IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 January 2023, accepted 14 February 2023, date of publication 22 February 2023, date of current version 27 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3247427

SURVEY

A Comprehensive Survey on Arabic Sarcasm Detection: Approaches, Challenges and Future Trends

ALAA RAHMA[1], SHAHIRA SHAABAN AZAB[1], AND AMMAR MOHAMMED[1,2]
[1]Department of Computer Science, Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, Giza 12613, Egypt
[2]Faculty of Computer Science, Modern Science and Arts University, 6th of October City 12566, Egypt

Corresponding author: Shahira Shaaban Azab (Shahiraazazy@cu.edu.eg)

ABSTRACT On social media platforms, it is essential to express one's thoughts, opinions, and reviews. One of the most widely used linguistic forms to criticize or express a person's ideas with ridicule is sarcasm, where the written text has both intended and unintended meanings. The sarcastic text frequently reverses the polarity of the sentiment. Therefore, detecting sarcasm in the text has a positive impact on the sentiment analysis task and ensures more accurate results. Although Arabic is one of the most frequently used languages for web content sharing, the sarcasm detection of Arabic content is restricted and yet still naive due to several challenges, including the morphological structure of the Arabic language, the variety of dialects, and the lack of adequate data sources. Despite that, researchers started investigating this area by introducing the first Arabic dataset and experiment for irony detection in 2017. Thus, our review focuses on studies published between 2017 and 2022 on Arabic sarcasm detection. We provide a thorough literature review of Artificial Intelligence (AI) techniques and benchmarks used for Arabic sarcasm detection. In addition, the challenges of Arabic sarcasm detection are investigated, along with future directions, focusing on the challenge of publicly available Arabic sarcasm datasets.

INDEX TERMS Artificial intelligence (AI), Arabic sarcasm detection, deep learning (DL), machine learning (ML), natural language processing (NLP), sentiment analysis (SA).

I. INTRODUCTION

Social media's proliferation and prevalence in people's lives resulted in huge transformations in societies, encouraging people to express their opinions and thoughts widely in different ways. Sarcasm is one of the most common and effective figurative devices used to express implicit sentiments on social media. The increasing number of products and services receiving public feedback, including explicit and implicit sentiments, has created the need for sarcasm detection and analysis as a sub-task of Sentiment Analysis (SA). On the other hand, SA mainly investigates opinions, emotions, and perspectives of people [1] in a specific domain such as products, individuals, organizations, events, or various topics and issues [2], [3], [4]. Furthermore, the significance of sarcasm detection in texts is evident in various applications and domains, such as in business to know the actual reviews of some products, which allows companies to improve their services [5], [6], [7]. Another application domain is healthcare, where sarcasm detection helps to discover the diagnosis of mental illness such as depression [8]. At the same time, it is also advantageous to know the public opinion about critical events in a political context [9].

Regarding the definition of sarcasm, various approaches and perspectives concur that it is a figurative linguistic form that uses an utterance with both implicit and explicit meanings [8], [10], [11]. There are several studies published about sarcasm detection in the English language [12], [13]. In contrast, in Arabic, a few research works have been published indicating that sarcasm detection in Arabic is still in its infancy [14], [15], [16], [17], [18]. Hence, researchers started to pay greater attention to this topic by publishing a set of papers consecutively that apply various approaches for sarcasm detection in the Arabic language. To keep track

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Tucci.

of this evolution, we aim to review the state of the art of the earlier studies in Arabic sarcasm detection. To the best of our knowledge, this is the first comprehensive survey that reviews recent studies, collected datasets, used machine learning (ML) and deep Learning (DL) models, preprocessing steps, and feature extraction methods. Moreover, it discusses the limitations and challenges in Arabic sarcasm detection, proposing future directions based on the findings.

The studies reviewed in this survey were subject to the inclusion and exclusion criteria: Firstly, the included primary study must have been recently published from 2017 to 2022, as the first introduced experiment was in 2017. Second, Arabic sarcasm detection has to be the core problem handled in the study. Third, the performance measurement must be precisely defined if it is an experimental study. Concerning the excluded primary studies, we applied the following criteria: First, studies that handle SA systems or challenges only as a main focus, without taking into account the existence of sarcasm, are disqualified. Second, studies on sarcasm detection in languages other than Arabic are neither included. Finally, studies that detect different linguistic figurative devices, such as parody and offense, are excluded, even if sarcasm is among them. Regarding our search terminology, we utilized irony-sarcasm-related keywords. For example, the terms ''Arabic'' and ''sarcasm detection'' or ''irony detection'' were used in multiple combinations. Moreover, we searched for surveys in this area, and the results were exclusively related to Arabic SA and author profiling, providing us with a strong motivation to conduct this survey. The databases used in our search process are, namely: Springer, Science Direct, ACM digital library, ACL Anthology, and IEEE xplore.

