Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection

Zheng Lin Chia [a,*], Michal Ptaszynski [a], Fumito Masui [a], Gniewosz Leliwa [b], Michal Wroczynski [b]

[a] *Department of Computer Science, Kitami Institute of Technology, Kitami, Japan*
[b] *Samurailabs, Gdansk, Poland*

ARTICLE INFO

Keywords:
Irony detection
Sarcasm detection
Machine Learning

ABSTRACT

Irony and sarcasm detection is considered a complex task in Natural Language Processing. This paper set out to explore the sarcasm and irony on Twitter, using Machine Learning and Feature Engineering techniques. First we review and clarify the definition of irony and sarcasm by discussing various studies focusing on the terms. Next the first experiment is conducted comparing between various types of classification methods including some popular classifiers for text classification task. For the second experiment, different types of data preprocessing methods were compared and analyzed. Finally, the relationship between irony, sarcasm, and cyberbullying are discussed. The results are interesting as we observed high similarity between them.

1. Introduction

Irony and sarcasm detection has been increasingly recognized as an important task in the field of Natural Language Processing, with many related studies conducted on this topic in recent years (Barbieri, 2017; Burfoot & Baldwin, 2009; Ghosh & Veale, 2017; Reyes, Rosso, & Buscaldi, 2012). Irony, often used together or interchangeably with sarcasm, is considered an important component of human communication as one of the most prominent and pervasive figurative and creative language tools widely used dating back from ancient religious texts to modern time (Ghosh & Veale, 2017).

There has been little agreement on the correct definitions of irony and sarcasm despite their popularity as figurative means of expression in everyday communication, especially on the Internet (Hancock, 0000; Li & Ma, 2016; Papapicco & Mininni, 2020). A common and thorough clarification of differences between irony and sarcasm accepted by the whole research community does not seem to exist yet even with a considerable amount of related literature published in the past decades. Moreover, Hee (2017), in her attempt to define the concept, pointed out that many studies have also struggled to distinguish between irony, in particular verbal irony, and sarcasm. Thus she suggested not to distinguish between the terms due to the lack of clear quantifiable distinctions between those two phenomena despite numerous efforts. The difficulty of this task comes from the fact that, Similarly to other types of figurative language, ironic texts should not be interpreted in its literal sense due to requiring a more complex understanding based on associations with context and external world knowledge.

The nature of irony makes it an important component in Natural Language Processing (NLP) tasks, which include implications for the analysis, understanding and production of human language. In fact, automatic irony detection has a large potential for various applications in the domain of text mining, especially those that require semantic analysis, such as sentiment analysis

[*] Corresponding author.
E-mail address: chia@ialab.cs.kitami-it.ac.jp (Z.L. Chia).

https://doi.org/10.1016/j.ipm.2021.102600
Received 31 October 2020; Received in revised form 10 February 2021; Accepted 24 March 2021
Available online 29 April 2021

(Hee, 2017), author profiling, or detection of online harassment. Rosenthal, Ritter, Nakov, and Stoyanov (2014) demonstrated the impact of irony on automatic sentiment classification by attempting to analyze a test set of irony tweets with standard sentiment analysis tools, and showing the inability of those tools to maintain high performance on ironical texts. Thus automatic detection of irony and sarcasm has gradually become an important task in Natural Language Processing. Various types of approaches were developed and improved to tackle the problem of irony detection (Van Hee, Lefever, & Hoste, 2018). Some of the most popular approaches with better performance are rule-based, statistical, and Deep Learning-based (Amir, Wallace, Lyu, Carvalho, & Silva, 2016; Barbieri, 2017; Tsur, Davidov, & Rappoport, 2010). Most studies have focused on irony detection, however, there is still much space for improvement as indicated by Kumar, Somani, and Bhattacharyya (2017). Irony occurs typically due to incongruity in text, but sometimes a system may need to go beyond the information contained in the text to detect irony, such as by applying author details or further context.

The increase in social media users over the past decade has attracted researchers' interest in analyzing the new type of creative language used on the Internet to better explore the depth of human thoughts and communication. Among many Social Networking Services (SNS), one of the most popular platform for people to express their opinions, share their thoughts and report real-time events, has been Twitter (https://twitter.com/). Reported with more than 336 million monthly active users and 500 million tweets sent daily,[1] many companies and organizations have been interested in applying the data appearing on Twitter for the purpose of studying the opinion of people towards different products, facilities and events taking place around the globe. Therefore it has been suggested that datasets composed of tweets may be able to bring out the best performance of irony detection approaches Bouazizi and Otsuki (2016).

However, the lack of empirical investigations into optimal approaches for irony and sarcasm detection is a serious oversight in many related studies carried throughout recent years on data collected from Twitter. Moreover, the methods of collection, implementation of the datasets and their limitations have raised questions from researchers (Van Hee et al., 2018), and there have been no studies comparing the differences in data preprocessing and manipulation of the dataset to improve the results of irony and sarcasm detection.

This research seeks to remedy the above problems by analyzing some of the previous research gaps and proposing a novel approach after summarizing the results of experiments on various types of approaches and selecting the most suitable approach with optimal settings for additional experiments. We aim to evaluate whether it is possible to develop a universal classification model for irony and sarcasm detection, which can be further updated and tuned for better performance, with various Natural Language Processing methods, with particular focus on recent developments in Natural Language Processing and Artificial Intelligence (AI), such as Deep Neural Networks.

The main contributions of this paper are as follows:

- Clarification and understanding of differences and relationships between irony and sarcasm
- Investigation of the potential of various Natural Language Processing methods in irony and sarcasm detection
- Identification of the benefits of irony and sarcasm detection for other Natural Language Processing tasks, such as automatic cyberbullying detection

In the remained of this paper we firstly review and analyze previous state-of-the-art research on text-based irony and sarcasm detection (Section 2), investigate the potential of various Natural Language Processing and feature engineering methods in improving classification of irony (Section 3) and sarcasm (Section 4), and perform a deeper study into the actual practical differences between irony and sarcasm and their potential in other Natural Language Processing tasks (Section 5.1).

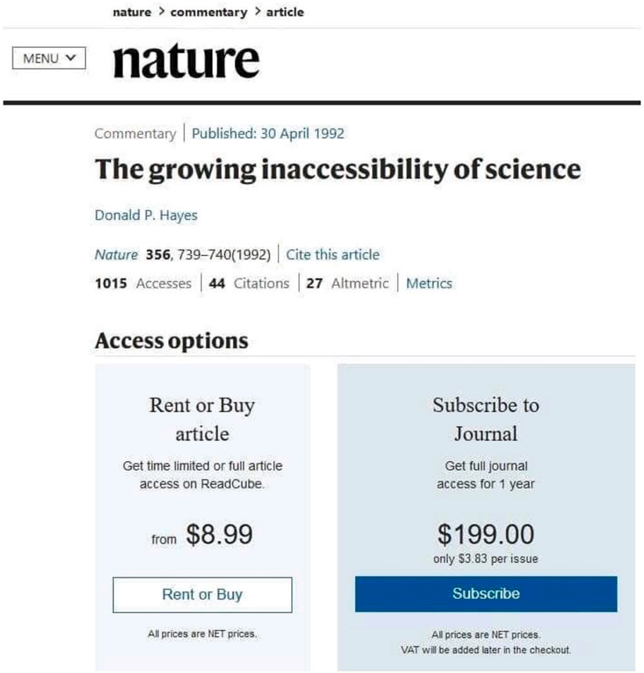
2. Related work

2.1. Definition of applied nomenclature

The Oxford Dictionary (https://en.oxforddictionaries.com/definition/irony) defines irony as "The expression of one's meaning by using language that normally signifies the opposite typically for humorous or emphatic effect". In accordance with the example above (Fig. 1), we will also include the definition of irony provided by the Merriam-Webster dictionary (https://www.merriam-webster.com/), which is the following: "The use of words to express something other than and especially opposite of the literal meaning".

