Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2939334, 13 pages
https://doi.org/10.1155/2021/2939334

Research Article

Explainable Artificial Intelligence for Sarcasm Detection in Dialogues

Akshi Kumar[1,2] **Shubham Dikshit**[3] **and Victor Hugo C. Albuquerque**[1,4]

[1]*Graduate Program on Telecommunication Engineering, Federal Institute of Education, Science and Technology of Ceará, Fortaleza, CE, Brazil*
[2]*Department of Computer Science and Engineering, Delhi Technological University, Delhi, India*
[3]*Department of Computer Science and Engineering, IMS Engineering College, Ghaziabad, India*
[4]*Graduate Program on Teleinformatics Engineering, Federal University of Ceará, Fortaleza, CE, Brazil*

Correspondence should be addressed to Akshi Kumar; akshikumar@dce.ac.in

Received 17 May 2021; Revised 11 June 2021; Accepted 21 June 2021; Published 3 July 2021

Academic Editor: Mohammad R. Khosravi

Sarcasm detection in dialogues has been gaining popularity among natural language processing (NLP) researchers with the increased use of conversational threads on social media. Capturing the knowledge of the domain of discourse, context propagation during the course of dialogue, and situational context and tone of the speaker are some important features to train the machine learning models for detecting sarcasm in real time. As situational comedies vibrantly represent human mannerism and behaviour in everyday real-life situations, this research demonstrates the use of an ensemble supervised learning algorithm to detect sarcasm in the benchmark dialogue dataset, MUStARD. The punch-line utterance and its associated context are taken as features to train the eXtreme Gradient Boosting (XGBoost) method. The primary goal is to predict sarcasm in each utterance of the speaker using the chronological nature of a scene. Further, it is vital to prevent model bias and help decision makers understand how to use the models in the right way. Therefore, as a twin goal of this research, we make the learning model used for conversational sarcasm detection interpretable. This is done using two post hoc interpretability approaches, Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive exPlanations (SHAP), to generate explanations for the output of a trained classifier. The classification results clearly depict the importance of capturing the intersentence context to detect sarcasm in conversational threads. The interpretability methods show the words (features) that influence the decision of the model the most and help the user understand how the model is making the decision for detecting sarcasm in dialogues.

1. Introduction

Natural language is a vital information source of human sentiments. Automated sarcasm detection is often described as a natural language processing (NLP) problem as it primarily requires understanding the human expressions, language, and/or emotions articulated via textual or nontextual content. Sarcasm detection has attracted growing interest over the past decade as it facilitates accurate analytics in online comments and reviews [1, 2]. As a figurative literary device, sarcasm makes use of words in a way that deviates from the conventional order and meaning thereby misleading polarity classification results. For example, in a statement "*Staying up till 2:30am was a brilliant idea to miss my office meeting*," the positive word "*brilliant*" along with the adverse situation "*miss my office meeting*" conveys the sarcasm, because sarcasm has an implied sentiment (negative) that is different from surface sentiment (positive due to presence of "brilliant"). Various rule-based, statistical, machine learning, and deep learning-based approaches have been reported in pertinent literature on automatic sarcasm detection in single sentences that often rely on the content of utterances in isolation. These include a range of techniques such as sense disambiguation [3] to polarity flip detection in text [4] and multimodal (text +image) content [5, 6]. Furthermore, its use on social media platforms like Twitter and Reddit is primarily to convey user's frivolous intent, and therefore, the dialect is more casual and includes the use of microtext like

wordplay, neologism, emojis, and slangs. Few recent works have taken into account the additional contextual information along with the utterance to deal with these challenges. Different researchers have considered varied operational cues to typify context. In 2019, Kumar and Garg [7] defined five broad categories of context, namely, social-graph, temporal, content, modality, and user-profile based which can be used for improving the classification accuracy. Evidently, it is essential to capture the operational concern, that is, the pragmatic meaning defined by "context" as sarcasm. But the use of sarcasm in dialogues and conversational threads has further added to the challenges making it vital to capture the knowledge of the domain of discourse, context propagation during the course of dialogue, and situational context and tone of the speaker. For example, recently, several Indian airlines took to Twitter to engage users in a long thread meant to elicit laughs and sarcastic comebacks amid the coronavirus lockdown that has kept passengers and airlines firmly on the ground. IndiGo playfully teased its rivals by engaging in a Twitter banter resulting in comic wordplays on airlines' advertising slogans. IndiGo began by asking Air Vistara "*not flying higher?*" in reply to which the airlines tagged peer GoAir, punning on its tagline "fly smart" and what followed was other key airlines like AirAsia and SpiceJet joining the thread exchange equipped with witty responses using each other's trademark business taglines (https://www.deccanherald.com/business/coronavirus-indigo-vistara-spicejet-engage-in-banter-keep-twitterati-in-splits-amid-lockdown-blues-823677.html).

As seen in Figure 1, it is not only important to capture the intrasentence context but the intersentence context too to detect sarcasm in conversational threads. Moreover, the sarcastic intent of the thread is difficult to comprehend without the situational context as in this case is the unprecedented travel restrictions, including the grounding of all domestic and international passenger flights, to break the chain of the coronavirus disease (COVID-19) transmission.

But as sarcasm is a convoluted form of expression which can cheat and mislead analytic systems, it is equally important to achieve high prediction accuracy with decision understanding and traceability of actions taken. As models cannot account for all the factors that will affect the decision, explainability can account for context and help understand the included factors that will affect decision making so that one can adjust prediction on additional factors. Explainable artificial intelligence (XAI) [8, 9] is the new buzzword in the domain of machine learning which intends to justify the actions and understand the model behaviour. It enables building robust models with better decision-making capabilities.

Thus, in this paper, we firstly demonstrate the role of context in conversational threads to detect sarcasm in the MUStARD dataset [5], which is a multimodal video corpus for research in automated sarcasm discovery compiled using dialogues from famous sitcoms, namely, "Friends" by Bright, Kauffman, Crane Productions, and Warner Bros. Entertainment Inc., "The Big Bang Theory" by Chuck Lorre, Bill Prady, CBS, "The Golden Girls" by Susan Harris, NBC, and "Sarcasmaholics Anonymous." The data is labelled with true

Figure 1: Online sarcastic conversational thread.

and false for the sarcastic and nonsarcastic dialogues using the sequential nature of scenes in the episodes, and we use eXtreme Gradient Boosting (XGBoost) method [10] to primarily investigate how conversational context can facilitate automatic prediction of sarcasm. As a twin goal of this research, we aim to make the supervised learning models used for conversational sarcasm detection interpretable with the help of XAI. The goal is to show the words (features) that influence the decision of the model the most.

