Abstract

Sarcasm is a form of verbal irony intended to express contempt or ridicule. Computational sarcasm refers to computational approaches to process sarcasm i.e. to detect, interpret and generate sarcasm. Research in sarcasm detection spans almost a decade. However, sarcasm interpretation and sarcasm generation are relatively new areas. [Tepperman et al., 2006] marked the onset of computational sarcasm research with an approach for detecting sarcasm in speech. Automatic sarcasm detection and interpretation is of great interest to the sentiment analysis community. In this paper, we present a concise survey of past approaches in computational sarcasm research for different languages, namely, English, Chinese, Dutch, Italian, Czech, Hindi and Indonesian respectively. We describe datasets, approaches, and issues in sarcasm detection, interpretation and generation respectively. To summarize past work in computational sarcasm research, we present a prominent table which presents a comparision along three different dimensions: (i) approaches (rule-based, statistical feature-based, deep learning-based), (ii) type of datasets, and (iii) reported performance values.

1 Introduction

Sarcasm is a cutting, often ironic remark intended to express contempt or ridicule\(^1\). Sarcasm is one of the most difficult challenge to sentiment analysis because it uses verbal irony to express contempt or ridicule, thereby, potentially confusing typical sentiment classifiers. Sarcasm is hard to interpret, especially non-verbal sarcasm. [Joshi et al., 2016b] show that sarcasm may not be understood by people from some cultures. Sarcasm expressed in a native language is difficult to interpret by non native speakers. In this paper, we present a survey of past approaches in automatic sarcasm detection in text for different languages. We also describe approaches for sarcasm interpretation and generation respectively.

In verbal communication, sarcastic utterances are accompanied by a certain tone of voice and facial expressions (For eg., rolling of eyes). However, in textual communication, these cues are absent which makes identification and interpretation of sarcasm very challenging even for humans. Sarcasm on the internet is hard to interpret because of the following reasons:

1. Speaker’s body language is unknown which is a major part of how people communicate with each other.
2. Tone of voice makes a huge difference. Words on a computer screen and face to face conversation are very different.
3. Many sentences can be sarcastic for a particular context.

All these factors make it difficult to interpret sarcasm. This is why understanding the actual meaning from a sarcastic utterance is a very interesting and challenging problem. Sarcasmic sentences may appear positive, negative or neutral on the surface. However, the implied sentiment is always negative. Consider the following sarcastic examples:

- ‘Visiting the doctor is so much fun!’
- ‘He performed terribly in the game anyway’ in response to the criticism of the best player in the game.
- ‘and I am the Prime Minister of India’

In the three sarcastic sentence above, the surface sentiment is positive, negative and neutral respectively. [Liu, 2012] states that sarcasm is a challenging task for sentiment analysis community because it is metaphorical in nature. Since sarcasm implies sentiment, it is crucial to detect and interpret sarcasm accurately in order to predict the correct sentiment of the text.

[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 focussed only on approaches for automatic sarcasm detection in English language. Therefore, in this paper, we address these issues and present a concise survey of past approaches for automatic sarcasm detection, interpretation and generation respectively. In addition to English, we also present approaches in computational sarcasm research focussing on other languages such as Chinese, Dutch, Italian, Czech, Hindi and Indonesian respectively. We observe that sarcasm interpretation and generation are relatively new areas and have limited research. The aim of this
paper is to present a concise summary of previous approaches in computational sarcasm research for different languages. We believe that our survey will allow new researchers to understand the state of the art in this domain.

The rest of the paper is organized as follows. We first present a linguistic perspective of sarcasm in Section 2. In Section 3, we present various problem formulations. We describe datasets, approaches and reported results in Section 4, 5 and 6 respectively. In Section 7, we discuss issues in computational sarcasm research. Finally, we conclude the paper in Section 8.

2 A Linguistic Perspective of Sarcasm

Before we embark on computational approaches to process sarcasm, in this section, we present linguistic theories related to sarcasm. [Grice, 1975] states that sarcasm is a form of metaphorical language in which the intended meaning is the opposite of the literal meaning. [Liebrecht et al., 2013] present a hypothesis investigating the extralinguistic equivalence between explicit markers such as hashtags and non-verbal cues that people employ in live interaction when conveying sarcasm. Sarcasm is based on well studied linguistic theories. We describe some of them below.

