history tweets. To estimate TF and IDF, the authors regarded the set of history tweets for a given dataset as one document and used all

tweets in the training corpus to generate additional documents. In addition, Ren et al. (2018) used TF-IDF to sort all words in the sarcastic tweets through modeling all the contextual tweets as one document. They selected the most important words that had the highest TF-IDF values as an input feature of the machine learning algorithm. The utilization of TF-IDF features in sarcasm detection provides a way to determine the importance of a word for a document within the corpus. This can potentially facilitate the process of detecting sarcasm messages by helping machine learning algorithms to deal only with important words when detecting sarcastic tweets.

Part-Of-Speech (POS) taggers

Part-of-Speech (POS) taggers were developed to categorize words based on their parts of speech forms. It is commonly used in sentiment analysis due to the following reasons: a) words such as nouns and pronouns usually do not contain any sentiment. It can filter out such word with the help of a POS tagger; b) a POS tagger can also be applied to identify words that can be used in different parts of speech. The advantages of POS tagger have motivated researchers to apply it in the analysis of sarcastic tweets. For example, Ghosh, Guo, and Muresan (2015) used POS as an approach to model contextual information for co-training algorithms, which helped in creating a solid corpus and accurate predictions. Kovaz, Kreuz, and Riordan (2013) applied the Stanford POS tagger to annotate each statement of the obtained corpora. The authors were interested in analyzing adjectives, adverbs, and interjection statements, particularly immediate co-occurrences of <adverb + adjective> and <adjective + adjective>. Prasad, Sanjana, Bhat, and Harish (2017) reported the potential of using POS tagging in classifying words in a tweet and their parts of speech. It is argued that if a person uses a lot of adjectives, there is a possibility that he/she is providing hints about sarcastic tweets. Another study by Barbieri et al. (2015) used POS by including certain features designed to capture the structure of positive and negative

tweets. Consequently, due to the strong contribution of POS tagging to the classification problem, researchers have continuously utilized this technique in statement annotation. Before performing the classification procedure, applying POS tagging in complicated tasks such as sarcasm detection is necessary for assessing whether the statement can be labeled as sarcastic or not.

Ambiguity

According to Barbieri et al. (2015), if a word has many meanings (synset associated), it is more likely to be used in an ambiguous way. In the sarcasm detection task, Barbieri et al. (2015) calculated (for each word) several aspects including the maximum number of synsets associated with a single word, the mean synset number of all the words, and the synset gap (i.e., the difference between the two previous features). The authors determined the value of these features by including all the words of each tweet and also by considering only nouns, verbs, adjectives or adverbs. In general, it is very important to extract words with clear meaning at the very early stage (i.e., data pre-processing stage). The extracted words can highly contribute to the performance of the predictive model and this would enhance the processing decision of the algorithm.

Synonyms

The synonyms-related features refer to the process of extracting the features that share words of the same meaning. This approach seems to be useful in sarcasm detection when expressing a specific opinion in many ways. Therefore, in the sarcasm detection task, Barbieri et al. (2015) reported the process of choosing synonyms by retrieving the list of synonyms for each word, then computed (across all the words of the tweet) the greatest/lowest number of synonyms

with frequency higher than the one present in the tweet. They also determined the greatest/lowest number of synonyms as well as the mean number of synonyms of the words with frequency greater/lower than the one present in the tweet (gap feature). This includes computing the set of synonyms features along with all the words that contain POS features. Based on this, the use of synonyms would increase the performance of the classifier by processing different features that are relevant to the sarcasm detection task. In addition, synonyms can enrich the training set to be used for training the classifier during the detection task.

Personality

Over the past few decades, the five-factor (Big Five) model has emerged as one of the dominant models of personality that can be used to understand humans' behavior (Dhou, 2018, 2019). The five-factor model has been widely used to explore various personal psychological aspects. This makes personality an important factor for sarcasm detection-related tasks. Poria et al. (2016) used the five personality traits, such as Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism, for the training process. They then utilized a corpus containing 2,400 essays labeled by one of the five personality traits with a fully-connected layer of 150 neurons, which was treated as the main feature. It was found that the use of personality in sarcasm detection is effective. The reason behind that can be due to the fact that tweets contain hidden clues that are related to the individual's opinions/emotions.