Using dialogue dataset from sitcoms can invariably relate to any real-life utterance making this work relevant for various sentiment analysis-based market and business intelligence applications for assessing insights from conversational threads on social media. Most situational comedies or sitcoms are led by the comedy of manners, vaudeville, and our tacit perceptions of everyday life. These are the story of our psychodynamics and sociodynamics on situations that could arise in everyday life and unfold the unexpected and ironic comedy of human behaviour in real-life situations. For example, in Friends, season 10, episode 3, Ross walks in with a clearly overdone tan to the point that his skin color is very dark and looks truly ridiculous. He tells Chandler that he went to the tanning place his wife (Monica) suggested. And Chandler came up with a sarcastic statement "*Was that place the sun?*" as it looked like the only tanning place that could make someone's skin look like that would be sitting directly beneath the scorching sun! The sarcasm in Chandler's dialogue could only be understood considering the entire conversation and not taking his dialogue in isolation (Figure 2).

XAI in a typical NLP task setting can offer twofold advantages, namely, transferability, as machine learning models are trained in a controlled setting, deployment in real time should also ensure that the model has truly learned to detect underlying phenomenon, and secondly, it can help determining the contextual factors that affect the decision. The terms interpretability and explainability are often used

Figure 2: Friends: season 10, episode 3.

interchangeably as both play a complementary role in understanding predictive models [11]. The term interpretability tells us what is going on in the algorithm, i.e., it enables us to predict what will happen if there are some changes in the parameters or input, and explainability tells the extent to which the internal working of any machine learning or deep learning model can be explained in human terms. Characteristically, interpretable machine learning systems provide explanations for their outputs. According to Miller [12], interpretability is defined as the capability to understand the decision and means that the cause and effect can be determined. Interpretable machine learning (ML) describes the process of revealing causes of predictions and explaining a derived decision in a way that is understandable to humans. The ability to understand the causes that lead to a certain prediction enables data scientists to ensure that the model is consistent with the domain knowledge of an expert. Furthermore, interpretability is critical to obtain trust in a model and to be able to tackle problems like unfair biases or discrimination. One way to apply interpretable ML is by using models that are intrinsically interpretable and known to be easy for humans to understand such as linear/logistic regression, decision trees, and K-nearest neighbors [13]. Alternatively, we can train a black-box model and apply post hoc interpretability techniques [14] (Figure 3) to provide explanations.

In this paper, we use two post hoc model agnostic explainability techniques called Local Interpretable Model-agnostic Explanations (LIME) [15, 16] and Shapley Additive exPlanations (SHAP) [17, 18] to analyze the models on the dataset by checking the evaluation metrics and select the model where explanation can be separated from the models. The intent is to evaluate the black-box model much easily on how each word plays an important role in the prediction of the sarcastic dialogues by the speaker using the sequential nature of a scene in the TV series. Thus, the key contributions of this research are as follows:

	Global	Local
Model-specific	Model internals; Intrinsic feature importance	Rule sets (Tree structure)
Model-agnostic	Partial dependence plots; Feature importance; Global surrogate models	Individual conditional expectation; Local surrogate models

Figure 3: Post hoc interpretability techniques.

(i) Using sequence of utterances to detect sarcasm in real-time dialogues

(ii) Using post hoc model-agnostic local surrogate machine learning interpretability methods to comprehend which words within a dialogue are the most important for predicting sarcasm

The scope of the research can be extended to real-time AI-driven sentiment analysis for improving customer experience where these explanations would help the service desk to detect sarcasm and word importance while predicting sentiment. The organization of the paper is as follows: the next section briefs about the taxonomy of machine learning interpretability methods followed by related work within the domain of sarcasm detection specifically in conversational data in Section 3. Section 4 discusses the key techniques used in this research followed by the results and conclusion in Section 5 and Section 6, respectively.

2. Taxonomy of Machine Interpretability Methods

Artificial intelligence (AI) is gradually participating in day-to-day experiences. Its entrusted adoption and encouraging acceptance in various real-time domains are highly contingent upon the transparency, interpretability, and explainability of models built. Particularly in customer-centric environments, trust and fairness can help customers achieve better outcomes. Introduced in the early 1980s, XAI is a framework and tool which helps humans to understand the model behaviour and enables building robust models with better decision-making capabilities. It is used for understanding the logic behind the predictions made by the model and justifies its results to the user.

A trade-off between the model interpretability and predictive power is commonly observed as shown in Figure 4. As the model gets more advanced, it becomes harder to explain how it works. High interpretability models include traditional regression algorithms (linear models, for example), decision trees, and rule-based learning. Basically, these are approximate monotonic linear functions. On the other hand, low interpretability models include ensemble methods and deep learning where the black-box feature extraction offers poor explainability.

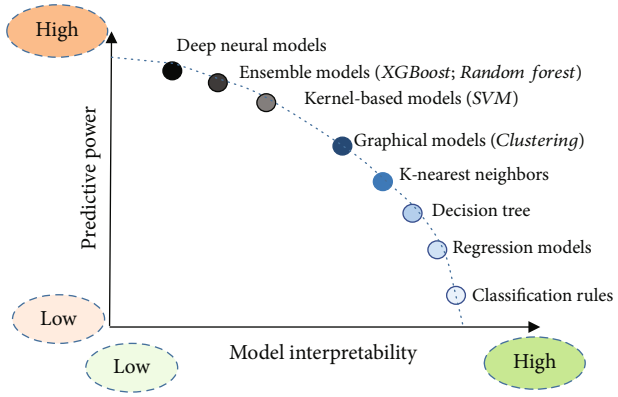

Figure 4: Predictive power vs. interpretability trade-off.

Machine interpretability methods are often categorized along three main criteria [19, 20]. The first discriminates based on the coverage of explanation as local or global for explanation for at instance-level (individual predictions) or model-level (entire model), respectively. Global interpretability methods explain the entire ML model at once from input to prediction, for example, decision trees and linear regression. Local interpretability methods explain how predictions change for when input changes and are applicable for a single prediction or a group of predictions. The second criteria differentiate between the explanations based on the interpretable design capabilities as intrinsically interpretable models and post hoc models (Figure 5). Intrinsically interpretable models are models that are interpretable by design, and no postprocessing steps are needed to achieve interpretability. These are self-explaining, and explainability is often achieved as a by-product of model training. On the other hand, in post hoc methods, explainability is often achieved after the model is trained and it requires postprocessing using external methods to achieve interpretability.

The third criterion to categorize interpretability methods is the applicability limitation to specific models or any ML model. Based on these criteria, the methods are divided into model-specific and model-agnostic methods. Model-specific techniques can be used for a specific architecture and require training the model using a dataset. Intrinsic methods are by definition model-specific. On the contrary, model-agnostic methods can be used across many black-box models without considering their inner processing or internal representations and do not require training the model. Post hoc methods are usually model-agnostic.