1. Incongruity: Incongruity is defined as "the state of being not in agreement, as with principles." Context incongruity is a necessary condition for sarcasm. [Turner, 1995] states that verbal irony is a technique of using incongruity to suggest a distinction between reality and expectation. Since sarcasm and irony are related, study of incongruity theory also helps in understanding sarcasm. There are two types of irony: verbal and situational. Verbal irony is expressed in words. The sentence, ‘I swear! I never swear. I detest the habit. What the devil do you mean?’ is an example of a verbal irony. On the other hand, situational irony arises due to a situation. For example, a situation where a fire station got burned down due to fire, is a situational irony. [Ivanko and Pexman, 2003] states that sarcasm is understood because of incongruity. Deriving from the notion of incongruity, [Joshi et al., 2015] define two types of incongruity in sarcasm that are analogous to two degrees of incongruity: (i) Explicit Incongruity: It is openly expressed by use of sentiment words of opposite polarities (For example, ‘I just love it when people ignore me!’ where there is a positive word ‘love’ and a negative word ‘ignore’). (ii) Implicit Incongruity: It is expressed using phrases of implied sentiment. Implicit incongruity is hard to detect as compared to explicit incongruity because the sentiment is hidden in a phrase. For example, “I love this paper so much that I made a doggy bag out of it”

2. Types of sarcasm: [Camp, 2012] describes four types of sarcasm: (i) Propositional: This type of sarcasm delivers an implication that is the contrary of a proposition that would have been expressed by a sincere utterance. For example, “Since you’re so enthusiastic, let's have you present the plan to the Dean at next week's meeting.” This sentence may be interpreted as non-sarcastic, if the context is not understood. (ii) Embedded: This type of sarcasm has an embedded incongruity in the form of words and phrases themselves. Embedded sarcasm is a fairly commonplace and flexible phenomenon. For example, “If you manage to generate one more half-baked, inconsequential idea like that, then you’ll get tenure for sure.” (iii) Like-Prefixed: This type of sarcasm explicitly uses ‘Like’ and ‘As if’ as prefixes. This inevitably includes the sentence’s focal content, and often only that content. For example, “Like that’s a good idea!” (iv) Illocutionary: The scope of this type of sarcasm encompasses not just some element within the uttered sentence, but the entire illocutionary act. It includes the entire speech act such as prosodic variations, hand gestures, eye movements, etc. For example, rolling one’s eyes when saying ‘You sure know a lot!’. In such cases, non-textual variations play a role. The examples above are from [Camp, 2012].

3. Sarcasm Representation: [Ivanko and Pexman, 2003] represent sarcasm as a 6-tuple consisting of <S, H, C, u, p, p’> where: S = Speaker, H = Listener, C = Context, u = Utterance, p = Literal Proposition, and p’ = Intended Proposition. This tuple can be read as: Speaker S generates an utterance u in Context C meaning proposition p but intending that hearer H understands the intended proposition p’.

4. Sarcasm as a dropped negation: [Giora, 1995] considers sarcasm as a form of dropped negation. [Joshi et al., 2016b] mention that when one expresses sarcasm, a negation is intended, without a explicit negation word like ‘not’. For example, the literal interpretation of the sarcastic sentence ‘headaches are fun’ is the non-sarcastic sentence ‘headaches are not fun’. Recently, [Dubey et al., 2019a] use this linguistic theory and propose a rule-based approach for converting sarcastic sentences into their non-sarcastic interpretation by simply applying an appropriate negation.

5. Irony, deception and humble bragging: Sarcasm and irony are related to each other. [Lee and Katz, 1998] state that sarcasm has an element of ridicule that irony does not. [Turner, 1995] states that the difference between literal proposition and deception lies in intention of the speaker while [Long and Graesser, 1988] state that the difference between sarcasm and deception lies in shared knowledge between speaker and listener. Another related phenomenon to sarcasm is humble bragging. For example, ‘I had to sign 500 autographs in the event, my life is miserable!’