In general, Arabic sarcasm detection studies included in this survey are one of three types: Experimental studies, experimental-Arabic sarcasm corpus provider studies, and studies that provide only Arabic sarcasm or irony corpus without any experiment, which are rarely found (resulted in one paper only to be included). Referring to the data extraction process, we extracted from the included papers the main concepts related to sarcasm, irony, and the Arabic language. Furthermore, we extracted the details related to AI techniques applied for Arabic sarcasm detection. Moreover, preprocessing and feature extraction methods were also investigated; for the new corpora, data collection and annotation processes are captured in detail. Challenges presented in some papers, whether they are related to the experiment or related to the sarcasm detection task itself, were taken into consideration. At the end of the search, 37 studies that handled Arabic sarcasm detection were included in our survey. Two of them present an overview of the main collaborative tasks carried out in this area [9], [19].

The main purpose of Table 1 is to show the amount of published research work in English in this area and compare it with its counterpart in Arabic, where few studies handle sarcasm detection. Search results differ from one database to another depending on the available filters to use. We mainly limited our search, as possible, to a predetermined period (from 2017 to 2022), the type of research work (e.g. articles, conference papers, and surveys), and the precise topic of research (i.e., sarcasm detection). The results reflect the total number of studies we found using the keywords mentioned above, whether they were handling sarcasm detection specifically or some related points. The 37 included studies are divided into 27 studies, distributed among the used databases, and ten studies introduced in the shared task of Arabic irony detection [9] in the Forum for Information Retrieval Evaluation (FIRE) conference. The included papers summarize the primary studies in the literature review section. Meanwhile, some excluded papers were used as secondary references, providing background information. For example, Springer provided eight surveys and literature reviews, mostly SA-related. Moreover, we include seven surveys from IEEE xplore.

In summary, this survey provides a brief background of the Arabic language's characteristics, sentiment analysis relationship with sarcasm detection, and the approaches to handling the classification process for sarcastic text in section II. After that, we mention the common preprocessing, feature extraction methods, and AI techniques used for Arabic sarcasm detection in sections III and IV respectively. Then, we list a detailed description of the collected corpora in section V. Moreover, a brief description of the experiments in the literature on Arabic sarcasm detection is stated in section VI. In section VII, we investigate the problems and propose possible future directions. Finally, in section VIII, we conclude this survey.

II. BACKGROUND INFORMATION

Natural language processing (NLP) is a field that helps computers to figure out how people use language [4]; This means that NLP deals with the techniques used to build computer systems that can understand and interact with human languages. In recent years, Arabic Natural Language Processing (ANLP) has attracted researchers to work on it. Consequently, various systems and applications have been developed for several tasks such as text categorization, machine translation, and sentiment analysis [4].

The Arabic language is rich in its morphology but has a few resources with less explored structures and morphology compared to English [5], [20], [21]. There are over 300 million Arabic speakers in 22 countries [1], [8], [22]. In addition, it is the fourth most commonly used language on social media platforms such as Facebook and Twitter [8], [16], [22], [23]. Concerning the most representative characteristics of the Arabic language, the following are the most distinctive [4]: 1) The language consists of 28 characters. 2) It is written from right to left, in contrast with English [4], [8]. 3) Numbers might be singular, dual or plural. 4) Adjectives, nouns, verbs, and singular adverbs have masculine and feminine forms. 5) The verb could be combined with a prefix or a suffix. For example, verbs in the past tense are identified by suffixes, whereas future and present tenses are designated by a prefix. For example, 'ذهبتْ' ''dahabat'' means ''she went'', whereas,

TABLE 1. Sarcasm and irony detection search results.

Search Database	Arabic sarcasm/irony	English sarcasm /irony	Type of publication			Included	Excluded
			Article	Conference-paper	Survey		
Springer	37 /13	604/162	✓	✓	✓	2	814
Science Direct	110 /45	499 /598	✓	—	✓	1	1251
ACM digital library	22/366	308/373	✓	—	✓	1	1068
ACL Anthology	100	1840	✓	✓	—	21	1900
IEEE xplore	7/2	91/42	✓	✓	✓	3	95
Others	0/10	—-	—-	✓	—	10	—-

‘تذهب’ ‘‘tadhabu’’ means ‘‘she goes’’. 6) The Arabic sentence may begin with verbs followed by the subject. 7) subject pronouns are sometimes removed from the sentence.