Post hoc interpretability methods consider interpretability of predictions made by black-box models after they have been built. These can further be categorized into four categories as surrogate models, feature contribution, visualisations, and case-based methods [19, 21]. Figure 6 shows the key model-agnostic methods available in literature [14].

In this work, we use two popular Python libraries, SHAP and LIME, to interpret the output and leverage model explanations.

3. Related Work

There is notable literary evidence apropos the versatile use of machine learning and deep learning algorithms for automated sarcasm detection. In the past, rule-based algorithms were employed initially to detect sarcasm [22]. Later, many researchers [23–29] used ML algorithms to detect sarcasm in textual content. Naive Bayes and fuzzy clustering models were employed by Mukherjee et al. [30] for sarcasm detection in microblogs. The researchers concluded that Naive Bayes models are more effective and relevant than the fuzzy clustering models. Prasad et al. [31] analyzed and compared various ML and DL algorithms to conclude that gradient boost outperforms the other models in terms of accuracy. In 2018, Ren et al. [32] employed contextual information for sarcasm detection on Twitter dataset by utilizing two different context-augmented neural models. They demonstrated that the proposed model performs better than the other SOTA models. In 2019, Kumar and Garg [33] compared various ML techniques like SVM, DT, LR, RF, KNN, and NN for sarcasm detection on Twitter and Reddit datasets. A hybrid deep learning model of soft attention-based bi-LSTM and convolution neural network with GloVe for word embeddings was proposed by Kumar et al. [34]. The results demonstrated that the proposed hybrid outperforms CNN, LSTM, and bi-LSTM. Kumar and Garg [4] reported a study on context-based sarcasm detection on Twitter and Reddit datasets using a variety of ML techniques trained using tf-idf and DL techniques using GloVe embedding.

Recent studies have also been reported on multimodal sarcasm detection. In 2019, Cai et al. [35] used bi-LSTM for detection of sarcasm in multimodal Twitter data. In the same year, Kumar and Garg [6] employed various supervised ML techniques to study context in sarcasm detection in typographic memes and demonstrated that multilayer perceptron is best among all the models. In 2020, a study by Kumar et al. [36] built a feature-rich support vector machine and proposed a multihead attention-based bi-LSTM model for sarcasm detection in Reddit comments. Few studies on sarcasm detection in online multilingual content have also been reported. In 2020, Jain et al. [2] had put forward a hybrid of bi-LSTM with softmax attention and CNN for sarcasm detection in multilingual tweets. In 2021, Farha et al. [37] compared many transformer-based language models like BERT and GPA on Arabic data for sarcasm detection. Faraj et al. [38] proposed a model based on ensemble techniques with an AraBERT pretrained model for sarcasm detection in Arabic text with an accuracy of 78%.

Sarcasm detection in conversations and dialogues has created a great interest with NLP researchers. Ghosh et al. [39] used conditional LSTM and LSTM with sentence-level attention to understand the role of context in social media discussions. Hazarika et al. [40] proposed a CASCADE (a ContextuAl SarCasm DEtector) model which extracted contextual information from online social media discussions on Reddit to detect sarcasm by taking into consideration

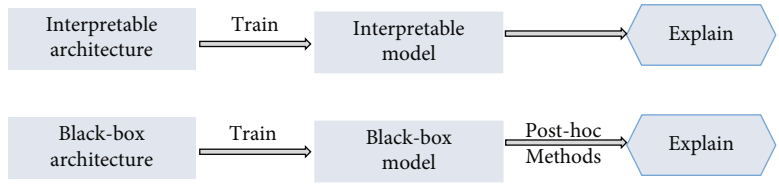

Figure 5: Intrinsic vs. post hoc interpretability.

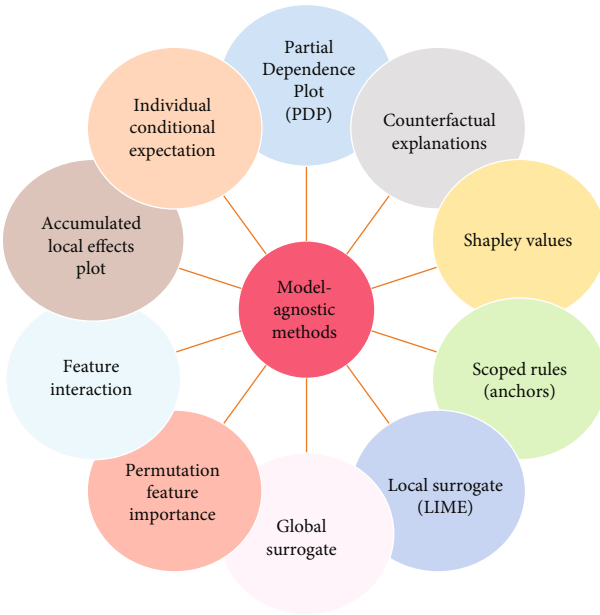

Figure 6: Model-agnostic methods.

stylometric and personality features of the users and trained a CNN for content-based feature extraction. Castro et al. [5] proposed the MUStARD dataset which contains audio-visual data from popular sitcoms and showed how multimodal cues enhance the primary sarcasm classification task. In 2020, Baruah et al. [41] implemented BERT, bi-LSTM, and SVM classifiers for sarcasm detection utilizing the context of conversations. Jena et al. [42] performed the task of sarcasm detection in conversations using a C-Net model which comprised BERT models. Recently, Zhang et al. [43] proposed a model based on quantum theory and fuzzy logic to detect sarcasm in conversations in MUStARD and Reddit datasets.

The use of explainable AI for interpretability of the underlying ML techniques for sarcasm detection has been studied by few researchers. In 2018, researchers Tay et al. [44] improved the interpretability of the algorithms by employing multidimensional intra-attention mechanisms in their proposed attention-based neural model. The proposed model was validated on various benchmark datasets of Twitter and Reddit and compared with other baseline models. Akula et al. [45] focused on detecting sarcasm in texts from online discussion forums of Twitter, dialogues, and Reddit datasets by employing BERT for multihead self-attention and gated recurrent units, to develop an interpretable DL model as self-attention is inherently interpretable.

4. XAI for Sarcasm Detection in Dialogue Dataset

Black-box ML models have observable input-output relationships but lack transparency around inner workings. This is typical of deep-learning and boosted/random forest models which model very complex problems with high nonlinearity and interactions between inputs. It is important to decompose the model into interpretable components and simplify the model's decision making for humans. In this research, we use XAI to provide insights into the decision points and feature importance used to make a prediction about sarcastic disposition of conversations. The architectural flow of the research undertaken in this paper is shown in Figure 7.