From Table 3, it can be observed that CNN-SVM offers the highest classification performance when predicting the sarcastic tweets (97.71 %). The mentioned features can be used to add emotional and psychological information to the predictive model, thus contributing to its performance. This can be reasoned to the significant role of both lexical and personal features that were utilized by the CNN-SVM classifier. In addition, embedding features that are related to

the individual's personality traits in their post would enhance the performance of the machine learning algorithm.

Table 3: Common feature-related approaches that were applied for sarcasm detecting in Twitter

Algorithm	Author(s)	Sarcasm Features								
		Lexical	Stem	Pragmatic	Frequency	TF-IDF	POS	Ambiguity	Synonyms	Personality
SVM	González-Ibánez et al. (2011)	X		X	X		X			
	Tungthamthiti et al. (2014)	X		X						
	Bouazizi and Ohtsuki (2015)	X		X						
	Ghosh et al. (2015)	X		X	X		X			
	Barbieri et al. (2015)	X			X		X	X	X	
	Signhaniya et al. (2015)	X	X				X			
	Tungthamthiti et al. (2016)	X			X		X			
Logistic regression	Kovaz et al. (2013)	X			X		X			
	Jain and Hsu (2015)	X			X					
	Abercrombie and Hovy (2016)	X					X			
	Bali and Singh (2016)	X		X			X			
Naïve Bayes	Saha et al. (2017)	X	X				X			
	Das et al. (2018)	X			X					

	Parde and Nielsen (2018)	X		X	X			
Random Forest	Bouazizi and Ohtsuki (2018)	X		X			X	
	Bouazizi and Ohtsuki (2016)	X		X	X		X	
SASI	Davidov et al. (2010)	X			X			
Bootstrapping	Riloff et al. (2013)	X					X	
MaxEnt	Ptáček, Habernal, and Hong (2014)	X		X	X		X	
SCUBA	Rajadesingan et al. (2015)	X					X	
LibSVM	Joshi et al. (2015)	X		X				
CUE-CNN	Amir et al. (2016)	X			X		X	
GRNN	Zhang et al. (2016)	X				X		
CNN-SVM	Poria et al. (2016)	X						X
CNN+ LSTM+ DNN	Ghosh and Veale (2016)	X		X	X			
Gradient Boost	Prasad et al. (2017)	X	X				X	
MIARN	Tay, Tuan, Hui, and Su (2018)	X						

MODEL-KEY	Ren et al. (2018)	X		X	
FBCA	Parmar, Limbasiya, and Dhamecha (2018)	X	X		X

Contribution of sarcastic features to machine learning performance

In this work, we also attempted to identify the sarcastic features that are shared by the machine leaning algorithms of both AMLA and CMLA groups, because such features have high contribution to the performance of the classifier. This was achieved by creating a dataset of all the features that were used when performing the sarcasm detection task (see Table 3). Then, we clustered the data from each study using the "Hierarchical" clustering algorithm to find feature similarities between classifiers. The clustering results yielded eight distinct groups, shown in Figure 5. Each group contains the algorithms that share the same sarcastic features. From Figure 5, it can be observed that both cluster one (Cl-1) and cluster four (Cl-4) are having only one independent classifier (SVM and CNN-SVM, respectively). Moreover, SVM belongs to AMLA, while CNN-SVM belongs to CMLA. The classification performance of the two machine learning algorithms was the highest in its group. Thus, it can be said that using lexical, pragmatic, frequency, and POS tagging contribute to the performance of SVM (91.8 %), while both lexical and personal features can contribute to the performance of CNN-SVM (97.71 %).

Findings from this review may offer an in-depth understanding and justification for the performance of the algorithms that have the best performance in sarcasm detection. In other words, the utilized features within the SVM and CNN-SVM algorithms contributed to the quality of their predictions. Hence, using such features would enhance the detection of the sarcastic tweets. This observation helps future researchers and machine learning developers to consider the significant role of the sarcastic features when dealing with certain sarcasm classification problems.