While most of the previous research in irony detection within the field of AI focused on binary classification between ironic or non-ironic contents, two types of irony have been widely distinguished in most of the previous linguistic and communication studies on irony: verbal irony and situational irony (Barbieri, 2017; Hee, 2017; Sulis, Fariaz, Rosso, & Patti, 2016; Van Hee et al., 2018). This distinction has also been acknowledged in the Semantic Evaluation 2018 Workshop, a workshop in the form of a contest where multiple teams attempt to develop a Machine Learning method based on a unified dataset. The workshop's Task B especially focused on multi-class irony classification. The task had the participants compete in predicting one out of four labels describing (i) verbal irony realized though a polarity contrast, (ii) verbal irony without a polarity contrast, (iii) descriptions of situational irony, and (iv) non-irony (Barbieri, 2017; Van Hee et al., 2018).

[1] https://www.statista.com/topics/737/twitter/.



nature > commentary > article

MENU

nature

Commentary | Published: 30 April 1992

The growing inaccessibility of science

Donald P. Hayes

Nature **356**, 739–740(1992) | Cite this article

1015 Accesses | **44** Citations | **27** Altmetric | Metrics

Access options

Rent or Buy article

Get time limited or full article access on ReadCube.

from $8.99

Rent or Buy

All prices are NET prices.

Subscribe to Journal

Get full journal access for 1 year

$199.00

only $3.83 per issue

Subscribe

All prices are NET prices.
VAT will be added later in the checkout.

Fig. 1. An example of irony. The article in the figure indicated the growing inaccessibility of science, yet it charges readers for the access, which makes the article inaccessible for people who could not or do not wanted to pay.

Situational irony is an unexpected or incongruous event in a specific situation that fails to meet an expectation (Barbieri, 2017; Shelley, 2001). Shelley (2001) gives an example of a typically ironic situation regarding firefighters who left something cooking, had a fire in their kitchen while they were out putting down a fire in the other part of the city. As firemen are usually the ones who extinguish fire instead of starting it, this situation is quite unexpected and is considered ironic. This shows that situational irony is usually produced unintentionally and unplanned. As indicated by Grant, Hardy, Oswick, and Putnam (2004) "Situational irony focuses on the surprising and inevitable fragility of the human condition, in which the consequences of actions are often the opposite of what was expected".

According to "A glossary of literary terms" by Abrams and Harpham (2009), verbal irony is a statement in which the meaning that a speaker employs is sharply different from the meaning that is ostensibly expressed. An ironic statement usually involves the explicit expression of one's attitude or evaluation but with intended implications being very different, and often opposite, to the literal attitude or evaluation. Verbal irony is considered different from situational irony in that it is produced intentionally by the speakers.

On the other hand, sarcasm is defined to be "a way of using words that are the opposite of what you mean in order to be unpleasant to somebody or to make fun of them" by the Oxford dictionary (Oxford, 2020) As an attempt to provide explanation on the differences between irony and sarcasm, "A Dictionary of Modern English Usage" (Fowler, 1926) points out that sarcasm does not necessarily involve irony and irony has often no touch of sarcasm, even though sarcasm is often expressed with irony as a tool. Therefore, the relationship between verbal irony and sarcasm have been confused in many studies. On the other hand, Kreuz and Glucksberg (1989) argued that sarcasm and irony are similar in that both are forms of a reminder, yet different in that sarcasm conveys ridicule of a specific victim whereas irony does not. Lee and Katz (1998) followed up with an indication that a ridicule of a specific victim plays a more important role in sarcasm than in irony. They also pointed out that a sarcastic utterance brings to mind the expectation of a specific person who is identified by that expectation, whereas irony brings to mind the collective expectation of numerous people. In the same vein, Jorgensen (1996) coined the term "sarcastic irony" which is typically used to complain to or criticize intimates, who are usually the target of the remarks.

Attardo (1999) argues that sarcasm is an overtly aggressive type of irony and also claims that there is no consensus on whether sarcasm and irony are essentially the same thing, with superficial difference, and that they do not differ significantly. Many studies also claim that there is no way to distinguish between the terms (Tsur et al., 2010). Barbieri (2017) points out another reason why many researchers do not differentiate between the irony and sarcasm which is due to the observation of a shift in meaning between the two terms. They also conclude that while research efforts on irony and sarcasm are expanding, a formal definition is still lacking in the literature. Therefore, many researchers tend to not distinguish between the terms and consistently use either of them throughout their studies (Bouazizi & Otsuki, 2016; Buschmeier, Cimiano, & Klinger, 2014; Hee, 2017).

On the other hand, in some languages other than English, such as Japanese (where irony/sarcasm is called *hiniku*), have no distinction between irony and sarcasm. This is because the figurative function of irony, which in English is considered as a sophisticated figure of speech enriching conversation, in Japanese is used only in its aggressive (sarcastic) context.

In conclusion of all that has been mentioned so far, one may suppose that the difference between verbal irony and sarcasm is not definitive as most of the recent studies interpreted both the concepts as either the same, or with the differences between them being ambiguous (Buschmeier et al., 2014; Filatova, 2012). In this research, we performed experiments in both irony and sarcasm detection separately, then ultimately compared between them interchangeably to specify to what extent irony and sarcasm are overlapping concepts.

2.2. Feature engineering in irony and sarcasm detection

Feature engineering is a central task in data preparation for machine learning. It is the practice of constructing suitable features from given features that lead to improved predictive performance (Nargesian, Samulowitz, Khurana, Khalil, & Turaga, 2017). The feature engineering process usually includes brainstorming or testing features, creating features, and improving features if needed (Jalal & Adeeb, 2018). Galli (2020), Kuhn and Johnson (2020) and Zheng and Casari (2018) listed a few popular feature engineering techniques including feature creation, extraction, filtering, binning, and scaling, and variable transformation, categorical encoding and data imputation.