However, there are additional characteristics that reflect the complexity of Arabic and result in ambiguity while handling Arabic texts for different NLP tasks. For example, 1) The absence of vowels, which are replaced with diacritics in the Arabic language [1], [4]. Furthermore, the modern writers of Arabic do not use diacritics. Thus, understanding Arabic depends to a great extent on the reader’s knowledge. This is related to structural and lexical ambiguity, as diacritics may lead to different meanings for the same word. 2) Another issue is the use of dots, where distinct letters have the same structure but are distinguished from one another by the quantity of dots. 3)The letter’s form differs depending on where it appears in the word, for example, ‘ع’ which could be written as ع at the end, ‘ـعـ’ at the beginning or ‘ـعـ’ in the middle of the word [1], [4]. 5) The adjective has different forms depending on whether it is masculine or feminine. 6) Word synonyms are diverse. For example, darkness has 52 synonyms. 7) There are no uppercase and lowercase letters that are used to indicate the sentence’s beginning or to emphasize the importance of some words in the English language. 8) The word in Arabic may be a mix of prefix, lemma and suffix attached together. For example, the word ‘فسيأكلونها’ which means ‘‘they will eat it’’ and has the prefixes ‘ف س ي’, the lemma ‘أكل’, and the suffix ‘و ن ها’.

There are various dialects of the Arabic language, beginning with classical Arabic (CA), which was spoken by the native Arabs and is associated with Islam and The Holy Quran. This version of Arabic has been evolved into the Modern Standard Arabic (MSA) used nowadays in the majority of formal situations [4], [22]. The MSA participates with the CA on the same structure and syntax of the sentence [4]. Moreover, each Arab country has its own dialect that is spoken informally in daily life [4], [8], [16]. Additionally, some of Arab users of social media use a new version of writing, which is known as Arabizi. Arabizi is only a writing style that is based on writing Arabic words using Latin characters and it has no impact on the spoken language [16].

Before diving into sarcasm detection, it is worth noting its relationship with sentiment analysis. Commonly, sentiment analysis means analyzing people’s opinions, emotions, or attitudes towards a specific entity such as services, products, and organizations [4]. SA is important to reflect public opinion on social media on specific topics [4], [16]. Nowadays, sentiment analysis includes more than reviewing products and services. It has expanded to include dealing with politics and technology [24]. Generally, the sentiment is classified as negative, positive, and sometimes neutral opinions as presented in the dataset on which the classifier model will be trained [1], [17]. This categorization sometimes leads to ambiguity because each sentence could have sub-emotions and could have both positive and negative sentiments. Thus, a lexicon-based approach could be more beneficial, where the sentiment is categorized into degrees of positivity and negativity. For example, multiple user’s comments could share a common sentiment, but with different levels of intensity [25]. Otherwise, emotion extraction is to distinguish among different emotions, for example; happy, depressed, angry, and so on [1]. According to [14], [26], [27], and [28], sarcasm detection is considered a sub-task of the sentiment analysis task. Whereas sentiment analysis classifies opinions into negative and positive, sarcasm deals with the implicit sentiments, which usually turns the positive into negative and vice versa. Consequently, the existence of sarcasm becomes a challenge for sentiment analysis because of the polarity contradiction [6], [14], [17], [26], [29], [30]; this means that detecting sarcasm in a corpus has a positive impact on the sentiment analysis task [31]. Hence, researchers in [32] have experimented with SA-based application on sarcasm detection but they had low performance, which refers to the need to build specified sarcasm detection models [10]. In brief, detecting sarcasm is not an easy task even for humans [27], [29] as we need having enough knowledge about the culture and the context in which it exists [29].

Currently, there is a linguistic debate evoked amongst researchers where they discuss the different definitions of sarcasm and irony terms. It is worth mentioning that this argument is inherited from other languages, such as English [10], [29]. Some researchers use the term ''sarcasm'', while others use the term ''irony'' instead, and some of them use both terms interchangeably [15], [33]. Furthermore, both terms are defined identically, for instance, in [33], which is not entirely accurate. Thus, the Merriam-Webster's dictionary defines irony as: ''incongruity between the actual result of a sequence of events and the normal or expected result''. This definition, along with others, refers to the additional humorous connotation of irony, which is consistent with the definition of irony in [32]. According to [15], it was defined as ''the conflict between using the verbal meaning of a sentence and its intended meaning''. Whereas in [34] it was defined as ''an evaluative expression whose polarity (i.e., positive, negative) is inverted between the literal and the intended evaluation, resulting in an incongruity between the literal evaluation and its context''. A number of definitions with the same meaning were mentioned in [35] and [36]. All these definitions agree that irony has an intended and unintended meanings.