\(^2\)This example is taken from [Joshi et al., 2015]
3 Different Problem Formulations

In this section, we describe how the problem of automatic sarcasm detection, interpretation and generation have been defined in the past work.

1. Sarcasm Detection: Automatic sarcasm detection is commonly formulated as a classification task. Given a text utterance, predict whether it is sarcastic or non-sarcastic. According to this formulation, the sentence, ‘I love headaches’ should be predicted as sarcastic whereas the sentence ‘I hate headaches’ should be predicted as non-sarcastic. However, other formulations also exist. For example, [Joshi et al., 2016c] model the problem of sarcasm detection as a sequence labelling task. [Ghosh et al., 2015] model sarcasm detection as a sense disambiguation task.

2. Sarcasm Interpretation: It is a relatively new area and is still evolving. The task of sarcasm interpretation is commonly formulated as the generation of a non-sarcastic utterance conveying the same message as the original sarcastic one. [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.

3. Sarcasm Generation: It is also a relatively new area. Automatic sarcasm generation in text refers to the task of producing sarcastic utterances. [Joshi, 2015] 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’.

4 Datasets

In this section, we describe datasets for computational sarcasm. We classify them into categories based on two dimensions: language (English vs other) and length of instances (short vs long) in the dataset.

A lot of user generated data on social media platforms like Twitter, Facebook, Reddit etc. is sarcastic. This has led researchers in computational sarcasm domain to use social media platforms to collect datasets to train systems for sarcasm detection, interpretation and generation. 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. [Dubey et al., 2019b] introduce a labelled dataset of tweets where sarcasm arises due to numbers. For example, ‘wow..from 305 to 255...significant discount’.

Dataset of 1254 reviews labelled with sarcasm which can be used for identifying sarcasm on two levels: a document and a text utterance (where a text utterance can be as short as a sentence and as long as a whole document). [Buschmeier et al., 2014b] and [Tsir et al., 2010] present a dataset of 1254 and 66000 Amazon reviews for sarcasm detection.

[Ptáček et al., 2014] made the first attempt at sarcasm detection in the Czech language. They create a Czech Twitter corpus of 7000 manually labelled tweets and provide it to the community. They also discuss and tackle issues that arises due to the rich morphological nature of the Czech language.

[Barbieri et al., 2014] present first automated system targeted to detect irony in Italian tweets. They introduce a corpus of 25450 tweets labelled with sarcasm. The set of ironic tweets in their dataset is an aggregation of the posts from popular Italian Twitter accounts which are known to post sharp satire on politics. They retrieve a set of non-ironic tweets from Twitter accounts of popular Italian daily newspapers.

[Liebrecht et al., 2013] collect a dataset of around 78000 Dutch tweets. They collect tweets containing ‘hashtag’ marker, which means sarcasm in Dutch with the hashtag prefix. To enhance the quality of their dataset, they manually annotate a sample and report that 85% of these tweets are indeed sarcastic.

[Desai and Dave, 2016] collect reviews from movie domain. They collect Hindi sentences which contain ‘#kataksh’ (word for sarcasm in Hindi) from online sources. The dataset also consists of Hindi tweets translated from English tweets with help of language experts and polarity labelled corpus of Hindi sentences [Joshi et al., 2010], they generate a total of 1410 sarcastic sentences. [Swami et al., 2018] present a dataset of 5250 English-Hindi co-mixed tweets out of which 504 tweets are marked as sarcastic. Each tweet is labelled with sarcasm and each token is also annotated with a language tag. They collect sarcastic and non-sarcastic tweets using hashtags and manually select English-Hindi code-mixed tweets from them. Each tweet is manually annotated for presence of sarcasm.

[Lunando and Purwarianti, 2013] introduce a dataset of Indonesian tweets from various topics like politics, food, movie, etc. The training set contains 980 tweets out of which 502 are neutral, 250 are sarcastic and 228 are non-sarcastic. The testing set contains 300 tweets out of which 200 are neutral, 60 are sarcastic and 40 non-sarcastic.