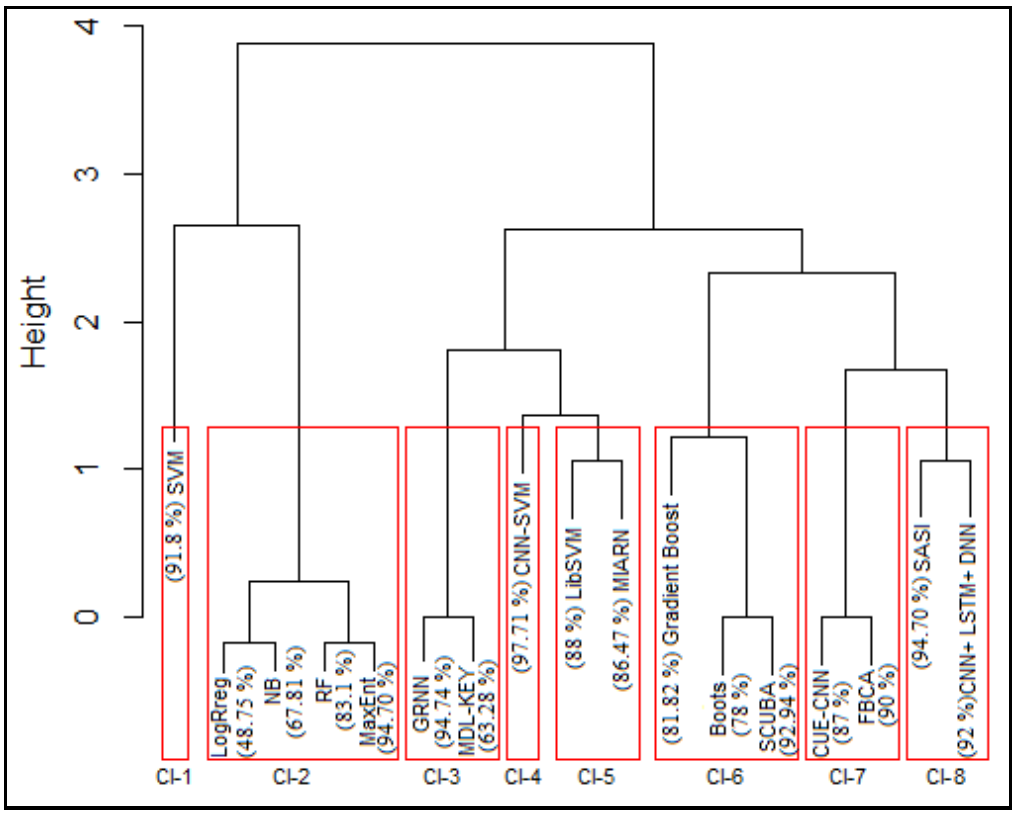

Figure 5: Cluster analysis result of algorithms and features for sarcasm detection

Conclusion

Sarcasm is a sophisticated form of irony that was extensively found on the Twitter platform. Detecting the sarcastic tweets is an essential matter in text classification, and thus has many implications. Therefore, this study reviewed various machine learning algorithms that were used to classify the sarcastic statements in Twitter. In this context, the PRISMA statement was used to provide a detailed guideline of the preferred reporting style for systematic reviews of classification methods that were used to perform sarcasm detection on Twitter. Extensive database searching led to the inclusion of 31 studies classified into two groups: AMLA and CMLA. Our results showed that SVM algorithm was the most used algorithm in the AMLA group for detecting the sarcasm in Twitter platform. In addition, CNN and SVM was found to offer a high prediction performance. However, other CMLAs used different text processing and

classification features. The variations in the design parameters of CMLAs resulted in different performance results. From the classification results, both SVM and CNN-SVM were the most efficient machine learning algorithms to predict sarcasm on Twitter. Our results also showed the potential of lexical, pragmatic, frequency, and POS tagging in improving the SVM performance. Also, we found that both lexical and personal features were the main predictors of sarcasm when using CNN-SVM. Based on these, it can be recommended that certain lexical, pragmatic, frequency, POS tagging, and personal features can be used in the recognition of sarcasm on microblogs. This study also recommends the use of two target labels when detecting the sarcastic statements tweets. Such labeling method (e.g., negative/positive or sarcastic/non-sarcastic) could highly contribute to the learning process of the utilized machine learning algorithm and thus boost the classification task. However, current study was limited by the published literature on sarcasm detection within the Twitter platform only. In addition, since sarcasm is one type of irony, we excluded the work that focused on irony detection. Furthermore, data-related issues such as number of utilized words, characteristics of the normalized instances technique, on the one hand, and algorithms-related core parameters such as the kernel type, on the other hand, were not discussed in this review. Based on that, future studies could consider the potential effect underlying such aspects on the performance of AMLAs and CMLAs for sarcasm detection in Twitter platform.

References

Abercrombie, G., & Hovy, D. (2016). Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations. In *Proceedings of the ACL 2016 Student Research Workshop* (pp. 107-113). ACL.