On the other hand, sarcasm tends to be harsher, humiliating, degrading, and more aggressive [5], [10], [15]. As defined by [5], [27], and [30] sarcasm is a form of verbal irony that is intended to express contempt or ridicule. Merriam Webster's dictionary stated the definition of sarcasm as: ''a sharp and often satirical or ironic utterance designed to cut or give pain''. Moreover, some researchers define it as an utterance with both intended and unintended meanings, which is similar to the definition of irony [17], [31]. Consequently, irony and sarcasm are confused with other figurative language devices such as satire, parody, and humour [10], [34], [37]. This debate led to the conclusion that the nature of the collected datasets for irony and sarcasm detection varies from one researcher to another, as the usage of a term may reflect unintended meaning based on the perspective they adopt regarding the definition. Moreover, it is obvious that the majority of the studies published on sarcasm detection in the Arabic language did not address this issue nor define the term they selected to use. Nonetheless, some studies, such as [37] and [10] were more explicit regarding their linguistic perspective, where they used the irony term as an umbrella that includes sarcasm as a subset [17]. In short, researchers should pay more attention to this issue as it may lead to the development of inaccurate classification models.

In the literature, sarcasm detection is typically addressed as a classification problem. This process begins with the text preprocessing to prepare it for feature extraction, which generates the text inputs that will be fed to the selected ML/DL model to determine whether or not a text is sarcastic. In this regard, sarcasm detection employs various approaches, which are listed in [6] as follows:

1) **Standalone Approach:** which is to determine whether a tweet is sarcastic or not by focusing solely on the tweet and ignoring all other factors. This means using the unigram, bigram, and trigram features only to determine the label of the tweet.
2) **Behavioral Approach:** this approach is considered when determining whether or not a tweet is sarcastic based on multiple factors. For instance, the user's age, their most recent tweet, and the time at which the tweet was posted.
3) **Context-based Approach:** some tweets could not be considered sarcastic if the context in which they appear is not explicit. For example, the following sentence: ''that's what I wanted!'' could be a sarcastic sentence or could not be.
4) **Concept Level Approach:** this is the case when additional information is required to comprehend the tweet. For instance, ''Shoaib Akhtar could have bowled a little faster'' is a sarcastic sentence for someone who knows that Shoaib is the fastest bowler in the world.
5) **Hybrid Approach:** is when we combine several of the preceding approaches simultaneously. For instance, researchers in [38] considered two predictor features to detect sarcasm, which are the past behaviour of the user in addition to the tweet itself.

III. PREPROCESSING AND FEATURE EXTRACTION TECHNIQUES

Preprocessing and feature extraction methods have a significant effect on the results of the classification process [4], [6]. The applied preprocessing or feature extraction techniques depend on the language characteristics and the nature of the task for which they are used. In this section, we describe the most prevalent preprocessing steps followed by the feature extraction methods, which are used for Arabic sarcasm detection.

A. PREPROCESSING STEPS

1) **Data Cleaning:** this step helps in data reduction by removing unnecessary and redundant characters to decrease the feature space [15]. Here are some examples of the procedures applied for data cleaning:
   (1) Removal of diacritics, which is a unique linguistic characteristic in the Arabic language [35], [39], [40].
   (2) Punctuation marks are also removed such as in [15], [18], [35], [39], and [41], with exception of ellipsis, which indicate ironic content [35].
   (3) Stop words removal, which means removing words of little or no semantic significance in the preprocessed text [5], [18], [39], [40], [42], [43], [44].
   (4) Usernames and URLs removal [5], [18], [34], [39], [40], [43].
   (5) Duplicates and retweets removal [7], [37].

(6) Pictures removal [34], [37].
(7) Emojis are removed [8], [44] unless they are related to the ironic content. In this case, emojis will not be removed because researchers such as [13], [35] believe that they play an important role in sarcasm detection.
(8) Multiple spaces are reduced to a single one [12], [35].
(9) Hashtags are also removed [37],except in some works, such as [8], [34], where they are irony-related keywords. In this case, hashtags could be used during the data collection process.
(10) Emoticons and repeated characters are removed. whereas a white space is added before and after non-Arabic digits, English digits, or the alphabet, and between numbers and words [12].
(11) Arabizi or Arabic characters written in English, as well as Arabic or English numbers, are also removed [13], [35], [41].
(12) Finally, some researchers eliminate the tweets which has characters less than a specific number such as in [5].

2) **Normalization:** is the process of unifying tokens with multiple forms to a single form. For example, user mentions are replaced with the words 'مستخدم' [12] or 'يوزر' [8]. Whereas emails are replaced with 'بريد' and URLs are replaced with 'رابط' [12]. In addition, emojis are replaced with their Arabic name without duplication, such as in [18], [39], [41]. Occasionally, hashtags were replaced with the word of 'هاشتاق'. Moreover, letters coming in different forms are normalized into one form, such as 'إ', 'أ' and 'آ' which will be transformed into 'ا' [13], [21], [39], [40].
3) **Tokenization:** means to cut the sentence into tokens that could include one or more words, characters, digits, or symbols [5]. This process also includes another two steps which are lemmatization and stemming [5], [7], [8]. Text normalization and lemmatization were applied by using the AraBERT preprocessor [20] by authors of [18].