[Peled and Reichart, 2017] present a parallel corpora of 150000 sarcastic tweets with their non-sarcastic interpretation for the task of automatic sarcasm interpretation. They divide

http://sempub.tahn.upf.edu/tw/clicit2014/

http://twiqs.nl/
the corpus into three parts: 12000 train, 1500 development and 1500 test. Informed by linguistic theories, [Karou et al., 2017] propose a multi-layered annotation schema for irony and its application to a corpus of French, English and Italian tweets. [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. They also present specific characteristics of sarcasm in Chinese language.

5 Approaches
In this section, we describe past approaches in computational sarcasm research. We classify them into three categories pertaining to three paradigms of NLP: rule-based, statistical feature-based and deep learning-based approaches.

5.1 Rule-based Approaches
[Maynard and Greenwood, 2014] perform an analysis of the effect of sarcasm on the polarity of tweets. They have compiled a number of rules for comparing sentiment expressed by a hashtag and rest of the tweet to predict sarcasm. [Dubey et al., 2019a] model sarcasm as a form of dropped negation and present a rule-based approach for sarcasm interpretation. They maintain a list of negation words and associate them with verbs present in the sarcastic utterance, thereby, producing the non-sarcasm sentence. [Riloff et al., 2013] look for contrast between positive verb and negative situation phrase in a sentence. [Bharti et al., 2015] use a phrase-based lexicon generation algorithm. They present a rule-based approach which predicts the sentence as sarcastic if a positive sentence contains a negative phrase. [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, 2015] implements 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.

5.2 Feature-based Approaches
In this section, we describe a set of features and the corresponding statistical classifiers for computational sarcasm. Table 1 summarizes the popular features. Most of the past approaches use features related to the text: (i) Unigrams, bag-of-words and Pragmatic features, (ii) Stylistic patterns and patterns related to situational disparity, and (iii) Hastag. However, recent approaches show improvements in performance by incorporating contextual features (features that use information beyond the target text).

[Mishra et al., 2016] propose a different approach and augment the feature vector with cognitive features extracted from eye movement patterns of human readers. They use a set of gaze-based features such as average fixation duration, regression count and skip count. [González-Ibáñez et al., 2011] state that incorporating sentiment and emoticon related features also improve the performance of sarcasm detection systems. Past work using the described features commonly use variants of Support Vector Machines (SVM). Naive Bayes and ensemble methods like Bagging, Boosting etc., have also been reported in the past. [Lunando and Purwarianti, 2013] propose a feature augmentation based approach to enhance sentiment analysis in Indonesian language by applying sarcasm detector on top of sentiment classifier. They incorporate features related to negativity information and the number of interjection words. [Liu et al., 2014] propose a language specific feature-based approach to detect sarcasm in Chinese. They use language independent features: punctuation, recurring sequences and semantic imbalance rate. They also use language dependent features: rhetoric-based, homophony-based and construction-based. They use ensemble-based strategy to make prediction.

5.3 Deep Learning-based Approaches
End-to-end deep learning architectures are very popular for solving NLP problems these days. Recently, there is a rise in deep learning-based approaches for computational sarcasm. [Ghosh and Veale, 2016] propose a semantic model which is a combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for sarcasm detection. They show an improvement over recursive SVM by using their approach. [Poria et al., 2016] propose a novel CNN-based architecture to detect sarcasm. [Amir et al., 2016] propose a novel CNN-based architecture to learn additional context in the form of form of user embeddings and use that for sarcasm detection. [Zhang et al., 2016] use a bi-directional GRU followed by a pooling neural network to detect sarcasm. [Ghosh and Veale, 2017] propose a neural architecture that considers speaker’s mood for sarcasm detection. [Dubey et al., 2019b] present two deep learning-based architectures for detecting sarcasm arising due to numbers in tweets. Recently, [Riloff et al., 2018] propose a hybrid approach incorporating content, context and user embeddings for detecting sarcasm in online discussions. [Peled and Reichart, 2017] introduce the task of sarcasm interpretation. They use monolingual machine translation-based approach and present two systems: (i) RNN-based and (ii) MOSES-based, to obtain non-sarcastic interpretations of sarcastic tweets. [Dubey et al., 2019a] use three deep learning-based models for the task of sarcasm interpretation: (i) Encoder-Decoder Network, (ii) Attention Network, and (iii) Pointer Generator Network.