The MUStARD dataset used for this research consists of 690 dialogues by the speakers from four famous television shows. It is publicly available and manually annotated for sarcasm. The dataset consists of details about the speaker, utterance, context, context speakers, and sarcasm. For example, the dataset entry for a conversational scene as given in Figure 8 from Friends, season 2, episode 20, is shown in Table 1.

It is noted that most of the dialogues in this dataset are from two most popular shows, namely, the Big Bang Theory and Friends. The data is balanced with an equal number of sarcastic and nonsarcastic dialogues. Figure 9 shows the dataset distribution for the respective TV shows.

The following subsections discuss the details.

4.1. Supervised Machine Learning for Sarcasm Detection in Dialogues. The data was cleaned as the dialogues obtained had some errors in spelling, emoticons, and unnecessary brackets and names of the subtitle providers; any column which had any missing values or wrong data was removed from the dataset. The evaluation of an utterance relies strongly on its context. The contextual interaction between associated chronological dialogues is based on conversational common ground and thereby raising it to prominence in the current context as shown in Figure 10.

Therefore, we use the punch-line utterance, its accompanied context, and the sarcastic/nonsarcastic label to train our model. tf-idf vectorization [46] is done to transform the textual features into representation of numbers. The data is trained using an ensemble learning approach, eXtreme Gradient Boosting (XGBoost). As a popular implementation of gradient tree boosting, XGBoost provides superior classification performance in many ML challenges. In gradient

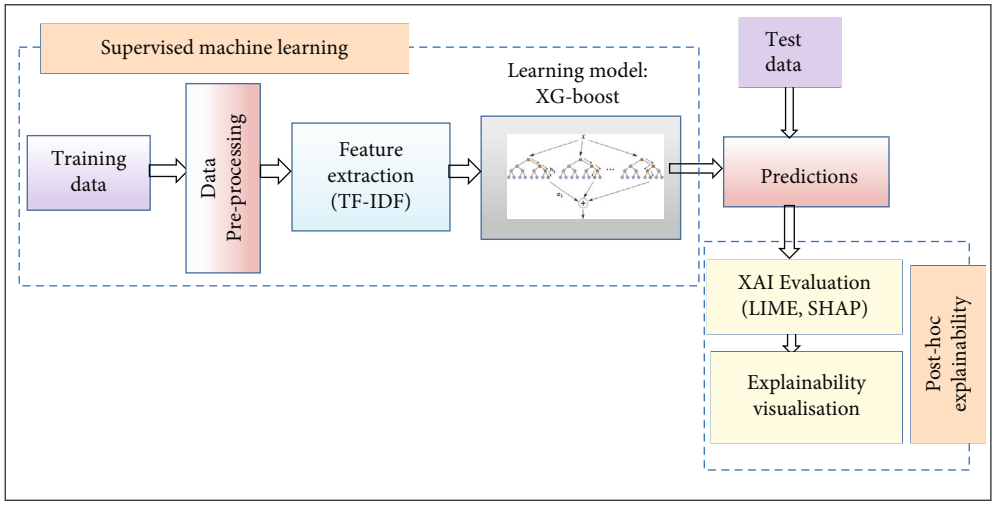

Figure 7: The architectural flow of the research undertaken.

Figure 8: Friends, season 2, episode 20.

boosting, a shallow and weak tree is first trained and then the next tree is trained based on the errors of the first tree. The process continues with a new tree being sequentially added to the ensemble, and the new successive tree improves on the errors of the ensemble of preceding trees. The key advantages of using XGBoost are that it is highly flexible, leverages the power of parallel processing, supports regularization, handles missing values, allows tree pruning, and has built-in cross-validation and high computational speed. On the flip side, explaining the XGBoost predictions seems hard and powerful tools are required for confidently interpreting tree models such as XGBoost. Subsequently, we discuss the two model-agnostic methods selected for seeking explanations that justify and rationalize the black-box model of XGBoost for sarcasm detection in dialogues.

4.2. Post Hoc Explainability Models for Sarcasm Detection in Dialogues. Post hoc interpretability approaches propose to generate explanations for the output of a trained classifier in a step distinct from the prediction step. These approximate the behaviour of a black box by extracting relationships between feature values and the predictions. Two widely accepted categories of post hoc approaches are surrogate models and counterfactual explanations [14]. Surrogate

Table 1: Dataset entry for the given conversational scene from Friends.

Utterance	An utterance is a unit of speech bound by breaths or pauses. The dialogues are spoken by the speaker with respect to the context in the scene.	But younger than some buildings!
Speaker	The character of the series who is giving the dialogue delivery.	Chandler
Context speakers	The side characters to whom the dialogue is being uttered by the main character of that scene.	Chandler
Context	The reason or the scene on the series which led to the dialogue utterance by the speaker.	I know Richard's really nice and everything, it's just that we do not know him really well you know, plus he is old *(Monica glares)* -er than some people.
Show	Name of the show	Friends
Sarcasm	This is the feature in the data to show whether the dialogue utterance by the speaker is sarcastic or nonsarcastic utterance is given as true and nonsarcastic comment is given as false in the dataset.	True

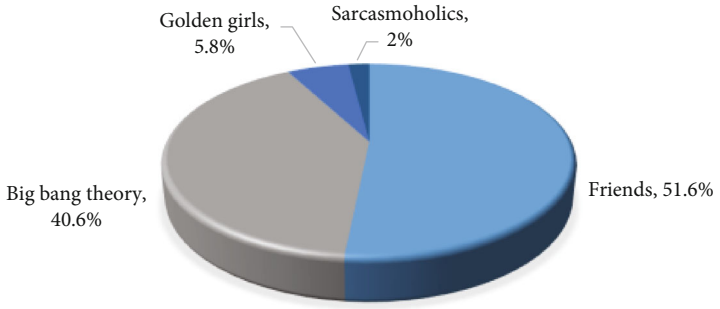

Figure 9: Dataset distribution for TV shows.