Some of the most popular and simple representation of feature engineering in text classification is the bag of words model(BoW) and Term Frequency–Inverse Document Frequency(TF*IDF) (Scott & Matwin, 1999; Zhang, Taketoshi, & Tang, 2011). Bag of words and TF*IDF are examples of feature creation and scaling (Zheng & Casari, 2018). For most bag of words representations, each feature corresponds to a single word found in the training corpus, usually with case and punctuation removed. Conceptually, we can view bag-of-word model as a special case of the n-gram model, with n = 1. TF*IDF is a traditional weight calculation scheme evolved from IDF which is proposed by Jones (2004) with heuristic intuition that a query term which occurs in many documents is not a good discriminator, and should be given less weight than one which occurs in few documents (Zhang et al., 2011). A survey conducted by Beel, Gipp, Langer, and Breitinger (2016) showed that 83% of text-based systems in digital libraries use TF*IDF.

2.3. Previous research

Tepperman's (2006) spoken dialogue system used feature extraction approach for sarcasm detection as a sub-function in their system, by which they introduced sarcasm detection into the scene of Nature Language Processing. They presented some experiments toward sarcasm recognition using voice-based features such as prosodic, spectral, and contextual cues. One of the first text-based sarcasm detection work is the study by Davidov, Tsur, and Rappoport (2010) and Tsur et al. (2010), who utilized tweets and Amazon reviews with feature engineering and statistical classifiers. Their semi-supervised algorithmic methodology is based on patterns where they extracted, selected, and matched with their corresponding feature values, while also exploring the utility of Twitter sarcastic hashtag (e.g. #sarcasm) which served an important role in sarcasm and irony detection on tweets.

Most of the automatic detection models proposed in the previous studies focus on short text from microblogs (e.g., Twitter) (Barbieri, 2017). However, there are actually some related works which focused on long text datasets (e.g. Amazon review, News Article, etc.) (Tsur et al., 2010). A number of studies have sought to detail the recent trend in sarcasm detection approaches, which can roughly be classified into three parts: rule-based, statistical, and Deep Learning approaches (Barbieri, 2017; Kumar et al., 2017). We summarize the most representative research from each category, below.

2.3.1. Rule-based methods

Rule-based approaches attempt to identify irony through specific evidence which could be captured in terms of rules that rely on indicators of irony and sarcasm. Barbieri (2017) argued that rule-based approaches which require no training mostly rely on lexical information and do not perform as well as statistical approaches. However, it is interesting to study how researchers designed their systems starting from the beginning. Veale and Hao (2010) used Google search for determining whether a given simile is intended to be sarcastic and presented a 9-step approach where the data was validated using the size of the search results.

Riloff et al. (2013) aimed to recognize positive words in negative sentences while presenting a bootstrapping algorithm that automatically learns the rules from certain situations. Maynard and Greenwood (2014) proposed a model with a set of rules to predict the irony from hashtags. They highlight the importance of hashtags which are included by the author to indicate the irony in their tweets.

2.3.2. Statistical methods

Most of the early works on sarcasm detection applied statistical approaches which varied in terms of features and learning algorithms, basically composed of two phases where data were converted into feature vectors before being classified using machine learning algorithms. Statistical approaches to sarcasm and irony detection vary in different aspects. Some of the most often used algorithms have been Support Vector Machines (SVM), and Naïve Bayes.

One of the first attempts in this approach by Tsur et al. (2010) compiled a set of sarcastic patterns composed of common combinations of words extracted from sarcastic examples. They extracted pattern-based features and punctuation-based features before applying k-Nearest Neighbor (kNN) algorithm for classification. Gonzalez-Ibanez, Muresan, and Wacholder (2011) composed a model with three pragmatic features which were positive emoticons, negative emoticons, and users' tagging. Reyes, Rosso, and Veale (2013) proposed another model based on four features, signatures, unexpectedness, style and polarity, and emotional scenarios. Liebrecht, Kunneman, and den Bosch (2013) and Kunneman, Liebrecht, Mulken, and Bosch (2015) introduced bi-gram and tri-gram based features into sarcasm detection. They designed a sarcasm detector which tested on tweet data marked with hashtag #sarcasme (Dutch for sarcasm).

2.3.3. Deep learning

Deep Learning approaches have been successfully brought into the scene of sarcasm detection when Amir et al. (2016) used a standard Convolutional Neural Network (CNN) in binary classification, while Poria, Cambria, Hazarika, and Vij (2016) implemented a combination of CNNs trained on different tasks. Popular deep learning algorithms include CNN (LeCun, Bottou, Bengio, & Haffner, 1998) and Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) networks.

In particular, the convolutional network-based method proposed by Amir et al. (2016), automatically learned and then exploited user embeddings, to be used in concert with lexical signals to recognize sarcasm, in contrast to most past methods which required laborious feature engineering. Zhang, Zhang, and Fu (2016) investigated the use of neural networks for tweet sarcasm detection, and compared the effects of the continuous automatic features with discrete manual features. They particularly used a bi-directional gated recurrent neural network (RNN) to capture syntactic and semantic information over tweets locally, and a pooling layer to extract contextual features automatically from previous tweets. Their results show improvement in accuracy for sarcasm detection from neural features with different error distributions when compared to discrete manual features.

Ghosh and Veale (2017) proposed a network model composed of CNN followed by an LSTM network which outperformed many other models at that time. They utilized CNN to reduce frequency variation through convolutional filters and extracted discriminating word sequences as a composite feature map for the LSTM layer. Then the output of the LSTM layer was passed to a fully connected Deep Neural Network(DNN) layer, which produced a higher order feature set based on the LSTM output. Their systems showed important improvements in sarcasm detection, however, they argued it still lacked an ability to differentiate sarcasm at conceptual level.

Following the Semantic Evaluation 2018 international workshop Task 3: Irony Detection in English Tweets (Van Hee et al., 2018) which received submissions from 43 teams worldwide for the binary classification task A, deep learning algorithms were further explored and optimized for irony detection tasks. The best ranked system submitted by team THU_NGN (Wu et al., 2018) consisted of densely connected LSTM network with multi-task learning strategy. Another system from one of the top teams, NTUA-SLP (Baziotis et al., 2018), which used an ensemble of two bi-directional LSTM network-based models, achieved comparable results. The submissions represented a variety of neural network-based approaches and other popular classification algorithms including SVM, Random Forest, and Naïve Bayes (Van Hee et al., 2018). Overall, the approaches with ensemble learners were the current trend to tackle the challenges in irony and sarcasm detection.

Zhang, Zhang, Chan, and Rosso (2019) introduced a sentiment-based transfer learning approach which uses sentiment knowledge to improve the attention mechanism of recurrent neural models for capturing hidden patterns for incongruity. They proposed a sentiment transferred Bi-LSTM model which is designed to transfer deep features from sentiment analysis into irony detection for learning both explicit and implicit context incongruity. Their model achieved state-of-the-art performance at the time and proved using sentiment knowledge is a very effective approach to improving irony detection. The appearance of the attention-based Transformer model proposed by Vaswani et al. (2017) has inspired various popular language representation models including the Bidirectional Encoder Representations from Transformer(BERT) (Devlin, Chang, Lee, & Toutanova, 2019). Potamias, Siolas, and Gergios (2019) proposed a neural network methodology that builds on a pre-trained transformer-based network architecture which is further enhanced with the employment in a recurrent convolutional neural network (RCNN). They tested their model on the Semantic Evaluation 2018 Task 3 dataset and achieved results surpassing all of the submissions. Their model also achieved start of the art performance under all benchmark datasets, outperforming all other methodologies and published studies.