B. FEATURE EXTRACTION TECHNIQUES

Feature extraction is the process of extracting the essential information and characteristics that represent the data to a large extent. Applying the appropriate feature extraction technique could ease feature selection and reduction of dimensionality in addition to enhancing the performance of the machine learning applied model for the classification process [4], [6], [45]. Regarding the techniques, there are two applied approaches depending on which classification model is used:

1) **Traditional techniques:** which are used with classical machine learning algorithms [24] such as Support Vector Machine (SVM), Linear regression (LR), and Naïve Bayes (NB). These techniques are:

(1) **Bag of Words:** where documents are represented as vectors of equal size to the vocabulary of the dataset encoding the presence or absence of these terms. This technique ignores the grammar and word's order [16], [46], which means that it does not consider the context of the data. The bag of words technique was used along with different features such as TF-IDF and a range of N-gram technique in [15] and [21].
(2) **N-grams:** refers to a contiguous sequence of number of (n) words. This technique could be applied to achieve more efficient classification by capturing the most frequent n-grams instead of using the entire corpora [24]. n-grams technique was used for sarcasm detection in Arabic with different combinations of classical algorithms in [47]. Furthermore, the n-grams technique was used along with other techniques such as TF-IDF in [48].
(3) **TF-IDF:** refers to term frequency and inverse document frequency, as knowing the frequency of a term in a corpus reflects its relative importance within the text [5], [24], [49]. It is an easy technique to compute and represent words' similarity but it is not efficient with the semantic issues which affect the algorithm's overall performance [41]. TF-IDF method was applied for Sarcasm detection in the Arabic language. For example, it was used to represent the syntactic of the tweets in [44]. Moreover, it could be used along with different techniques such as in [40].

2) **Linguistic Features:** they refer to lexical, syntactic, and stylistic features. These features were applied for English and French irony detection [37]. For example, authors of [8] used stylistic features such as exclamation, question, and quotation marks. Moreover, in [37] surface/stylistic features, sentiment features, shifters, and contextual features were used for Arabic irony detection. Regarding the shifters, they detect different linguistic phenomena, such as assertions in which there is a contradiction between reality and what is said. In addition, they discover exaggeration usage, for example, using the intensifiers 'too' and 'very'. Further, shifters detect the reported speech that is supposedly used frequently in ironic content rather than starting with reporting verbs such as 'قال' which means 'said'.
3) **Deep Learning techniques:** this means using neural networks for language representations. Therefore, the neural network is trained to represent the linguistic and semantic information required about the data [24]. There are two techniques used for sarcasm detection in Arabic:

(1) **Static Word Embedding:** it is a low dimensional dense vector in which the raw data is fed to the

network to be trained. After sufficient training, the semantics of the lexicon are learnt and a map is constructed where the semantically similar words are put together [24], [41]. Furthermore, static word embedding captures standalone representations, which are not dependent on their context [50]. In addition, it performs better with larger datasets than small ones [51]. Regarding its application on Arabic sarcasm detection, in [43], a word and sub-word embeddings were applied using the word2vec tool that has two models for representation; a continuous bag of words and a continuous skip-gram of words. Moreover, in [35], word embeddings were extracted using the Arabic FastText tool. Additionally, the deepmoji tool was used in [44] for emotion-feature extraction.

(2) **Contextual Word Embedding:** despite the efficiency of the static word embedding method, especially the embeddings trained on big datasets, it does not take into account the meaning of a word in different contexts. Thus, it performed poorly on domain-specific datasets because a word in some domain could have completely different meaning than the general context [52]. As a result, contextual word embedding is used to represent the word according to the context where it appears [13], [41]. Tools such as ElMo, for instance, handled this issue effectively, but they require larger datasets, which is not the case with domain-specific datasets. It is worth mentioning that using static word embedding is sometimes preferred in non-contextual tasks such as analyzing vector spaces. In addition, static word embedding has much less computational cost than contextual word embedding [50], [52]. For Arabic sarcasm detection, contextual word embedding was used in a number of studies such as [41] and [13].

IV. AI TECHNIQUES FOR ARABIC SARCASM DETECTION

Although research on Arabic sarcasm detection is quite recent, there are a number of diverse algorithms and techniques used for the classification and feature extraction processes related to this task. In this section, we state and outline the most frequently used AI techniques for Arabic sarcasm detection at the moment. Some researchers applied classical machine learning algorithms while others adapted deep learning techniques, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models. Furthermore, many researchers used the ensemble technique, where different algorithms are combined by stacking their performance results using different statistical methods. Moreover, transfer learning, which aims at fine-tuning the pretrained models [36], is a widely used technique that achieved notable results for Arabic sarcasm detection.