Altrabsheh, N., Cocea, M., & Fallahkhair, S. (2015). Detecting sarcasm from students' feedback in Twitter. In *Design for teaching and learning in a networked world* (pp. 551-555). Springer, Cham.

Amir, S., Wallace, B. C., Lyu, H., & Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Bali, T., & Singh, N. (2016, December). Sarcasm Detection: Building a contextual hierarchy. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)* (pp. 119-127). Osaka, Japan.

Bamman, D., & Smith, N. A. (2015, April). Contextualized sarcasm detection on twitter. *Proceedings of the Ninth International AAAI Conference on Web and Social Media* (pp.1-4). AAAI Press, New York.

Barbieri, F., Ronzano, F., & Saggion, H. (2015). UPF-taln: SemEval 2015 tasks 10 and 11. Sentiment analysis of literal and figurative language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 704-708). Denver, Colorado, June 2015. ACL.

Bouazizi, M., & Ohtsuki, T. (2015, August). Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*(pp. 1594-1597). IEEE.

Bouazizi, M., & Ohtsuki, T. (2018). Multi-class sentiment analysis in Twitter: What if classification is not the answer. *IEEE Access*, *6*, 64486-64502.

Bouazizi, M., & Ohtsuki, T. O. (2016). A pattern-based approach for sarcasm detection on Twitter. *IEEE Access*, *4*, 5477-5488.

Cambria, E., Poria, S., Bisio, F., Bajpai, R., & Chaturvedi, I. (2015, April). The CLSA model: A novel framework for concept-level sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 3-22). Springer, Cham.

Cerezo-Costas, H., & Celix-Salgado, D. (2015). Gradiant-analytics: Training polarity shifters with CRFs for message level polarity detection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 539-544). Denver, Colorado, June. ACL.

Christian, H., Agus, M. P., & Suhartono, D. (2016). Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, *7*(4), 285-294.

Cliche, M. (2014). The Sarcasm Detector: Learning Sarcasm From Tweets!. *The Sarcasm Detector, accessed Feb*, *19*(1).

Colston, H.L. (2000). On necessary conditions for verbal irony comprehension. *Pragmatics & Cognition, 8*(2), 277-324.

Cong, Y., Chan, Y. B., & Ragan, M. A. (2016). A novel alignment-free method for detection of lateral genetic transfer based on TF-IDF. *Scientific reports*, *6*, 30308.

Das, R., Kadam, S., Kalra, C., Nayak, V., & Govilkar, S. (2018). Sarcasm detection for english text.

Davidov, D., Tsur, O., & Rappoport, A. (2010, July). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning* (pp. 107-116). Association for Computational Linguistics.

Dhou, K. (2018, July). Towards a better understanding of chess players' personalities: A study using virtual chess players. In *International Conference on Human-Computer Interaction* (pp. 435-446). Springer, Cham.

Dhou, K. (2019, July). An Innovative Employment of Virtual Humans to Explore the Chess Personalities of Garry Kasparov and Other Class-A Players. In *International Conference on Human-Computer Interaction* (pp. 306-319). Springer, Cham.

Ghosh, D., Guo, W., & Muresan, S. (2015). Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1003-1012). Lisbon, Portugal. Association for Computational Linguistics.

Ghosh, A., & Veale, T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis* (pp. 161-169). San Diego, California. Association for Computational Linguistics

Gibbs, R., & Colston, H. (2007). The future of irony studies. In R. Gibbs & H. Colston (Eds.), *Irony in language and thought: A cognitive science reader* (pp. 581–595). New York: Erlbaum.

Gibbs, R.W. (2000). Irony in talk among friends. *Metaphor and symbol, 15*(1-2), 5-27.

González-Ibánez, R., Muresan, S., & Wacholder, N. (2011, June). Identifying sarcasm in Twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies: Short Papers-Volume 2 (pp. 581-586). Association for Computational Linguistics.

Jain, S., & Hsu, V. (2015). The lowest form of wit: Identifying sarcasm in social media.

Joshi, A., Sharma, V., & Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (Vol. 2, pp. 757-762). Beijing, China. Association for Computational Linguistics.

Joshi, A., Bhattacharyya, P., Carman, M., Saraswati, J., & Shukla, R. (2016). *How do cultural differences impact the quality of sarcasm annotation?: A case study of indian annotators and american text.* Paper presented at the Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities.