3. Exploring potential of various classifiers for irony detection

3.1. Dataset

In our first experiment, we implemented the dataset provided by Semantic Evaluation 2018 Task 3: Irony Detection in English Tweets (Van Hee et al., 2018). It was constructed by searching Twitter for hashtags #irony, #sarcasm, and #not, which could occur anywhere in the tweet that was finally included in the dataset. Hashtag is the hashtag symbol(#) combined with a relevant keyword or phrase in a tweet, used to categorize those tweets and help show them more easily in Twitter search. All tweets were collected between 2014/12/01 and 2015/01/04 and represent 2676 unique users, and were manually labeled using a fine-grained annotation scheme for irony (Hee, 2017). The entire dataset was cleaned by removing retweets, duplicates and non-English tweets, and replacing XML-escaped characters (e.g. &amp;).

The dataset consists of 4618 tweets (2222 ironic and 2396 non-ironic) that were manually labeled by three students using the Brat Rapid annotation tool[2] with an inter-annotator agreement study set up to assure the reliability of the annotations (Van Hee et al., 2018). Additionally, two duplicate sets of the data were applied with all the ironic hashtags removed and with hashtags only.

[2] https://brat.nlplab.org/.

3.2. Data preprocessing and feature weighting

Primarily, feature engineering were applied to the datasets as following. All of the tweets were transformed into lowercase and emojis were represented with their corresponding labels (e.g. :smiley_face:) using Emoji for Python (Kim & Wurster, 2019). Furthermore, all URLs (e.g. https://google.com/) and tagged users (e.g. @user123) were replaced with specific tokens "_url_" and "_tagged_". It is because the URLs and the tagged users were not likely to be contributing to the classification, but when left unprocessed could create undesirable bias.

Traditional weight calculation scheme for feature scaling was applied to the datasets. In particular, term frequency with inverse document frequency ($TF * IDF$) were used. Term frequency $tf(t,d)$ refers to the traditional raw frequency, which is the number of times a term t (word, token) occurs in a document d. Inverse document frequency $idf(t,D)$ is 1 divided by the logarithm of the total number of documents D containing the term t. Finally $TF * IDF$ refers to the term frequency multiplied by inverse document frequency as in Eq. (1).

$$idf(t,D) = log\frac{|D|}{n_t} \tag{1}$$

3.3. Classification methods and evaluation

Several types of classifiers were applied for comparison in this research.

Naïve Bayes classifier is a supervised learning algorithm applying Bayes' theorem which assigns class labels to problem instance represented as vectors of feature values, often applied as a baseline in text classification tasks.

The next classifier applied was the **k-Nearest Neighbors (kNN)** classifier, which takes an input k-closest training samples and classifies them based on the majority vote. It is often used as a baseline after Naïve Bayes. For the input sample to be assigned to the class of the first nearest neighbor, $k = 1$ setting was applied here.

JRip also known as **Repeated Incremental Pruning to Produce Error Reduction (RIPPER)** which is efficient in classifying noisy text (Sasaki & Kita, 1998), learns rules incrementally in order to optimize them.

Another classifier was **J48** which is an implementation of the **C4.5 Decision Tree** algorithm, which builds decision trees from dataset and the optimal splitting criterion are further chosen from tree nodes to make the decision.

Support Vector Machines (SVM) refers to a supervised machine learning algorithm designed for classification or regression problems which uses a technique called kernel trick to transform data and finds an optimal boundary between the possible output. Two types of SVM functions applied here were linear and radial.

Finally, **Convolutional Neural Networks (CNN)** which are a type of feed-forward neural network, were applied with Rectified Linear Units (ReLU) as neuron activation function. The implement CNN method consisted of two hidden convolutional layers, containing 20 and 100 feature maps with both layers having 5 × 5 size of patch and 2 × 2 max-pooling, and Stochastic Gradient Descent as the optimization function (LeCun et al., 1998).

Three separate datasets provided from the original processed dataset were applied in the experiment. Each of the classifiers above was tested on the three datasets in a 10-fold cross validation setting. The results were calculated using standard Accuracy (A), Precision (P), Recall (R), and balanced F-score (F1). The results ranking were determined based on the highest achieved balanced F-score.

3.4. Results and discussion

From Table 1 which shows the summarization of all results, the results from the dataset with hashtags included were significantly higher than for the other dataset without hashtags. As stated by Maynard and Greenwood (2014), even without considering ironic hashtags, the presence of hashtags greatly improves the results of irony detection on Twitter.

The kNN scored the lowest among the classifiers for both dataset and Naïve Bayes obtained closely low results. Even though these classifiers may be able to do well in typical sentiment analysis, few processing such as stemming and parsing are not applied to the dataset as a result of simple data preprocessing, resulting in some noisy language, which is a challenge for kNN and Naïve Bayes.

For the decision tree-based classifiers, J48 did better than Random Forest with hashtag included. However, J48 scored as low as kNN when hashtags were removed. Random Forest scored third highest for both datasets but it is unfortunately impractical due to being time-inefficient when comparing to SVM. The rule learner algorithm, JRip scored highest when hashtags were included but performed just slightly above kNN and J48 when hashtags were removed.

The most often used algorithms in irony detection are SVMs. From what we observed, the results of the radial-SVM is comparable to the proposed CNN. They achieved the same F-score on dataset with hashtags and SVM ranked second just after CNN for the dataset without hashtags. The linear-SVM, however, did not perform well enough in both conditions.

When it comes to the proposed CNN with two hidden layers, 5 × 5 patch size, max-pooling, and Stochastic Gradient, it outperformed all of the classifiers under the harsh conditions where all hashtags were removed (F-score = 0.66). While CNN is time-inefficient comparing to other classifiers in small datasets, larger dataset might produce different result. The current state-of-the-art irony detection system, Ghosh model (Ghosh & Veale, 2017) is also a neural network model composed of CNN. They applied a dataset of 39k tweets, which we will be implementing in the next study.

Table 1
Results of irony detection experiment (F-scores).

Classifiers	With hashtag	No hashtag	Only hashtag
kNN	0.753	0.571	0.881
Naïve Bayes	0.808	0.621	0.758
Random forest	0.883	0.641	0.898
J48	0.883	0.641	0.884
JRip	0.899	0.616	0.897
SVM-linear	0.826	0.615	0.893
SVM-radial	0.844	0.644	0.833
CNN	0.844	0.660	N/A

The last column of Table 1 shows the results of the dataset which consists of only the hashtags. Besides CNN which the results could not be calculated due to the information loss (too few features – only hashtags – caused incapability to properly generalize over the dataset), all the remaining classifiers attain high F-score comparable to the dataset with hashtag. Together these results provide important insights into the presence of hashtags in tweets, especially ironic hashtag for irony detection.

