Abstract
Sarcasm is “a sharp, bitter, or cutting expression” used to convey contempt or to mock. Sarcasm is mainly distinguished by the modulation of pitch with which it is spoken and is largely context-dependent. Prevalence of sarcasm is one of the main challenges in sentiment analysis. Therefore, automatic processing of sarcasm has been widely explored in the past decade. Processing sarcasm using computational approaches is known as Computational Sarcasm Processing. Sarcasm processing has four major sub-domains: Sarcasm Detection, Sarcasm Generation, Sarcasm Interpretation and Sarcasm Target Identification. In this paper, we will summarize the past work that has been done in these sub-domains. We will discuss the persisting issues, the datasets available as well as the various approaches reported.

1 Introduction
Sarcasm is a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual. Sarcasm processing interests the sentiment analysis community due to the property that sarcastic texts have implied meaning opposite to the literal meaning. One of the initial works done in the domain of computational sarcasm processing was by Tepperman et al. (2006). Sarcasm emanates from incongruity which becomes apparent as the sentence unfolds. Joshi et al. (2015b) presented a computational system that harnesses context incongruity as a basis for sarcasm detection.

Over the years, a lot of research has been done in the domain of Sarcasm Detection which aims to classify if a given piece of text is sarcastic. However, other domains like Sarcasm Target Identification, Sarcasm Interpretation and Sarcasm Generation are very less explored. This is mostly because of lack of labelled data in these domains and this has motivated the need to explore the unsupervised approaches.

Detection of sarcasm is of great importance and beneficial to many NLP applications, such as sentiment analysis, opinion mining and advertising. Wallace (2015) presents a survey of linguistic challenges of computational irony. Joshi et al. (2016a) present a summary of previous works in automatic sarcasm detection. The major limitations of these works are: (i) the survey is either very long or (ii) it is primarily focused only on approaches for automatic sarcasm detection. However in this survey we will cover all the sub-domains of sarcasm processing.

The rest of the paper is organised as follows. We first talk about the linguistic perspective of sarcasm in section 2. Then we give a step-by-step overview of the past approaches, starting with the problem definitions in section 3. Section 4 describes the available datasets. The approaches along with their reported performance are discussed in section 5 and 6. We finally conclude the paper in section 7.

2 Linguistic study of sarcasm
The computational approaches to sarcasm use the linguistic theories as their foundation. Grice (1975) states that sarcasm is a form of metaphorical language in which the intended meaning is the opposite of the literal meaning.

2.1 Sarcasm vs Irony
Sarcasm and irony are closely related to each other. Sarcasm and irony are similar in that both are forms of reminder, yet different in that sarcasm
conveys ridicule of a specific victim whereas irony does not. The remark "What a sunny day!" uttered during a severe thunderstorm would be sarcastic if it brought to mind a specific weather forecaster’s prediction that it would be a sunny day, whereas it would be ironic if it brought to mind a wistful desire for sunny weather Lee and Katz (1998). The sarcasm-versus-irony classification problem has been reported in past work. Wang (2013) present a linguistic analysis to understand differences between sarcasm and irony. According to them, aggressiveness is the distinguishing factor between the two. Sulis et al. (2016) present a set of classifiers that distinguish between sarcasm and irony.

2.2 Sarcasm formulation
Ivanko and M. (2003) represent sarcasm as a 6-tuple consisting of <S,H,C,u,p,p'> where S = Speaker, H = Hearer/Listener, C = Context, u = Utterance, p = Literal proposition, p' = Intended proposition. The tuple can be read as ‘Speaker S generates an utterance u in Context C meaning proposition p but intending that hearer H understands p’’. For example- Given that a student hasn’t done his assignment, if the teacher says, “Wow, you have done a great job”, then the sentence is sarcastic and the 6-tuple would be as follows:
S = teacher
H = Student
C = Student has not done his assignment
u = Wow, you have done a great job”
p = the student did a good job
p' = the student didn a bad job.