6 Reported Results
Table 1 presents the performance of popular/influential past works in computational sarcasm research along with dataset type and features/architectures used. Since different approaches use different experimental setup, datasets, pre-processing techniques and performance metrics, they are not directly comparable. However, this table provides a rough quantitative estimate of the present state of computational sarcasm research.

7 Issues in Computational Sarcasm
In this section, we discuss the three prominent issues in computational sarcasm research.
<table>
<thead>
<tr>
<th>Past Work</th>
<th>Details</th>
<th>Features/Architectures</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Rakov and Rosenberg, 2013]</td>
<td>Speech Data</td>
<td>Unigrams + Intensity Bigrams</td>
<td>Acc: 81.57</td>
</tr>
<tr>
<td>[Riloff et al., 2013]</td>
<td>Tweets</td>
<td>Contrast between positive verbs and negative situation phrases</td>
<td>F: 0.51</td>
</tr>
<tr>
<td>[Liebrecht et al., 2011]</td>
<td>Tweets</td>
<td>n-grams, emotions, intensifiers</td>
<td>AUC: 0.79</td>
</tr>
<tr>
<td>[Lunano and Pauri, 2013]</td>
<td>Tweets</td>
<td>Negativity information, interaction words</td>
<td>Acc: 54.1</td>
</tr>
<tr>
<td>[Püück et al., 2014]</td>
<td>Tweets</td>
<td>n-grams, POS tag, emotions, word-case</td>
<td>F: 56.9</td>
</tr>
<tr>
<td>[Buscemi et al., 2014]</td>
<td>Reviews</td>
<td>Hyperbole, emotions, interaction words</td>
<td>F: 71.7</td>
</tr>
<tr>
<td>[Barbieri et al., 2014]</td>
<td>Tweets</td>
<td>BoW, POS, word frequency-based, synonym-based, sentiment-based</td>
<td>F: 76, P: 75, R: 76</td>
</tr>
<tr>
<td>[Li et al., 2014]</td>
<td>Comments</td>
<td>Punctuation, rhetoric, homophony</td>
<td>AUC: 89.7</td>
</tr>
<tr>
<td>[Joshi et al., 2015]</td>
<td>Tweets, Discussion Forum</td>
<td>Implicit &amp; Explicit Incongruity-based</td>
<td>F: 88.76/64</td>
</tr>
<tr>
<td>[Wallace et al., 2015]</td>
<td>Reddit</td>
<td>Sentiment-based, subreddit-based, noun phrases</td>
<td>P: 14.1, R: 37.7</td>
</tr>
<tr>
<td>[Bharti et al., 2015]</td>
<td>Tweets</td>
<td>Parsing-based approach</td>
<td>F: 90, P: 85, R: 96</td>
</tr>
<tr>
<td>[Mishra et al., 2016]</td>
<td>Tweets, Reviews</td>
<td>Cognitive features</td>
<td>F: 75.7, P: 76.5, R: 75.3</td>
</tr>
<tr>
<td>[Joshi et al., 2016]</td>
<td>Book snippets</td>
<td>Word embedding similarity-based</td>
<td>F: 80.47</td>
</tr>
<tr>
<td>[Ghosh and Veale, 2016]</td>
<td>Tweets, Reviews</td>
<td>CNN, LSTM, DNN</td>
<td>F: 92.1</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>[Amir et al., 2016]</td>
<td>Tweets</td>
<td>BoW, author-based, n-grams local + contextual features</td>
<td>Acc: 87.2</td>
</tr>
<tr>
<td>[Zhang et al., 2016]</td>
<td>Tweets</td>
<td>Word &amp; Character n-grams, emoticons</td>
<td>Acc: 94.1, F: 90.26</td>
</tr>
<tr>
<td>[Pele and Rechtan, 2017]</td>
<td>Parallel Tweet Corpus</td>
<td>monolingual MT</td>
<td>BLEU: 66.96, ROUGE: 69.98</td>
</tr>
<tr>
<td>[Dubey et al., 2019]</td>
<td>Tweets containing numbers</td>
<td>CNN, Attention Network</td>
<td>F: 93, F: 91</td>
</tr>
<tr>
<td>[Hazarka et al., 2018]</td>
<td>Discussion Forum</td>
<td>Hybrid context driven approach using CNN Word &amp; Character n-grams, emoticons</td>
<td>Acc: 79, F: 86</td>
</tr>
<tr>
<td>[Swami et al., 2018]</td>
<td>English-Hindi Code Mixed Tweets</td>
<td>Word embedding similarity-based</td>
<td>F: 78.4</td>
</tr>
<tr>
<td>[Dubey et al., 2019a]</td>
<td>Parallel Tweet Corpus</td>
<td>monolingual MT</td>
<td>BLEU: 67.96, ROUGE: 68.81</td>
</tr>
</tbody>
</table>