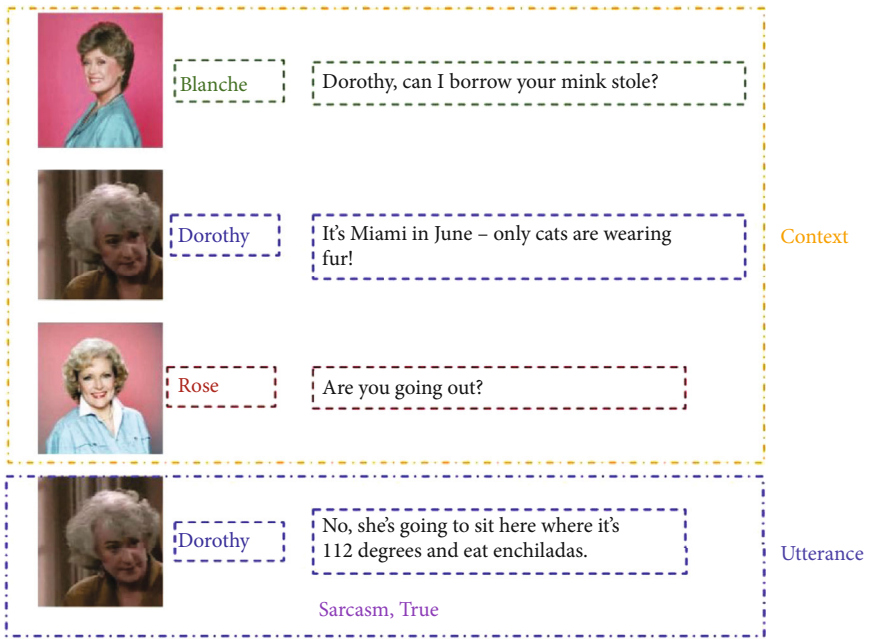

Figure 10: Utterance and associated context from a scene in The Golden Girls.

model approaches are aimed at fitting a surrogate model to imitate the behaviour of the classifier while facilitating the extraction of explanations. Often, the surrogate model is a simpler version of the original classifier. Global surrogates are aimed at replicating the behaviour of the classifier in its entirety. On the other hand, local surrogate models are

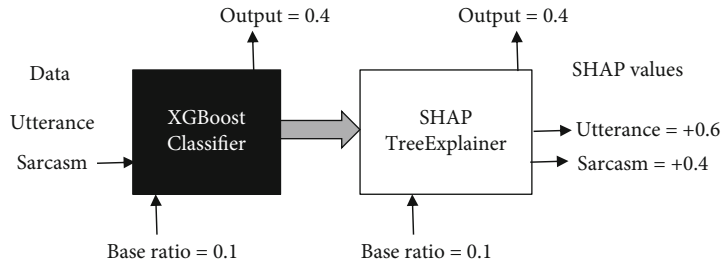

Figure 11: SHAP model architecture.

Table 2: Performance results using utterance + context.

Learning models	Accuracy	Precision	Recall	F-1 score
XGBoost	0. 931	0.965	0.887	0.924
Random forest	0.586	0.402	0.637	0.492
SVM [5]	—	0.579	0.545	0.541

Table 3: Performance results using only utterance.

Learning models	Accuracy	Precision	Recall	F-1 score
XGBoost	0.879	0.852	0.918	0.883
Random forest	0.547	0.369	0.579	0.405
SVM [5]	—	0.609	0.596	0.598

trained to focus on a specific part of the rationale of the trained classifier. In this research, we use two different post hoc local surrogate explainability methods, namely, Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive exPlanations (SHAP). The methods fundamentally differ in terms of the interpretability technique used to explain the working of the black-box model. Typically, LIME creates a new dataset or text from the original text by randomly removing words from the original test and gives the probability to each word to eventually predict based on the calculated probability. SHAP, on the other hand, does not create a separate dataset but uses Shapley values to explain the prediction of any input by computing the contribution of each feature for prediction.

4.2.1. LIME. LIME is available as an open-source Python package. It is a local surrogate approach that specifies the importance of each feature to an individual prediction. LIME does not work on the training data; in fact, it gives the prediction by testing it with variations of the data. It trains a linear model to approximate the local decision boundary for that instance, which then generates a new dataset consisting of all the permutation samples along with their corresponding predictions. New data is created by randomly removing words from the original data. The dataset is represented with binary features for each word. A feature is set to 1 if the corresponding word is included and 0 if it is not included. The new dataset of the LIME then trains the interpretable model, i.e., the RF model which is then weighted by the proximity of the sampled instances to the instance of interest. The learned model should be able to give the general idea of the machine learning model prediction locally, but it may not be a good global approximation. The generic steps of LIME include sampling of instances followed by training the surrogate model using these instances to finally generate the final explanation given to the user through a visual interface provided with the package. Mathematically, LIME explanations are determined using

$$\text{explanation}(x) = \text{argarg min } g \in G\, L(f, g, \pi_x) + \Omega_g. \quad (1)$$

According to the mathematical formula, the explanation model for instance x is the ML model (random forest, in our case) which then minimises the loss L, such as mean square error (MSE). This L measures the closeness of the explanation to the prediction of the original model f, while keeping the model complexity $\Omega(g)$ low. G is the pool of possible explanation, and π_x is the proximity measure of how large the neighborhood is around the instance x. LIME optimizes only the loss part of the data.

The idea for training the LIME model is simple:

(i) Select the instance which the user wants to have explanation of the black-box prediction

(ii) Add a small noisy shift to the dataset and get the black-box prediction of these new points

(iii) Weight the new point samples according to the proximity of the instance x

(iv) Weighted, interpretable models are trained on the dataset with the variations

(v) With the interpretable local model, the prediction is explained

4.2.2. SHAP. SHAP is aimed at explaining individual explanations based on the cooperative game theory Shapley values. Shapley values are used for the prediction to be explained by the assumption of each feature value of the instance as a "player." These values tell the user how fairly the distribution is among the "players" in game. The Shapley value is the average marginal contribution of a feature value across all possible coalitions. The reason to choose SHAP as our second explainable model was because SHAP computes the contribution of each feature of the prediction. These features act as "players" which will then be used to see if the payoff of the distribution is fair or not. It needs to satisfy the local accuracy, missingness, and consistency properties making predictions [17].

Utterance +context

	Predicted 0	Predicted 1
Actual 0	TN = 309	FP = 36
Actual 1	FN = 11	TP = 334

Only utterance

	Predicted 0	Predicted 1
Actual 0	TN = 290	FP = 28
Actual 1	FN = 55	TP = 317

Figure 12: Confusion matrix of XGBoost on MUStARD dataset.

SHAP explains the output of the black-box model by showing the working of the model to explain the prediction of an instance computing each feature's contribution to the prediction. As given in (2), mathematically, SHAP specifies explanation of each prediction as it gives out the local accuracy of the represented features

$$g\left(z'\right) = \varnothing_0 + \sum_{j=1}^{M} \phi_j z'_j, \tag{2}$$

where g is the explanation model and $z' \epsilon \{0,1\}^M$ is the coalition vector in the dataset. M denotes the maximum size of the coalition in SHAP where entry 1 represents that the feature is present and 0 represents that the feature is absent.