Joshi, A., Bhattacharyya, P., & Carman, M.J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR), 50*(5), 73.

Kannangara, S. (2018). *Mining twitter for fine-grained political opinion polarity classification, ideology detection and sarcasm detection.* Paper presented at the Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining.

Kovaz, D., Kreuz, R.J., & Riordan, M.A. (2013). Distinguishing sarcasm from literal language: Evidence from books and blogging. *Discourse Processes, 50*(8), 598-615.

Kumon-Nakamura, S., Glucksberg, S., & Brown, M. (2007). How about another piece of pie: The allusional pretense theory of discourse irony. *Irony in language and thought*, 57-96.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing, 2*(2010), 627-666.

Li, J., Fong, S., Zhuang, Y., & Khoury, R. (2016). Hierarchical classification in text mining for sentiment analysis of online news. *Soft Computing, 20*(9), 3411-3420.

Joshi, A., Bhattacharyya, P., Carman, M., Saraswati, J., & Shukla, R. (2016). How do cultural differences impact the quality of sarcasm annotation. In *Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities at ACL*.

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D.G. (2009). Preferred reporting items for systematic reviews and meta-analyses: The prisma statement. *Annals of internal medicine, 151*(4), 264-269.

Nayak, A. S., Kanive, A. P., Chandavekar, N., & Balasubramani, R. (2016). Survey on pre-processing techniques for text Mining. *International Journal Of Engineering And Computer Science, ISSN*, *5*(6), 16875- 16879.

Parde, N., & Nielsen, R. (2018, June). Detecting Sarcasm is Extremely Easy;-. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles* (pp. 21-26). New Orleans, Louisiana. Association for Computational Linguistics.

Parmar, K., Limbasiya, N., & Dhamecha, M. (2018, February). Feature based Composite Approach for Sarcasm Detection using MapReduce. In *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*(pp. 587-591). IEEE.

Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*.

Prasad, A. G., Sanjana, S., Bhat, S. M., & Harish, B. S. (2017, October). Sentiment analysis for sarcasm detection on streaming short text data. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)* (pp. 1-5). IEEE.

Ptáček, T., Habernal, I., & Hong, J. (2014). Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 213-223). Dublin, Ireland. University and Association for Computational Linguistics.

Rajadesingan, A., Zafarani, R., & Liu, H. (2015, February). Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (pp. 97-106). ACM.

Rani, S.R., Ramesh, B., Anusha, M., & Sathiaseelan, J. (2015). Evaluation of stemming techniques for text classification. *International Journal of Computer Science and Mobile Computing, 4*(3), 165-171.

Ren, Y., Ji, D., & Ren, H. (2018). Context-augmented convolutional neural networks for twitter sarcasm detection. *Neurocomputing, 308*, 1-7.

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

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 704-714). Seattle, Washington, USA. Association for Computational Linguistics.

Saha, S., Yadav, J., & Ranjan, P. (2017). Proposed approach for sarcasm detection in twitter. *Indian J. Sci. Technol, 10*(25), 1-8.

Shrout, P.E., & Fleiss, J.L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological bulletin, 86*(2), 420.

Signhaniya, A., Shenoy, G., & Kondekar, R. (2015). Sarcasm detection in social media.

Tay, Y., Tuan, L.A., Hui, S.C., & Su, J. (2018). Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*.

Tian, F., Gao, P., Li, L., Zhang, W., Liang, H., Qian, Y., & Zhao, R. (2014). Recognizing and regulating e-learners' emotions based on interactive chinese texts in e-learning systems. *Knowledge-Based Systems, 55*, 148-164.

Tungthamthiti, P., Kiyoaki, S., & Mohd, M. (2014). Recognition of sarcasms in tweets based on concept level sentiment analysis and supervised learning approaches. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing (*pp. 404–413*)*. Stroudsburg. ACL.

Tungthamthiti, P., Shirai, K., & Mohd, M. (2016). Recognition of sarcasm in microblogging based on sentiment analysis and coherence identification. *Journal of Natural Language Processing, 23*(5), 383-405.

Yadollahi, A., Shahraki, A.G., & Zaiane, O.R. (2017). Current state of text sentiment analysis from opinion to emotion mining. *ACM Computing Surveys (CSUR), 50*(2), 25.

Zhang, M., Zhang, Y., & Fu, G. (2016, December). Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers* (pp. 2449-2460). Osaka, Japan.