A taxonomy of the applied AI techniques on Arabic sarcasm detection is illustrated in Figure 1. Further, a distribution of the used AI techniques over the published studies is depicted in Figure 2. In addition, the following subsections describe the most common applied AI techniques:

1) **The Classical Machine Learning Algorithms:** in general, these algorithms were used for different text binary classification tasks as mentioned in [49] where they used algorithms such as NB, SVM, LR, and logistic regression Cross-validation (LRCV). The classification was done on three multi-domain datasets using different n-gram features. Researchers reported that the LR and LRCV algorithms outperformed the SVM. In addition, increasing the n-grams led to a decrease in the overall performance. Furthermore, SVM, Logistic Regression, Random Forests (RF), and XGBoost were used in [51] for sarcasm detection. Therefore, they reported that classical ML algorithms achieve the best results for classification problems with small datasets. Moreover, a comparison between classical and transfer learning techniques was introduced in [36] where NB, SVM, XGboost, FastText, and bidirectional pretrained transformer models (BERT) were applied to detect ironic content in three variants of the Spanish language. As a result, they found that the SVM and XGBoost algorithms outperformed the other models with macro F1-score values of 0.70 and 0.69 respectively. whereas the BERT model did not improve the performance and was very close to the NB algorithm's results. To improve the results, they used an ensemble technique of SVM and XGBoost which enhanced the F1-score to 0.71.
Concerning Arabic sarcasm detection, the classical ML algorithms were applied in a set of studies such as [8], [15], and [41]. The SVM was one of the most commonly applied algorithms and achieved outperforming results, such as in [43]. Moreover, researchers in [44] used the ensemble classical ML-based technique to improve the performance.

2) **Deep Learning Models:** deep learning models are currently used in different NLP tasks. These models achieved more reasonable results in various text classification tasks compared to traditional machine learning models, as they are able to represent the word embeddings and classify the text at the same time [46]. Deep neural networks such as RNNs with word embeddings are more appropriate to be applied on large datasets but with higher cost and computational complexity [24], [46]. Regarding Arabic sarcasm detection, there are three different DL models used, which are:

   (1) **Convolutional Neural Networks(CNN):** a deep neural network that solves problems related to spatial data (two or three-dimensional data) such

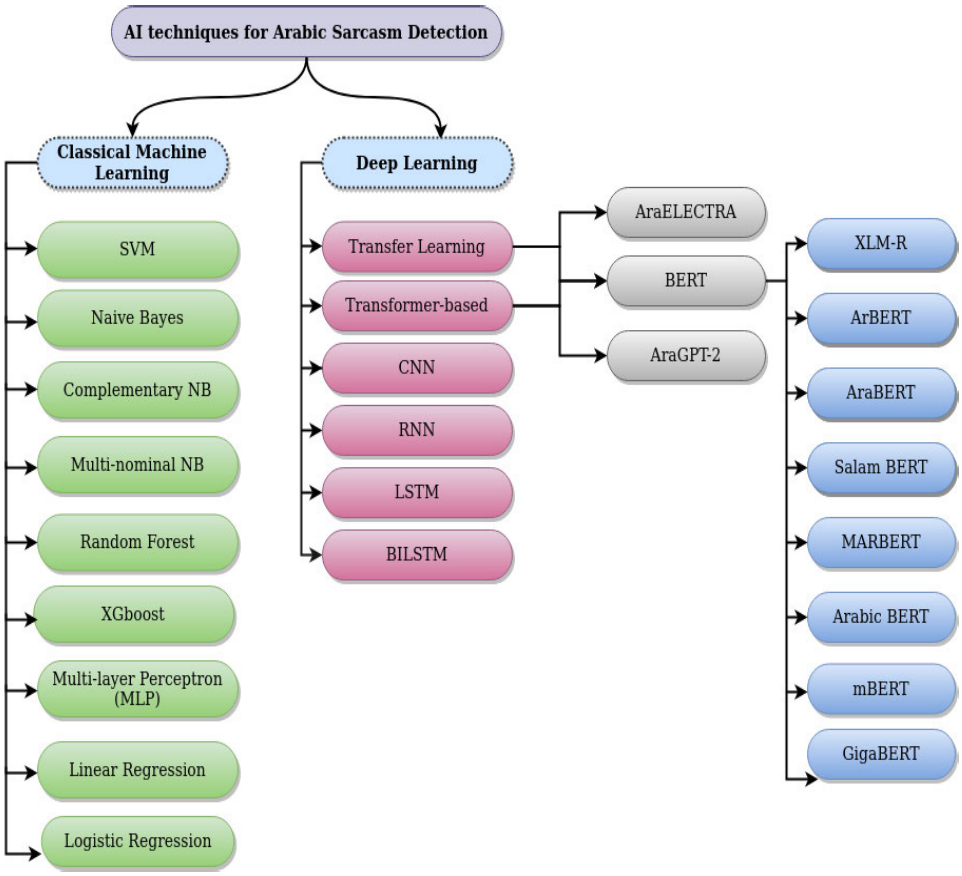

FIGURE 1. Taxonomy of AI techniques for Arabic sarcasm detection.