Table 1: Performance of popular/influential past works in computational sarcasm research along with dataset type and features/architectures used. P → Precision, R → Recall, F → F-score, AUC → Area Under the Curve, BoW → Bag of Words, MT → Machine Translation, * → Sarcasm Interpretation Task

1. Language Morphology: Incorporating language specific features improve the performance of sarcasm detection systems [Püück et al., 2014]. Current approaches in computational sarcasm research is for English. Moreover, most of the feature-based approaches are language independent and do not take language specific characteristics into account. However, when these approaches are used on morphologically rich languages (Slavic languages, Dravidian languages etc.), they perform poorly. This opens up the possibility of designing and incorporating language specific features for sarcasm detection, interpretation and generation.

2. Dataset annotation: A lot of current sarcasm detection and interpretation systems are trained on datasets of tweets extracted using #sarcasm hashtag. However, a lot of tweets collected using this approach can be interpreted as non-sarcastic if the context is not understood. Hence, to enhance the quality of datasets, manual annotation is usually necessary. Since sarcasm is a phenomenon which is hard to understand even by humans, the quality of manual annotation is also a concern. The inter-annotator agreement values are diverse ranging from 0.34 in [Tsur et al., 2010] to 0.81 in [Riloff et al., 2013]. [Joshi et al., 2016b] study the understanding of sarcasm across different cultures. They present a comparison between sarcasm understanding of Indian and American annotators. Their study shows the importance of context in computational sarcasm. A recent trend is to validate on multiple datasets annotated manually as well as using hashtags.

3. Choice of performance metrics: We observe the skew in the datasets between sarcastic and non-sarcastic class because sarcasm is an infrequent phenomenon. For example, [Barbieri et al., 2014] introduce a dataset of Italian tweets, only 12.5% of which are sarcastic. Due to this skewness, selection of proper performance metrics is crucial (For example, micro vs macro F-score or AUC since it is a more reliable metric for unbalanced datasets).

8 Conclusion

This paper presented definitions, datasets, approaches, performance values, issues and recent trends as reported in the past work in computational sarcasm research. We presented a linguistic perspective of sarcasm and discussed existing linguistic theories. 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 uncovers 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 and discussed prominent issues in computational sarcasm research.

References


in Social Networks Analysis and Mining 2015, ASONAM ’15, pages 1373–1380, New York, NY, USA, 2015. ACM.


[Joshi et al., 2016b] Aditya Joshi, Pushpak Bhattacharyya, Mark James Carman, Jaya Saraswati, and Rajita Shukla. How do cultural differences impact the quality of sarcasm annotation?: A case study of indian annotators and american text. In LaTeCH@ACL, 2016.


[Rakov and Rosenberg, 2013] Rachel Rakov and Andrew Rosenberg. ”sure, i did the right thing”: a system for sarcasm detection in speech. In INTERSPEECH, 2013.