SHAP basically follows three properties for the result, and those properties are as follows:

(i) *Local Accuracy*. Local accuracy means that the explanation model should match the original model as given in

$$f(x) = g\left(x'\right) = \phi_0 + \sum_{j=1}^{M} \phi_j x'_j \tag{3}$$

(ii) *Missingness*. Missing feature gets the attribution score of 0 where 0 represents the absence of the feature. It means that the simplified input feature and the original input feature should be the same so that it does not have any impact. It is given as shown in

$$x'_j = 0 \Longrightarrow \phi_j = 0 \tag{4}$$

(iii) *Consistency*. Consistency means that the values increase or remain the same according to the marginal contribution of the feature values of the model. It is given by

$$f'_x\left(z'\right) - f'_x\left(z'_j\right) \geq f_x\left(z'\right) - f_x\left(z'_j\right) \tag{5}$$

In the paper, the features which are used for the target prediction and the SHAP value for the contribution of that feature are the difference between the actual prediction and the mean prediction. SHAP provides both local and global interpretability by calculating SHAP values on the local level for feature importance and then providing a global feature importance by summing the absolute SHAP values for each of the individual predictions. The SHAP model architecture is shown in Figure 11.

We use KernelSHAP (https://docs.seldon.io/projects/alibi/en/stable/methods/KernelSHAP.html) in this work for the estimation of the instance x of each feature contribution. KernelSHAP uses weighted local linear regression to estimate the Shapley values for any model.

5. Results and Discussion

We implemented the model using scikit-learn, a framework in Python. The classification performance of XGBoost was evaluated using accuracy, F1 score, precision, and recall as metrics. The training : test split was 70 : 30. The model is trained with default parameters using the Python XGBoost package. The performance of XGBoost was compared with another ensemble learning method—random forest and superior results were observed using XGBoost. Also, the primary goal of this research was to investigate the role and importance of context we trained and tested the model with and without context. A comparison with the existing work [5] that uses support vector machines (SVM) as the primary baseline for sarcasm classification in speaker-independent textual modality is also done. The results obtained using the punch-line utterance and its associated context are shown in Table 2 whereas the results obtained using only the punch-line utterance that is without using context as a feature are shown in Table 3.

It is evident from the results that sarcastic intent of the thread is more efficiently captured using context, improving the accuracy by nearly 5%. The confusion matrix for the XGBoost classifier with and without context is shown in Figure 12. To compute the confusion matrices, we take a count of four values as follows:

(i) *True Positives* (TP): number of sarcastic utterance correctly identified

(ii) *False Positives* (FP): number of nonsarcastic utterance that was incorrectly identified as sarcastic utterance

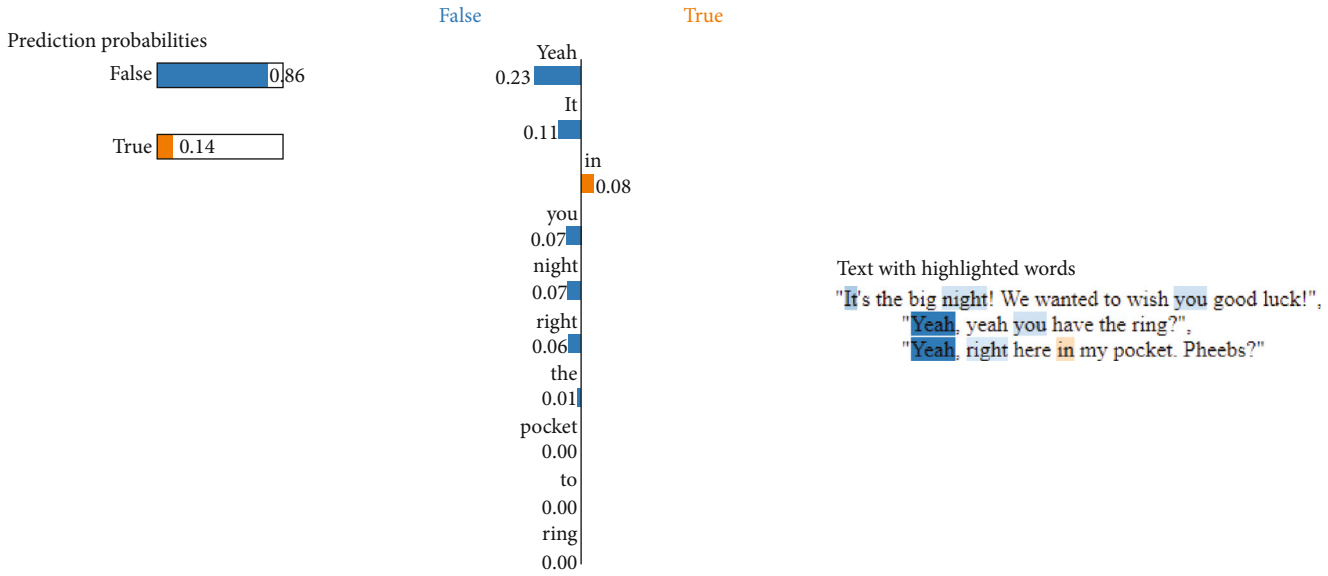

Figure 13: LIME visualisation.

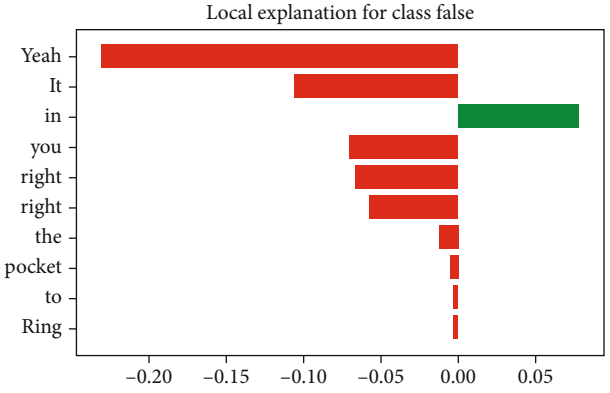

Figure 14: Local explanation for false class.

(iii) *False Negatives* (FN): number of sarcastic utterance that was incorrectly identified as nonsarcastic utterance

(iv) *True Negatives* (TN): number of nonsarcastic utterance correctly identified

The objective was not only to produce higher results but also to produce a better analysis. Therefore, after the evaluation of the learning algorithm, explainable models of LIME and SHAP were used for prediction interpretability. LIME text classifier and LIME text explainer were used to obtain the explanation model for LIME. The class names were set to true and false according to the label, for the LIME text explainer with random state of 42. For SHAP, it was trained and tested on the training and testing vectors generated by tf-idf vectors with 200 background samples to generate the force plot and summary plot of the XGBoost using utterance and context as features.