2.3 Types of Sarcasm
Camp (2012) distinguishes four subclasses of sarcasm, individuated in terms of the target of inversion. (i) Propositional: When the text appears to be a proposition but has an implicit sentiment involved. For example- This phone should have been a paper weight. This sentence may be interpreted as non-sarcastic, if the context is not understood. (ii) Embedded: This type of sarcasm has incongruity embedded within the sentence. For example- I love being ignored. The congruity is embedded within the sentence by the presence of two opposite polarity words, love and ignored. (iii) Like-Prefixed: This form of explicit sarcasm prefixes the relevant sentence with ‘Like’ or ‘As if’ and employs a sneering tone. For example- Like that’s a good idea. (iv) While speaking ironically, a speaker does not undertake a genuine illocutionary speech act; rather, she mentions or pretends or ‘makes as if to say’ something, in order to express an evaluative attitude towards an associated thought or a perspective, and thereby draw attention to some discrepancy between how things are and how they should be. For example- changing the pitch, rolling of eyes etc.

2.4 Incongruity as a feature
Incongruity is defined as the state of being incongruous (i.e., lacking in harmony; not in agreement with principles). Incongruity is an essential component of sarcasm, and the possibility of incongruity in different degrees is at the heart of sarcasm. Ivanko and M. (2003) state that sarcasm/irony is understood because of incongruity. They studied the relationship between context incongruity and sarcasm processing (by humans). Riloff et al. (2013) identifies sarcasm that arises from the contrast between a positive sentiment referring to a negative situation. The incongruity within a sentence can be due to sentiments expressed within the sentence or due to the semantics of the sentence. Joshi et al. (2015b) has classified sentiment incongruity into two: (i) Explicit sentiment incongruity, where the incongruity arises due to two opposite sentiment polarity words being present within the sentence. For example- I love being ignored. (ii) Implicit sentiment incongruity, when incongruity occurs without the presence of sentiment words of both polarities. For example- I love this paper so much that I made a doggy bag out of it. The word ‘love’ has positive sentiment and the clause, I made a doggy bag out of it carries a sentiment opposite to the word ‘love’. Semantic incongruity can be explained using an example, “A woman needs a man like a fish needs a bicycle.” Joshi et al. (2016d) shows that word vector-based similarity/discordance is indicative of semantic similarity which in turn is a handle for context incongruity.

2.5 Sarcasm as a dropped negation
Irony/sarcasm is a form of negation in which an explicit negation marker is lacking. That is, even though the sentence doesn’t explicitly contain the negative marker ‘not’, the sentence is negative. For example, the sarcastic sentence ‘Being awake at 4 am with a headache is fun’ is equivalent to
the non-sarcastic sentence ‘Being awake at 4 am with a head-ache is not fun’. Dubey et al. (2019a) uses this idea to convert sarcastic sentences to their non-sarcastic interpretations.

3 Problem Definition

We will now look at the problem of sarcasm processing as discussed in the past work. We will explore four sub-domains, namely, Sarcasm Detection, Sarcasm Generation, Sarcasm Target Identification, and Sarcasm Interpretation. The field of Sarcasm Detection has been widely explored whereas other areas are relatively new.

3.1 Sarcasm Detection

Most approaches in the past have formulated Sarcasm Detection as a classification problem where given a piece of text, the aim is to predict whether the text is sarcastic or not. So, the sentence, “I love being ignored.” should be predicted as sarcastic while the sentence, “I love being acknowledged.” should be predicted as non-sarcastic. Other reported formulations include, detecting sarcasm dialogue as a sequence labeling task, as presented by Joshi et al. (2016c), where sarcasm labels are hidden variables which are to be predicted. Ghosh et al. (2015) modelled sarcasm detection as a sense disambiguation task. They stated that a word may have a literal sense and a sarcastic sense.

3.2 Sarcasm Generation

Sarcasm Generation is formulated as a task of automatic generation of sarcastic texts. Joshi et al. (2015a) defines sarcasm generation as the task of producing sarcastic sentences as a response to the user input which may or may not be sarcastic. They present a sarcasm generation module (SarcasmBot) for chatbots and mention that integrating a sarcasm generation module allows existing chatbots to become more ‘human’. Abhijit Mishra (2018) proposes a framework which takes a literal negative opinion as input and translates it into a sarcasm version.

3.3 Sarcasm Target Identification

Sarcasm Target Identification is formulated a task of identifying the target of ridicule in a sarcastic text. For example- "Tooth-ache is fun", the target is ’tooth-ache’.