These findings enhance our understanding of the impact of hashtag, which greatly contributes to the improvement in irony detection. In general, irony detection is still a challenging and unsolved task in Natural Language Processing, but it can be considered an easy task in practice when performed on data from Twitter thanks to the deliberate use of ironic hashtags. Taken together, these results also suggest that hashtag is the product of authors who realize that their ironic phrases alone may not be enough for their audience to understand. This redefines irony in textual communication especially on social network services from implicit figurative speech to explicit speech.

4. Exploring data preprocessing methods for sarcasm detection

4.1. Dataset

The dataset used in the second study was the publicly available sarcasm detection dataset collected by Ghosh and Veale (2017) which consists of 51,189 tweets (24,453 sarcastic tweets and 26,736 non-sarcastic tweets) in which sarcastic tweets were automatically collected from Twitter using user's self-declaration of sarcasm/irony with sarcastic and ironic hashtags (e.g. #irony, #sarcasm) and annotated for confirmation.

The dataset is then implemented with seven different data preprocessing methods.

4.2. Dataset preprocessing

In the majority of recent studies applying machine learning methods to text classification, the datasets are usually used in their most basic form, namely, represented as tokens (words, punctuation marks, etc.), despite a wide variety of knowledge-based Natural Language Processing systems (e.g., stemmers, part-of-speech taggers, etc.) capable of initial preprocessing of datasets, thus providing more informative features to Machine Learning algorithms. However, there are studies about impact of data preprocessing on the performances of classifiers (Chandrasekar & Qian, 2016) and (Kalra & Aggarwal, 2018). Therefore, in the second study, we performed additional feature engineering and preprocessing to the dataset to verify usefulness of such knowledge-based systems in Machine Learning.

For the implemented datasets, each tweet was first transformed into lowercase and emojis were represented with their corresponding labels (e.g. :smiley face:) using Emoji for Python (Kim & Wurster, 2019). All tagged users (e.g. @user123) and URLs (e.g. http://google.com/) appearing in the text were replaced with specific neutral labels, such as "_tagged_" and "_url_". This initial preprocessing was the same as for the first experiment. All subsequent dataset preprocessing techniques used in this study were explained below and the example of the datasets are shown in Fig. 2. The product of the approaches are 7 different and independent datasets.

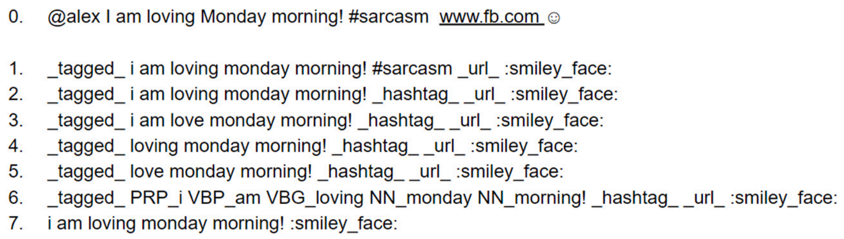
1. Only basic (initial) preprocessing.

To verify the depth of dependence of sarcasm detection on hashtags, all of the hashtags (e.g. #sarcasm) in the next 5 versions of the dataset shown below were replaced with a general label, e.g., "_hashtag_".

2. URLs, tagged users and hashtags replaced with labels.

Furthermore, we applied the knowledge-based tools for language processing provided by NLTK (https://www.nltk.org/).

3. Stemming of all words using Porter Stemmer (Porter, 2019)
4. Stopwords removal with NLTK built-in Stopwords Filtering Tool
5. Stemming of all words after stopwords removal
6. PoS tagging using NLTK Universal Part-of-Speech Tagset



0. @alex I am loving Monday morning! #sarcasm www.fb.com ☺

1. _tagged_ i am loving monday morning! #sarcasm _url_ :smiley_face:
2. _tagged_ i am loving monday morning! _hashtag_ _url_ :smiley_face:
3. _tagged_ i am love monday morning! _hashtag_ _url_ :smiley_face:
4. _tagged_ loving monday morning! _hashtag_ _url_ :smiley_face:
5. _tagged_ love monday morning! _hashtag_ _url_ :smiley_face:
6. _tagged_ PRP_i VBP_am VBG_loving NN_monday NN_morning! _hashtag_ _url_ :smiley_face:
7. i am loving monday morning! :smiley_face:

Fig. 2. Example of data preprocessing. Data 0 is the raw version of the ironic tweet.

Table 2
Results from seven datasets with different preprocessing. Dataset 1 included hashtags, dataset 2 had all social media markers replaced with neutral labels (hashtags included), dataset 3 had stemming applied, dataset 4 had stopwords removed, dataset 5 had stemming applied and stopwords removed, dataset 6 had PoS Tagging, and dataset 7 had all social media markers removed.

	Dataset	True positive	False positive	False negative	True negative	F-score
1	W/#	24 355	98	72	26 664	0.997
2	W/o #	24 055	398	5068	21668	0.898
3	Stem	24 013	440	5172	21 564	0.895
4	Stopw	24 009	444	5183	21 553	0.895
5	Stem + Stopw	24 163	290	5590	21 146	0.892
6	PoS Tag	23 904	549	5171	21 565	0.893
7	No label	16 509	7944	8677	18 059	0.665

Finally the last dataset 7 we used had its social media markers such as hashtags, URLs, and tagged users, etc., completely removed instead of being replaced with labels.

7. Tagged users, URLs, and hashtags removed

4.3. Experiment setup

For the following experiments, we will be implementing datasets mostly without hashtags included, where they are replaced with neutral labels. Therefore, based on the results of the first experiment, in this study we propose to implement the Convolutional Neural Networks (CNN) due to it having the best result for classifying tweets without ironic hashtags in comparison with other classifiers.

CNN are a type of feed-forward artificial neural network which is an improved neural network model originally designed for image recognition. CNN performance has been proved useful in various classification tasks including sentence classification and Natural Language Processing (Kim, 2014; Ptaszynski, Eronen, & Masui, 2017).

In the proposed CNN, we applied Rectified Linear Units (ReLU) as neuron activation function which is a piece-wise linear function that will output the input directly if positive, zero if negative. Max pooling, which applies a max filter to non-overlying sub-parts of the input to reduce dimensionality and in effect correct over-fitting, is also implemented accordingly. We also applied dropout regularization on penultimate layer, which prevents co-adaptation of hidden units by randomly dropping out some of the hidden units during training. The CNN consisted of two hidden convolutional layers, containing 20 and 100 feature maps, respectively, with both layers having 5 × 5 patch size and 2 × 2 max-pooling, and Stochastic Gradient Descent (LeCun et al., 1998).

Feature vector are created using Word2vec, which is a two-layer neural net that processes text by "vectorizing" words, with traditional weight calculation scheme ($TF * IDF$) and the results were calculated using standard balanced F-score (F1). Product from the dataset processed are used later in the input layer of classification.