The explanation model for LIME and SHAP shows which words in the dialogues of the characters influence the model

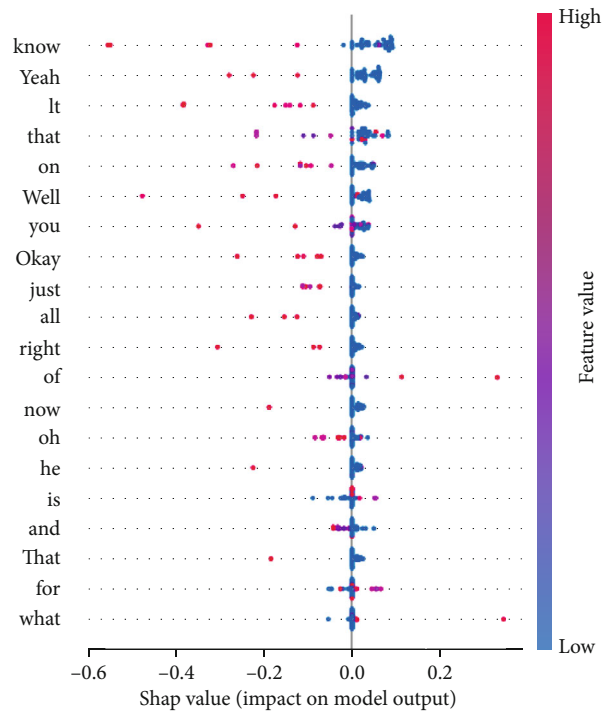

Figure 15: SHAP summary plot.

to label the utterance as sarcastic or not. The explainability scores from each of the methods are generated for every feature in the dataset. Evidently, for an utterance with sarcasm, certain words receive more importance than others. Figure 13 shows the LIME visualisation, where it can be observed that only some parts of the dialogue (taken arbitrarily) are being used to determine the probability of the sarcasm of the utterance by the speaker. As we randomly select an utterance in the test set, it happens to be an utterance that is labelled as nonsarcastic, and our model predicts it as

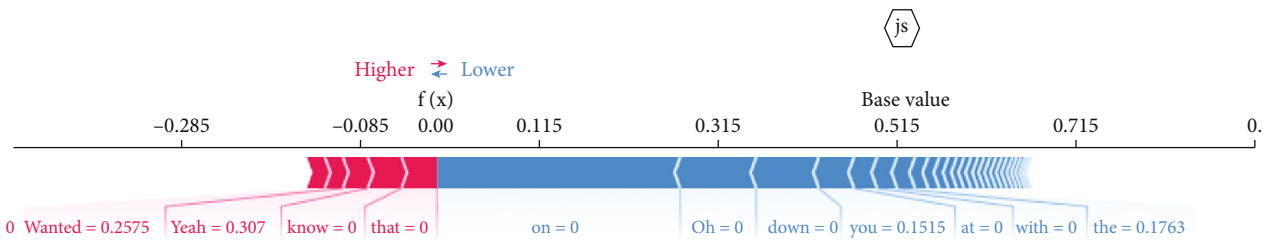

Figure 16: SHAP force plot.

nonsarcastic as well. Using this utterance, we generate explanations.

Noticeably, for this conversation, word "*Yeah*" has the highest negative score for class sarcasm and our model predicts this conversation should be labelled as nonsarcastic with the probability of 86%.

Figure 14 shows how the weights are trained, and the weights of each word given in the utterance are used to determine the sarcasm of the utterance by the speaker.

The same goes for the visualisation of the SHAP model as given in Figure 15, which helps the user understand how the model is making the decision for detecting sarcasm in dialogues. It is using each and every word as a "player" and giving the coalition of whether the model can equally pay off or not. This is a very helpful view that shows at a global level in which direction each feature contributes as compared to the average model prediction. The y-axis in the right side indicates the respective feature value being low vs. high. Each dot represents 1 instance in the data, and the cluster of dots indicates there are many instances in the data with that particular SHAP value.

Thus, the SHAP summary plot combines the feature effect with the importance of the feature. In the SHAP summary plot, each point is the Shapley value for a feature and an instance. The y-axis and x-axis in the summary plot show the feature and the Shapley values, respectively. The colors in the summary plot indicate the impact of the feature from high to low, and the overlapping points in the plot show the distribution of the Shapley values per feature.

Another way to understand the explainability of the utterance using SHAP can be done using the force plot of the data. A force plot helps visualising Shapley values for the features. Feature values in pink cause to increase the prediction. The size of the bar shows the magnitude of the feature's effect. Feature values in blue cause to decrease the prediction. Sum of all feature SHAP values explains why model prediction was different from the baseline. Figure 16 gives the multiprediction force plot used in the given instance with utterance and context for the analysis of the prediction path. Again, the word "*Yeah*" has higher feature importance.

The results support the hypothesis that how each word in the utterance with respect to the context of the dialogues is important for sarcasm detection.

6. Conclusion

With the accelerated use of sentiment technologies in online data streams, companies have integrated it as an enterprise solution for social listening. Sarcasm is one of the key NLP challenges to sentiment analysis accuracy. Context incongruity can be used to detect sarcasm in conversational threads and dialogues where the chronological statements formulate the context of the target utterance. We used an ensemble learning method to detect sarcasm in benchmark sitcom dialogue dataset. Results clearly establish the influence of using context with the punch-line utterance as features to train XGBoost. Further, the predictions given by the black-box XGBoost are explained using LIME and SHAP for local interpretations. These post hoc interpretability methods demonstrate that how few words unambiguously contribute to the decision and word importance is the key to accurate prediction of the sarcastic dialogues. As a future work, we would like to evaluate other XAI methods such as PDP for the detection of sarcasm. Also, temporal context and span analysis for context incongruity are another promising line of work. Gauging other rhetorical literary devices in online data streams is also an open domain of research. Auditory cues such as tone of the speaker and other acoustic markers such as voice pitch, frequency, empathetic stress and pauses, and visual cues for facial expressions that can assist sarcasm detection in audio-visual modalities need further investigation.

Data Availability

Publicly accessible data has been used by the authors.

Additional Points

Code Availability. Can be made available on request.

Ethical Approval

The work conducted is not plagiarized. No one has been harmed in this work.

Conflicts of Interest

The authors certify that there is no conflict of interest in the subject matter discussed in this manuscript.

Authors' Contributions

All the authors have equally contributed in the manuscript preparation. All the authors have given consent to submit the manuscript. The authors provide their consent for the publication.

References

[1] N. Majumder, S. Poria, H. Peng et al., "Sentiment and sarcasm classification with multitask learning," *IEEE Intelligent Systems*, vol. 34, no. 3, pp. 38–43, 2019.

[2] D. Jain, A. Kumar, and G. Garg, "Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN," *Applied Soft Computing*, vol. 91, article 106198, 2020.

[3] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: word embeddings to predict the literal or sarcastic meaning of words," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1003–1012, Lisbon, Portugal, 2015.

[4] A. Kumar and G. Garg, "Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets," *Journal of Ambient Intelligence and Humanized Computing*, 2019.

[5] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, "Towards multimodal sarcasm detection (an _obviously_ perfect paper)," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4619–4629, Florence, Italy, 2019.