3.4 Sarcasm Interpretation

Sarcasm interpretation is formulated as the generation of a non-sarcastic version of the text conveying the same message as the original sarcastic text. This is based on the theory of Sarcasm as dropped negation. Peled and Reichart (2017)), Dubey et al. (2019a) model sarcasm interpretation as a monolingual machine translation task. They define the purpose of the sarcasm interpretation task as the capability to generate a non-sarcastic utterance that captures the meaning behind the original sarcastic text.

4 Datasets

In this section, we will give an overview of the datasets used for experiments in computational sarcasm processing. These datasets can be classified on the basis of their languages as well as their lengths. With the advent of social media platforms like Twitter, Facebook, Reddit etc., data for a lot of researches is being collected from them. This is because a lot of people actively use such platforms and thus, there’s plenty of data available. Since, people can freely express their opinions on such platforms, a lot of user generated data on these platforms is sarcastic. This has specially attracted sarcasm processing community to collect datasets to train systems for computational sarcasm. Short text is characterized by situations where the length is limited. Twitter is a platform which allow users to post short texts upto 280 characters called tweets. The most popular choice of datasets for computational sarcasm are tweets because of the availability of the Twitter API, short length of tweets and the popularity of Twitter as a social media platform. All these factors makes Twitter an ideal platform for collecting datasets for computational sarcasm. These tweets can be collected manually or by using hashtag-based supervision. The hashtag-based approach is considered to be more reliable due to the assumption that the author knows the best about the sarcasm embedded within the text. Also, this approach is much easier and faster than the manual collection. González-Ibañez et al. (2011) use hashtag based approach to collect sarcastic tweets. Riloff et al. (2013), Maynard and Greenwood (2014), Mishra

There are a lot of datasets available in languages other than English. Liu et al. (2014) explore the characteristics of both English and Chinese sarcastic sentences and introduce a set of features specifically for detecting sarcasm in social media. Carvalho et al. (2009) dealt with irony in Portuguese newspapers. Liebrecht et al. (2013) designed a model to detect irony in Dutch Tweets. Andrea Gianzi and Caro (2012) collected and annotate a set of ironic examples from a common collective Italian blog. This corpus is also used in Bosco et al. (2013) for the study of sentiment analysis and opinion mining in Italian. Francesco Barbieri and Saggion (2014) present the first system for automatic detection of irony in Italian Tweets. Desai and Dave (2016) collect reviews from movie domain. They collect Hindi sentences which contain ‘#kataksh’ (word for sarcasm in Hindi) from online sources. Liebrecht et al. (2013) collect a dataset of around 78000 Dutch tweets. They collect tweets containing ‘#sarcisme’ marker, which means sarcasm in Dutch with the hashtag prefix. Lunando and Purwarianti (2013) introduce a dataset of Indonesian tweets from various topics like politics, food, movie, etc. Liu et al. (2014) create three datasets containing 3859, 5487, and 10356 comments respectively by crawling topic comments in Chinese language from different online sources.

Recently, cognitive datasets have also gained popularity. Mishra et al. (2017) propose a dataset that contains gaze data along with the text. Schifanella et al. (2016) collect data from three major social platforms that allow to post both text and images, namely Instagram (IG), Tumblr (TU) and Twitter (TW), using their available public APIs. Castro et al. (2019) propose a new sarcasm dataset, Multimodal Sarcasm Detection Dataset (MUSTARD), compiled from popular TV shows. MUSTARD consists of audiovisual utterances annotated with sarcasm labels. Each utterance is accompanied by its context of historical utterances in the dialogue.

Cross-cultural annotation: The annotation of dataset requires not only good linguistic background but also cultural knowledge. If the annotation is done by someone with different cultural background, it may result in sarcasm not being understood. Joshi et al. (2016b) used two datasets, one consisted of short texts (tweets) and another long texts (discussion forum posts), to analyse the impact of cultural differences on annotation. The datasets were originally labelled by American annotators. Two Indian annotators were asked to annotate the dataset. The difficulties in annotation faced were generally due to unfamiliar words, unknown contexts and unknown named entities.