as images and videos. This means that it could handle texts also, as they are sequential data with one dimension only. Whereas the RNNs' last item in the sequence has a relatively high impact on the outcome, the CNNs do not have this bias which could be an advantage over the RNNs [46]. Regarding Arabic sarcasm detection, the CNN model was applied and showed its superiority in the classification process over different variants of RNNs applied for the same experiment [35]. In addition, a CNN model was used in [18] and stacked with an RNN model to get the advantages of them both. The CNN-RNN model's ensemble technique was used to classify sarcastic content in other languages also [7].

(2) **Recurrent Neural Networks (RNN):** a deep neural network that mainly handles sequential data such as texts [46]. This model was used fundamentally to take the context in which a word appears into consideration [6]. Despite their advantages, the more epochs and layers in an RNN, the higher cost and complexity they have [24]. Moreover, the RNN achieved the best performance for different NLP tasks, for example, in [46] where the RNN model outperformed the CNN model and the classical algorithms. With respect to Arabic sarcasm detection, various RNN variants were applied, such as Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM) [35], and Bidirectional LSTM (BILSTM) [10], [18], [19].

(3) **Transformer-based models:** transformer model is based on multiheaded self-attention layers, ignoring convolution and recursive layers. The attention mechanism is used in transformers in three ways. First, it is used in encoder-decoder attention layers where the previous decoder layer produces queries and the encoder output comes with memory keys and values. Second, each encoder and decoder has six internal layers. Each one of them is composed of two sub-layers; one is multihead self-attention and the second sub-layer is position-wise fully connected feed-forward networks. Furthermore, transformers are less expensive in terms of time. In the following subsections we will explain the most common transformer-based models used for Arabic sarcasm detection.

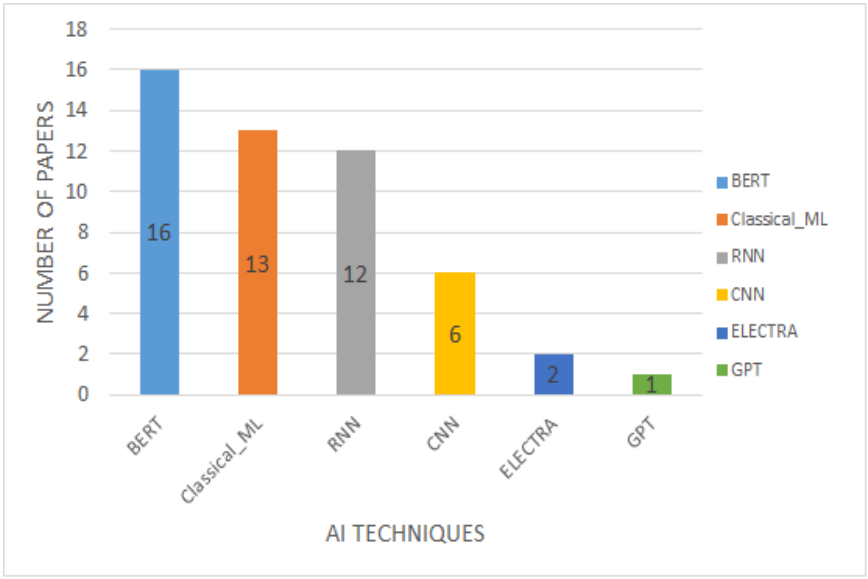

FIGURE 2. Distribution of Applied AI techniques over Arabic sarcasm detection studies.

TABLE 2. Arabic Sarcasm Datasets.

Reference	Social Media Platform	#records	Annotation Method	Labels	Data distribution		Availability
					Sarcastic	Non-sarcastic	
Soukhria [37]	Twitter	5,479	NA	Ironic and non-ironic	1,733	3,746	No
IDAT [9]	Twitter	5,030	Manual	Ironic and non-ironic	2,614	2,416	No
Arabic headlines [62]	Arabic News websites	5,998	Manual	Sarcastic and non-sarcastic	2,999	2,999	Yes
DAICT [34]	Twitter	5,358	Manual and automatic	Ironic and non-ironic	4,809	114	No
Arsarcasm [10]	Twitter	10,547	Manual and automatic	Sarcastic and non-sarcastic. Additional labels were added (i,e. sentiment and dialect labels)	1,682	8,865	Yes
Arsarcasm-v2 [19]	Twitter	15,548	Manual and automatic	Sarcastic and non-sarcastic. Additional labels were added (i,e. sentiment and dialect labels)	2,989	12,559	Yes
AST [8]	Twitter	365	NA	Sarcastic and non-sarcastic	236	106	No
ASAT [5]	Twitter	20,000	Manual and automatic	Sarcastic and non-sarcastic	10,000	10,000	No
SARQA [63]	Twitter	1554	Manual	Sarcastic and non-sarcastic	1165	389	No
DIAM [64]	Twitter	11,240	Manual	Ironic and non-ironic	5,620	5,620	No