All seven separate versions of the dataset (represented with various preprocessing techniques) were analyzed in the experiment using the proposed CNN method in a 10-fold cross validation setting.

4.4. Result and discussion

Table 2 shows the summary of all results from the seven datasets with different preprocessing techniques applied. Dataset 1, which is the dataset with all information included, yielded an F1 score of 0.997. Compared to the results from the first experiment which tested on a smaller dataset with only 4618 tweets and attained an F1 score of 0.844 with similar settings (hashtags included), significant increase are shown in the performance of the CNN model following the increase of the dataset size. This suggests that the model is tied to the size of the implemented dataset and the number of extracted features.

The results of dataset 1 (hashtags included) once again remind us of the impact of hashtags, which make a great difference in sarcasm and irony detection, especially on Twitter. However, due to the natural characteristics of deliberate sarcastic hashtags in

Table 3
Top 6 error feature occurrences for dataset 1, 2 and 7 (occ = occurrence).

Dataset 1	**occ**	Dataset 2	**occ**	Dataset 7	**occ**
#sarcasm	71	_hashtag_	5445	love	1656
sarcasm	60	_tagged_	1639	like	1216
tagged	51	love	413	not	1211
love	22	great	275	good	752
great	8	not	245	great	709
not	8	best	133	hate	488

Twitter, classification of tweets with hashtags included does not contribute much to the study of sarcasm detection from the linguistic point of view. Despite that, hashtags can be a very useful practical mean to handle sarcasm detection with high performance from the results shown.

While the remaining datasets were stripped of their hashtags (replaced with labels), dataset 2 has no further preprocessing while dataset 3 to 6 were further processed with different methods. Interestingly, dataset 2 still attained the highest F1 score among all the datasets without hashtags included. This discovery highlights the importance of linguistic features in irony detection and shows that increment in data preprocessing does not always provide better results. The reason of this is the oversimplification of data by manipulating or removing many vital and important features, which in classification tasks such as irony detection are of great importance.

However, further preprocessed datasets have their own value despite attaining lower F1 score. From our observation on the attributes extracted from their confusion matrices in Table 2, their true positive rate is higher than the dataset 2 which scored the highest F1 score among the datasets. dataset 5 which implemented both stemming and stop-word removal has obtained the highest true positive rate with only 290 false positives. This shows the implementation of various data preprocessing is also crucial to the sensitivity of the model.

Finally for the last dataset 7 which had all of its social media markers, such as tagged users (e.g. @user123), URLs, and hashtags completely removed, Table 2 shows that the result dropped significantly to an F1 score of 0.665, comparing to other datasets. This case has shown the impact of the labels which were supposed to be neutral to the classification, actually affected the results. Comparing to dataset 2 which had the social media markers replaced with labels, the significant increase in false negatives shows that the presence of the labels provides heavy contribution to the precision of the classification.

Moreover, what can be clearly seen from the numbers of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN), all models (except the highest scoring one based on dataset 1) make comparatively few mistakes when classifying a tweet as sarcasm, when in fact it is non-sarcastic (FP), and mostly make mistakes when deciding that a tweet is not sarcastic, when in reality it is (FN). This suggests, that the approach has a bias toward the non-sarcastic class. In the following section we perform error analysis to further investigate this result.

4.5. Error analysis

Table 3 shows the occurrences of top 6 error features extracted from dataset 1 (with hashtags), dataset 2 (hashtags replaced with labels) and dataset 7 (hashtags, URLs, and tagged users removed) after removing prepositions, conjunctions, and pronouns which do not contribute much to the classifications. For dataset 1, the error feature which occurred the most is the #sarcasm following the word sarcasm. This shows that even the sarcastic hashtags cannot assist the model to achieve 100% sensitivity.

For the dataset 2 results in the second column, the label _hashtag_ appeared 5445 times out of the 5466 misclassified instances (99.62%). Coming up next is the label _tagged_ which appeared 1639 times while the remaining words such as “love”, “great”, “not”, and “best”, which are popular errors in all the 3 implemented datasets. As previously noticed, the supposedly neutral labels, in fact contribute heavily to the precision of the classification. Therefore, removing them does not provide improvement to the results. Instead, it might even decrease the classification result.

Below are two examples of sarcastic tweets from dataset 2 without the sarcastic hashtag(labeled) being recognized as non-sarcastic:

```
1. _tagged_ An absolute stunning model of behavior for young kids. _hashtag_
2. less than 2 hours to champions league :))))) _hashtag_ _hashtag_
```

And the following two are examples of non-sarcastic tweets where the model classified as sarcastic:

```
1. That moment when you need food and everything on campus is closed #hungry
2. Its the most wonderful time of the year its #UFHateWeek
```

The evidence so far provides further support for the hypothesis that deliberate sarcastic hashtags play a significant role in sarcasm detection in tweets. Taken together, these results also suggest that hashtag is the product of authors who understand that their sarcastic phrases alone may not be sufficient for the audience to figure out the intended irony or sarcasm. However, these findings do not completely solve the general sarcasm detection nor do they redefine sarcasm or irony in textual communication especially on social network services.

Table 4
Comparison between Sarcasm and Irony.

#hashtag	Train set	Test set	F-score
Labeled	Irony	Sarcasm	0.852
Labeled	Sarcasm	Irony	0.860
Included	Irony	Sarcasm	0.940
Included	Sarcasm	Irony	0.945

5. Additional experiments

5.1. Exploring differences between irony and sarcasm

The definition and relationship between sarcasm and irony have been confused in many previous studies. To test the similarity between them, we conducted an experiment involving both sarcasm and irony. In this experiment to compare between sarcasm and irony, we used both irony dataset and sarcasm dataset with and without hashtags (labeled). Based on the previous experiments, we used JRip as the proposed classifier due to it having the best result for classifying tweets with hashtags when compared to other classifiers. JRip also known as Repeated Incremental Pruning to Produce Error Reduction(RIPPER), which is a propositional rule learner proposed as an optimized version of IREP.

All datasets were applied with the same initial preprocessing. Namely, each data sample was first transformed into lowercase and emojis were represented with their corresponding labels (e.g. :smiley face:) using Emoji for Python (Kim & Wurster, 2019). All tagged users (e.g. @user123) and URLs (e.g. http://google.com/) appearing in the text were replaced with specific neutral labels, such as "_tagged_" and "_url_". There were two version for each datasets, first with hashtags and second with all hashtags being replaced with a general label (e.g.: _hashtag_). Traditional weight calculation scheme ($TF * IDF$) was applied and the results were calculated using standard balanced F-score (F1).

5.1.1. Results and discussion

Table 4 shows the results of comparison between sarcasm and irony, with #hashtags included and labeled as "_hashtag_". For results on comparison with hashtags labeled, the results were 0.852 and 0.86 respectively. Considering there were no sarcasm or irony indicator such as #hashtags in the dataset, both results are similarly good indicating sarcasm and irony are in practice nearly identical even if mostly linguistic features are used. The results also support the claim which defines sarcasm as being an aggressive type of irony (Attardo, 1999).