5 Approaches

In this section, we discuss past approaches to process sarcasm using computational approaches. We categorise these approaches into three: rule-based, statistics-based and deep-learning based approaches.
5.1 Rule-based approaches

Rule-based approaches try to capture sarcasm in text using a set of rules. Riloff et al. (2013) present a bootstrapping algorithm that automatically learns lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. Maynard and Greenwood (2014) propose that hashtag sentiment is a key indicator of sarcasm. Hence, if the sentiment expressed by a hashtag does not agree with the rest of the tweet, the tweet is predicted as sarcastic. Dubey et al. (2019b) present a rule-based approach that considers noun phrases in the tweet as candidate contexts, and determines the optimal threshold of a numerical measure to predict sarcasm. Joshi et al. (2015a) implement eight rule-based approaches for generating different types of sarcasm. Depending upon the user input (question type, number of entities etc.), one of these eight rule-based approaches is chosen at run-time to generate a sarcastic reply. Joshi et al. (2018) present a rule-based extractor that takes as input the sarcastic sentence, and return a set of candidate sarcasm targets. Dubey et al. (2019a) model sarcasm as a form of dropped negation and present a rule-based approach for sarcasm interpretation.

5.2 Statistical approaches

Statistical approaches use mathematical models to infer the relationships between variables and obtain a general understanding of data to make predictions. Tepperman et al. (2006) use decision tree classifier with prosodic, contextual and spectral features to classify sarcasm. Riloff et al. (2013) use a SVM based approach with unigram and bigram features to classify sarcasm. They also combined this with their contrast based approach (discussed earlier) to improve recall further. Buschmeier et al. (2014) use feature set consisting of hyperbole, ellipsis, punctuation, interjections, emoticons, etc. to train a variety of classifiers including linear SVM, logistic regression, decision tree, random forest and naive bayes. Joshi et al. (2015b) use lexical and pragmatic features features based on two types of incongruity: implicit and explicit along with LibSVM and RBF kernel. Joshi et al. (2016d) use a SVM-based approach (SVM-perf) to identify sarcasm using word-embeddings similarity based features. Joshi et al. (2018) present a statistical extractor which uses a classifier that takes as input a word (along with its features) and returns if the word is a sarcasm target. They decompose the task into n-classification tasks, where ‘n’ is the total number of words in the sentence. This means that every word in input text is considered as an instance, such that the label can be 1 or 0, depending on whether or not the given word is a part of sarcasm target.

5.3 Deep-learning based approaches

Ghosh and Veale (2016) propose a neural network model composed of Convolution Neural Network(CNN) and followed by a Long short term memory (LSTM) network and finally a Deep neural network(DNN). Poria et al. (2016) use CNN-SVM (i.e., when the features extracted by CNN are fed to the SVM). The approach uses pre-trained convolutional neural network for extracting sentiment, emotion and personality features for sarcasm detection. Zhang et al. (2016) use a bi-directional gated recurrent neural network to capture syntactic and semantic information over tweets locally, and a pooling neural network to extract contextual features automatically from history tweets. Schifanella et al. (2016) propose a framework to detect sarcasm that integrate the textual and visual modalities. The method adapts a visual neural network initialized with unigrams as textual input and parameters trained on ImageNet to multimodal sarcastic posts. Hazarika et al. (2018) propose a Contextual SarCasm DEtector (CASCADE), which adopts a hybrid approach of both content- and context-driven modeling for sarcasm detection in online social media discussions. Since the sarcastic nature and form of expression can vary from person to person, CASCADE utilizes user embeddings that encode stylistic and personality features of users. Yi Tay (2018) classify the data using convolutional neural networks (CNN), recurrent neural networks (RNN) and a blend of these techniques to improve accuracy. Dubey et al. (2019b) propose a CNN based approach and an attention based model to detect sarcasm due to numbers. Abhijit Mishra (2018) present a framework that employs reinforced neural sequence to sequence learning and information retrieval and is trained only using unlabeled non-sarcastic and sarcastic opinions. Son et al. (2019) propose a deep learning model to detect sarcasm called sAtt-BLSTM convNet that is based on the hybrid of soft attention-based bidirec-
6 Reported performance

Table 1 summarises the performance reported by the past approaches. The performance in these works is calculated using different metrics and are on different datasets, hence, they are not directly comparable.