[6] A. Kumar and G. Garg, "Sarc-m: sarcasm detection in typographic memes," in *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019*, Uttaranchal University, Dehradun, India, 2019.

[7] A. Kumar and G. Garg, "The multifaceted concept of context in sentiment analysis," in *Cognitive Informatics and Soft Computing*, P. Mallick, V. Balas, A. Bhoi, and G. S. Chae, Eds., vol. 1040 of Advances in Intelligent Systems and Computing, pp. 413–421, Springer, Singapore, 2020.

[8] D. Gunning, *Explainable artificial intelligence (XAI)*, Defense Advanced Research Projects Agency (DARPA), 2017, https://www.cc.gatech.edu/~alanwags/DLAI2016/(Gunning)%20IJCAI-16%20DLAI%20WS.pdf.

[9] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model," *Complexity*, vol. 2021, Article ID 6634811, 11 pages, 2021.

[10] C. Li, X. Zheng, Z. Yang, and L. Kuang, "Predicting short-term electricity demand by combining the advantages of ARMA and XGboost in fog computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5018053, 18 pages, 2018.

[11] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: a survey," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 210–215, Opatija, Croatia, 2018.

[12] T. Miller, "Explanation in artificial intelligence: insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[13] G. Stiglic, P. Kocbek, N. Fijacko, M. Zitnik, K. Verbert, and L. Cilar, "Interpretability of machine learning-based prediction models in healthcare," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 5, article e1379, 2020.

[14] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: a survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": explaining the predictions of any classifier," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 1135–1144, San Diego, California, 2016.

[16] C. Meske and E. Bunde, "Transparency and trust in human-AI-interaction: the role of model-agnostic explanations in computer vision-based decision support," in *International Conference on Human-Computer Interaction, HCII 2020*, H. Degen and L. Reinerman-Jones, Eds., vol. 12217 of Lecture Notes in Computer Science, pp. 54–69, Springer, Cham, 2020.

[17] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017, https://arxiv.org/abs/1705.07874.

[18] A. Messalas, Y. Kanellopoulos, and C. Makris, "Model-agnostic interpretability with shapley values," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–7, Patras, Greece, 2019.

[19] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)," *IEEE access*, vol. 6, pp. 52138–52160, 2018.

[20] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: an overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89, Turin, Italy, 2018.

[21] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2019.

[22] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 704–714, Seattle, Washington, USA, 2013.

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

[24] J. Tepperman, D. Traum, and S. Narayanan, ""Yeah right": sarcasm recognition for spoken dialogue systems," in *Ninth international conference on spoken language processing*, Pittsburgh, Pennsylvania, 2006https://www.isca-speech.org/archive/archive_papers/interspeech_2006/i06_1821.pdf.

[25] R. Kreuz and G. Caucci, "Lexical influences on the perception of sarcasm," in *Proceedings of the Workshop on Computational Approaches to Figurative Language - FigLanguages '07*, pp. 1–4, Rochester, New York, 2007.

[26] O. Tsur, D. Davidov, and A. Rappoport, "ICWSM—a great catchy name: semi-supervised recognition of sarcastic sentences in online product reviews," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 4, no. 1, 2010.

[27] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcasm in Twitter and Amazon," in *Proceedings of the fourteenth conference on computational natural language learning*, pp. 107–116, Uppsala, Sweden, 2010.

[28] A. Reyes, P. Rosso, and T. Veale, "A multidimensional approach for detecting irony in Twitter," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 239–268, 2013.

[29] D. Bamman and N. Smith, "Contextualized sarcasm detection on Twitter," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, Oxford, UK, 2015.

[30] S. Mukherjee and P. K. Bala, "Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering," *Technology in Society*, vol. 48, pp. 19–27, 2017.

[31] A. G. Prasad, S. Sanjana, S. M. Bhat, and B. S. Harish, "Sentiment analysis for sarcasm detection on streaming short text data," in *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, pp. 1–5, London, UK, 2017.

[32] Y. Ren, D. Ji, and H. Ren, "Context-augmented convolutional neural networks for Twitter sarcasm detection," *Neurocomputing*, vol. 308, pp. 1–7, 2018.

[33] A. Kumar and G. Garg, "Sarcasm detection using feature-variant learning models," *Proceedings of ICETIT 2019*, P. Singh, B. Panigrahi, N. Suryadevara, S. Sharma, and A. Singh, Eds., , pp. 683–693, Springer, Champ, 2020.

[34] A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," *IEEE Access*, vol. 7, pp. 23319–23328, 2019.

[35] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in Twitter with hierarchical fusion model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2506–2515, Florence, Italy, 2019.

[36] A. Kumar, V. T. Narapareddy, V. Aditya Srikanth, A. Malapati, and L. B. M. Neti, "Sarcasm detection using multi-head attention based bidirectional LSTM," *IEEE Access*, vol. 8, pp. 6388–6397, 2020.

[37] I. A. Farha and W. Magdy, "Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pp. 21–31, Kyiv, Ukraine (Virtual), 2021.

[38] D. Faraj and M. Abdullah, "Sarcasmdet at sarcasm detection task 2021 in arabic using arabert pretrained model," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pp. 345–350, Kyiv, Ukraine (Virtual), 2021.

[39] D. Ghosh, A. R. Fabbri, and S. Muresan, "The role of conversation context for sarcasm detection in online interactions," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 186–196, Saarbrücken, Germany, 2017.

[40] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, "Cascade: contextual sarcasm detection in online discussion forums," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1837–1848, Santa Fe, New Mexico, USA, 2018.

[41] A. Baruah, K. Das, F. Barbhuiya, and K. Dey, "Context-aware sarcasm detection using BERT," in *Proceedings of the Second Workshop on Figurative Language Processing*, pp. 83–87, Online, 2020.

[42] A. K. Jena, A. Sinha, and R. Agarwal, "C-net: contextual network for sarcasm detection," in *Proceedings of the Second Workshop on Figurative Language Processing*, pp. 61–66, Online, 2020.

[43] Y. Zhang, Y. Liu, Q. Li et al., "CFN: a complex-valued fuzzy network for sarcasm detection in conversations," *IEEE Transactions on Fuzzy Systems*, p. 1, 2021.

[44] Y. Tay, L. A. Tuan, S. C. Hui, and J. Su, "Reasoning with sarcasm by reading in-between," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1010–1020, Melbourne, Australia, July 2018.

[45] R. Akula and I. Garibay, "Explainable detection of sarcasm in social media," in *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 34–39, Washington D.C., 2021.

[46] J. Ramos, "Using tf-idf to determine word relevance in document queries," *Proceedings of the First Instructional Conference on Machine Learning*, vol. 242, no. 1, pp. 29–48, 2003.