For results on comparing datasets with hashtags included, both attained a similar F-score of around 0.94. Interestingly, the results are very high and it is even better than the results of irony detection alone in Table 1.

This further indicates the similarity between sarcasm and irony, especially on the #hashtags which plays important roles for being intended sarcastic and irony markers provided by the authors of the tweets. This observation further supports the evidence in meaning shift occurring between sarcasm and irony mentioned above. Hence most authors of sarcastic and ironic tweets do no longer distinguish clearly between the term sarcasm and irony, resulting in applying them interchangeably.

5.2. Application of sarcasm in cyberbullying detection

Although the number of research on sarcasm and irony detection grows each year, practical implementation of such models have not been widely discussed. Ptaszynski et al. (2010) mentions, that sarcasm poses a problem in cyberbullying detection. Therefore, aiming to improve their expert system for automated Internet Patrol, in this additional experiment we propose a practical implementation of sarcasm detection in cyberbullying detection.

To quantify the extent to how such model would be useful, we applied a model trained on sarcastic dataset 2 in Table 2 and tested on the cyberbullying dataset provided by Ptaszynski, Leliwa, Piech, and Smywinski-Pohl (2018) as a part of Zero-shot learning. The cyberbullying dataset was initially collected and prepared by Reynolds, Kontostathis, and Edwards (2011) from the Formspring data which consist of 12,728 data samples. Ptaszynski et al. further enhanced the dataset by various stages of manual annotation with highly trained data annotators with sufficient background for high quality annotations (Ptaszynski et al., 2018).

Our model was trained on the sarcastic dataset 2(without raw hashtags), and tested on the cyberbullying dataset using similar settings. The result attained an F-score of 0.881 which is comparable to the result of sarcastic dataset 2 with an F-score of 0.898 in Table 2. Interestingly, it was also much higher than models trained on purely cyberbullying-related data (Ptaszynski et al., 2018). However, comparing to Ptaszynski et al. model with 0.974 correct rate, our model has only achieved an accuracy of 0.896. This observation shows the prevalence of sarcasm in cyberbullying, and proves the practical applicability of sarcasm detection in other Natural Language Processing tasks.

6. Conclusions and future work

In this paper, we set out to explore the sarcasm and irony on Twitter, using various Natural Language Processing and Machine Learning techniques.

First we reviewed and clarified the definitions of irony and sarcasm by discussing various studies focusing on those terms. The review extended our knowledge of the definitions for the terms irony and sarcasm, which indicates an occurrence of a meaning shift between those terms throughout the modern days. The terms were originally strictly distinguishable, however, most researchers and social media users no longer differentiate clearly between them. Therefore, the review suggested that the term irony and sarcasm are being used interchangeably on social media in modern days.

Next, we conducted first experiment comparing between various types of classification methods including some popular classifiers for text classification task. The results of this experiment show that machine learning methods, especially deep learning methods, are rising in the latest trend by having the most potential for classification tasks. However, the downside of deep learning methods is the requirement for having a large dataset in order to achieve the best result. We also observed the importance of the social media markers (e.g., #hashtags in Twitter) which greatly impact the classification results.

For the second experiment, we compared between different types of data preprocessing methods with the classifier ranked best from the previous experiment based on the dataset with all hashtags removed, which is the Convolutional Neural Network. The findings from the results enhanced our understanding of data preprocessing where the best result came from the dataset with the least preprocessing methods applied. This is due to oversimplification of data which causes many vital and important features, on which irony and sarcasm detection heavily depended, being manipulated or removed. However, we observed that further data processing could still be crucial to the sensitivity of the results.

We also compared between sarcasm and irony utilizing their respective dataset from previous experiments. We trained models on sarcasm dataset and tested on irony dataset, and vice versa. Interestingly, the highest result attained an F-score of 0.94 which provided additional evidence with respect to the similarity between sarcasm and irony. The results also supported the claim that sarcasm is mostly a type of irony (aggressive irony).

Finally, we conducted a small experiment where model trained on sarcasm dataset was tested on a cyberbullying dataset. The result attained an F-score of 0.889 which is comparable to the result of sarcastic dataset itself. An implication of this is the possibility that there are preponderance of sarcasm in cyberbullying, and this extends our knowledge of the practical application of sarcasm detection in other tasks.

Further research might explore into larger variety of preprocessing methods for sarcasm and irony detection, or it would be interesting to assess the effects of irony and sarcasm detection in cyberbullying detection tasks.

CRediT authorship contribution statement

Zheng Lin Chia: Conceptualization, Methodology, Writing - original draft, Writing - review & edit. **Michal Ptaszynski:** Conceptualization, Methodology, Supervision. **Fumito Masui:** Conceptualization, Methodology, Supervision. **Gniewosz Leliwa:** Data curation. **Michal Wroczynski:** Data curation.

References

Abrams, M. H., & Harpham, G. G. (2009). *A glossary of literary terms*. Wadsworth Cengage Learning.

Amir, I., Wallace, B. C., Lyu, H., Carvalho, P., & Silva, M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of the 20th signll conference on computational natural language learning (CoNLL)*. Association for Computational Linguistics.

Attardo, S. (1999). Irony as relevant inappropriateness. *Journal of Pragmatics*.

Barbieri, F. (2017). *Machine learning methods for understanding social media communication: modeling irony and emojis*. Department DTIC.

Baziotis, C., Athanasiou, N., Papalampidi, P., Kolovou, A., Paraskevopoulos, G., Ellinas, M., et al. (2018). NTUA-SLP At semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive RNNs. In *Proceedings of the 12th international workshop on semantic evaluation(SemEval-2018)*. Association for Computational Linguistics.

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Research-paper recommender systems : a literature survey. *International Journal on Digital Libraries*.

Bouazizi, M., & Otsuki, T. (2016). A pattern-based approach for sarcasm detection on Twitter. In *Digital object*. IEEE Access.

Burfoot, C., & Baldwin, T. (2009). Automatic satire detection: Are you having a laugh. In *Proceedings of the ACL-IJCNLP 2009*. ACL and AFNLP.

Buschmeier, K., Cimiano, P., & Klinger, R. (2014). Am impact of analysis of features in a classification approach to irony detection in product reviews. In *Workshop on computational approaches to subjectivity, sentiment and social media*. Association for Computational Linguistics.

Chandrasekar, P., & Qian, K. (2016). The impact of data preprocessing on the performance of a naive Bayes classifier. In *IEEE 40th annual computer software and applications conference*. IEEE.

Davidov, D., Tsur, O., & Rappoport (2010). Semi-supervised recognition of sarcastic sentences in Twitter and amazon. In *Proceedings of the 14th conference on computational natural language learning*. Association of Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. Google AI Language.

Filatova, E. (2012). Irony and sarcasm: Corpus generation and analysis using crowdsourcing.

Fowler, H. W. (1926). *A Dictionary of Modern English*. Oxford University Press.

Galli, S. (2020). *Python feature engineering cookbook*. Packt.