7 Conclusion

This paper presented the various problem definitions available in the domain of computational sarcasm processing. We presented a linguistic perspective of sarcasm and discussed existing linguistic theories. We explored the various datasets available in the domain. We observed that rule-based approaches are useful to get an insight into the problem. The rule-based approaches convey the crux of the sarcasm detection problem, namely, incongruity. The feature-based approaches uncover the indicators i.e., features of such sarcasm. However, a recent trend indicates that current state of the art models are deep learning-based that incorporate additional context beyond target text. We also looked at some language dependent approaches for sarcasm detection. Finally, we presented a comparison of past works along different dimensions, reported their performance.

References


<table>
<thead>
<tr>
<th>Research work</th>
<th>Dataset details</th>
<th>Approach</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sarcasm Detection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Riloff et al. (2013)</td>
<td>Tweets</td>
<td>Hybrid: Bootstrapped lexicon + SVM</td>
<td>F: 0.51</td>
</tr>
<tr>
<td>Buschmeier et al. (2014)</td>
<td>Reviews</td>
<td>Hyperbole, emoticons, intersection + Logistic regression</td>
<td>F: 74.4</td>
</tr>
<tr>
<td>Joshi et al. (2015b)</td>
<td>Tweets/ Discussion forum posts</td>
<td>Incongruity based features with SVM</td>
<td>F: 0.8876, F: 0.64</td>
</tr>
<tr>
<td>Schifanella et al. (2016)</td>
<td>Tumblr/ Instagram posts</td>
<td>Multimodal features + SVM</td>
<td>Acc: 89.7</td>
</tr>
<tr>
<td>Mishra et al. (2016b)</td>
<td>Tweets, Reviews</td>
<td>Cognitive features</td>
<td>F: 75.7</td>
</tr>
<tr>
<td>Joshi et al. (2016d)</td>
<td>Book snippets</td>
<td>Word embeddings based similarity with SVM-perf</td>
<td>F: 81.19</td>
</tr>
<tr>
<td>Ghosh and Veale (2016)</td>
<td>Tweets and reviews</td>
<td>CNN + LSTM + DNN</td>
<td>F: 0.921</td>
</tr>
<tr>
<td>Poria et al. (2016)</td>
<td>Tweets</td>
<td>CNN-SVM, pretrained CNNs on sentiment, emotion and personality based features</td>
<td>F: 97.7</td>
</tr>
<tr>
<td>Zhang et al. (2016)</td>
<td>Tweets</td>
<td>local + contextual features</td>
<td>Acc: 94.1, F: 90.26</td>
</tr>
<tr>
<td>Mishra et al. (2017)</td>
<td>Tweets</td>
<td>CNN + text and gaze features</td>
<td>F: 76.24</td>
</tr>
<tr>
<td>Son et al. (2019)</td>
<td>Twitter</td>
<td>sAtt-BLSTM convNet</td>
<td>Acc: 97.87</td>
</tr>
<tr>
<td>Dubey et al. (2019b)</td>
<td>Tweets having numbers</td>
<td>CNN</td>
<td>0.93</td>
</tr>
</tbody>
</table>

| Sarcasm Target Identification |
|-----------------------------|-------------------|-------------|-------------|

| Sarcasm Generation |
|-------------------|-----------------|-------------|-------------|
| Joshi et al. (2015a) | - | Rule-based generator | Acc: 87.09 |
| Abhijit Mishra (2018) | Tweets and short snippets | Reinforced neural sequence to sequence learning | Acc: 73.3 Fluency: 3.7 Adequacy: 3.8 |

| Sarcasm Interpretation |
|------------------------|-----------------|-------------|-------------|
| Peled and Reichart (2017) | Parallel monolingual MT | BLEU: 66.96 |
| Dubey et al. (2019a) | Corpus | 69.98 |
| Dubey et al. (2019a) | Parallel monolingual MT | BLEU: 67.96 |
| Dubey et al. (2019a) | Tweet | ROUGE: |

Table 1: Reported performance of past approaches along with the dataset details. Acc: Accuracy, F: F-score, DS: Dice score
impact analysis of features in a classification approach to irony detection in product reviews. pages 42–49, June.