Ghosh, A., & Veale, T. (2017). Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT 2016*. Association for Computational Linguistics.

Gonzalez-Ibanez, R., Muresan, S., & Wacholder, N. (2011). Identifyinng sarcasm in Twitterl: A closer look. In *Proceedings of the 49th annual meeting of the association for computational linguistics*. Assosiation for Computational Linguistics.

Grant, D., Hardy, C., Oswick, C., & Putnam, L. L. (2004). *The SAGE handbook of Organizational Discourse*. SAGE knowledge.

Hancock, J. T. (0000). Verbal irony use in face-to-face and computer-mediated conversations, Journal of Language and Social Psychology, SAGE journals, 204.

Hee, C. V. (2017). *Can machine sense irony? Exploring automatic irony detection on social media*. University Gent.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory.

Jalal, & Adeeb, A. (2018). Big data and intelligent software systems. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, IOS Press.

Jones, S. (2004). IDF Term weighting and IR research lessons. *Journal of Documentation*, Emerald Group.

Jorgensen, J. (1996). The functions of sarcastic irony in speech. *Journal of Pragmactics*, *26*(5), Elsevier.

Kalra, V., & Aggarwal, R. (2018). Importance of text data preprocessing and implementation in rapidminer. In *Proceedings of first international conference on information technology and knowledge management*. ICITKM.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Association for Computational Linguistics.

Kim, T., & Wurster, K. (2019). Emoji for python. https://pypi.org/project/emoji/.

Kreuz, R. J., & Glucksberg, S. (1989). How to be sarcstic: The echoic reminder theory of verbal irony. *Journal of Experimental Psychology*, American Psychological Association.

Kuhn, M., & Johnson, K. (2020). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.

Kumar, L., Somani, A., & Bhattacharyya, P. (2017). *Approaches for computational sarcasm detection: A survey*. ACM CSUR.

Kunneman, F., Liebrecht, C., Mulken, M. v., & Bosch, A. v. d. (2015). Signaling sarcasm: From hyperbole to hashtag. In *Information processing and management*. Elsevier.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proc of the IEEE*.

Lee, C. J., & Katz, A. N. (1998). The differential role of ridicule in sarcasm and irony. *Journal of Metaphor and Symbol*.

Li, X., & Ma, Z. (2016). Computational pragmatics: A survey in China and the world. In *NLPIR2018*. ACM.

Liebrecht, C., Kunneman, F., & den Bosch, A. V. (2013). The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*.

Maynard, D., & Greenwood, M. A. (2014). Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In *LREC 2014 proceedings*.

Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., & Turaga, D. (2017). Learning feature engineering for classification. In *Proceedings of 26th international joint conference on artificial intelligence*.

Oxford (2020). Definition of sarcasm. https://www.oxfordlearnersdictionaries.com/definition/english/sarcasm.

Papapicco, C., & Mininni, G. (2020). Twitter Culture: irony comes faster than tourist mobility. *Journal of Tourism Anad Cultural Change*, Taylor & Francis.

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

Porter, M. (2019). The porter stemming algorithm. https://tartarus.org/martin/PorterStemmer/.

Potamias, R. A., Siolas, G., & Gergios, A. (2019). A transformer-based approach to irony and sarcasm detection.

Ptaszynski, M., Dybala, P., Matsuba, T., Masui, F., Rzepka, R., Araki, K., et al. (2010). In the service of online order: Tackling cyber-bullying with machine learning and affect analysis. In *International Journal of Computational Linguistics Research*. Hokkaido University.

Ptaszynski, M., Eronen, J. K. K., & Masui, F. (2017). Learning deep on cyberbullying is always better than brute force. In *IJCAI 2017 3rd workshop on linguistic and cognitive approaches to dialogue agents (LaCATODA 2017)*.

Ptaszynski, M., Leliwa, G., Piech, M., & Smywinski-Pohl, A. (2018). *Cyberbullying Detection - Technical Report 2/2018*. Department of Computer Science AGH, University of Science and Technology.

Reyes, A., Rosso, P., & Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. In *Data & knowledge engineering*. Elsevier.

Reyes, A., Rosso, P., & Veale, T. (2013). *A multidimensional approach for detecting irony in twitter*. Lang Resources and Evaluation.

Reynolds, K., Kontostathis, A., & Edwards, L. (2011). Using machine learning to detect cyberbullying. In *2011 10th international conference on machine learning and applications and workshops*. IEEE.

Riloff, E., Qadir, A., Surve, P., Silva, L. D., Gilber, N., & Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceddings of the 2013 conference on empirical methods in natural language processing(EMNLP 2013)*. EMNLP.

Rosenthal, S., Ritter, A., Nakov, P., & Stoyanov, V. (2014). Semeval-2014 task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th international workshop on semantic evaluation*. SemEval 2014.

Sasaki, M., & Kita, K. (1998). Rule-based text categorization using hierarchical categories. In *Systems, man, and cybernetics conference*. IEEE.

Scott, S., & Matwin, S. (1999). Feature engineering for text classification. *Proceedings of the 16th international conference on machine learning*. ICML.

Shelley, C. (2001). The bicoherence theory of situational irony. In *Cognitive science, Vol. 25*. Elsevier.

Sulis, E., Fariaz, D. I. H., Rosso, P., & Patti, V. (2016). Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not. In *Knowledge-based systems*. Elsevier.

Tepperman, J. (2006). YEAH RIGHT: Sarcasm recognition for spoken dialogue system. In *Interspeech 2006*. ICSLP.

Tsur, O., Davidov, D., & Rappoport, A. (2010). ICWSM – A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the 4th international aaai conference on weblogs and social media*. Association for the Advancement of Artificial Intelligence.

Van Hee, C., Lefever, E., & Hoste, V. (2018). Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th international workshop on semantic evaluation (SemEval-2018)*. Association for Computational Linguistic.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *31st conference on neural information processing systems*. NIPS 2017.

Veale, T., & Hao, Y. (2010). Detecting ironic intent in creative comparisons. In *19th european conference on artificial intelligence*.

Wu, C., Wu, F., Wu, S., Liu, J., Yuan, Z., & Huang, Y. (2018). Thu_Ngn at semeval-2018 task 3: Tweet irony detection with densely connected LSTM and multi-task learning. In *Proceedings of the 12th international workshop on semantic evaluation(SemEval-2018)*. Association for Computational Linguistics.

Zhang, W., Taketoshi, Y., & Tang, X. (2011). A comparative study of tf*idf, LSI and multi-words for text classification. In *Expert Systems with Applications Vol. 38*. Elsevier.

Zhang, S., Zhang, X., Chan, J., & Rosso, P. (2019). Irony detection via sentiment-based transfer learning. In *Information Processing and Management*. Elsevier.

Zhang, M., Zhang, Y., & Fu, G. (2016). Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016*. 26th International Conference on Computational Linguistics.

Zheng, A., & Casari, A. (2018). *Feature enginnering for machine learning: Principles and techniques for data scientists*. O'reilly.