(1) **BERT-based Models:** is a language representation model which refers to Bidirectional Encoder Representations from Transformers. The BERT model pretrains the unlabeled data considering the bidirectional representations in both left and right contexts [53]. It is considered the state of the art model for different NLP tasks [12], [40], [53]. A set

of BERT-based models was applied in [26] and compared to different LSTM variants and ensemble technique of CNN and LSTM. The experiments showed that BERT-based models outperform the other used models. In addition, another comparison was introduced in [54] among BERT-based models and classical machine learning algorithms, which results in the outperformance of the BERT models even with the smallest dataset in the experiment. Furthermore, a number of research studies have used BERT-based models to detect sarcasm in the Arabic language. AraBERT [20], for example, was used by the authors of [12], [32]. Further, the experiment showed that AraBERT-v02 achieved the best performance. Moreover, the AraBERT model was used along with an ensemble-based model [18] reporting that the ensemble-based model outperformed the standalone AraBERT model. The MARBERT model [55] was used in [14], [40], [56] and outperformed all the other algorithms that it was compared with, including the multilingual BERT (mBERT) model. Hence, this result reflects the superiority of the monolingual models [20], [55].

(2) **ELECTRA-based models:** ELECTRA is a pretraining tool for text encoders as discriminators rather than generators. This model uses two transformers which are the generator and the discriminator. The ELECTRA model, contrary to the BERT model that depends on masked language; uses a small generator to replace some tokens from the input with alternatives. Further, the discriminator does not predict the original tokens, instead, it predicts whether each token in the corrupted input was replaced with a generator sample or not [57]. AraElectra [58] is the Arabic variant of ELECTRA model. It was pretrained on a large Arabic text using Replaced token detection and was evaluated by different Arabic NLP tasks such as sentiment analysis and named entity recognition(NER). Regarding Arabic sarcasm detection, AraElECTRA was applied within a narrow range of studies such as [12] and [59].

(3) **GPT-based models:** refers to the Generative Pretraining language model that uses a semisupervised approach by applying unsupervised pretraining and supervised fine-tuning approaches. The GPT is a transformer-based model and is pretrained on a large corpus of unlabeled data. Afterwards, the pretrained model is fine-tuned on labelled data with minimum changes in its architecture [60]. AraGPT-2 [61] is the Arabic version of the GPT model with four variants. The model was trained on a large scale Arabic text extracted from the internet texts and news articles. The AraGPT-2 model was mainly used in text generation tasks. Thus, it is not common to be applied in Arabic sarcasm detection because it is, foremost, a classification problem. Hence, AraGPT-2 was applied once in [59] to classify the Arabic sarcastic tweets.

V. ARABIC SARCASM DATASETS

The first collected dataset for irony detection in Arabic was the Soukhria corpus, which is described in [37]. Soukhria is based on political tweets, as it is one of the most common topics on the social networks. It consists of 5,479 tweets distributed as follows: 1,733 ironic tweets and 3,746 non-ironic tweets. The collected tweets are written in MSA, dialectal Arabic, or a mix of both in the majority of the tweets. Mostly, the tweets were written in Egyptian, Syrian, and Saudi dialects. In addition, very rare tweets were written in Tunisian and Algerian dialects. Although the dataset is not public, it is freely available for research purposes. Additionally, a small dataset was collected by the authors of [8] and we give it the title of AST, which stands for Arabic sarcasm detection in Twitter. The dataset was collected using 11 different hashtags from Saudi tweets, which have no images nor videos. After preprocessing, the final dataset consists of 344 tweets, which are distributed as follows: 236 tweets are labelled as sarcastic and 106 are non-sarcastic; while there are 6 tweets on which the annotators did not agree.

Regarding the IDAT dataset, it was introduced in [9] for the shared task named IDAT@FIRE2019. The dataset is composed of 5,030 Arabic tweets. It is based on different political topics related to the Middle East and the Maghreb. The majority of the tweets were written in MSA in addition to Egyptian, Gulf, Levantine, and Maghrebi dialects. The data was manually annotated and distributed as follows: 2,614 are ironic tweets and 2,416 are non-ironic. Besides, a new Arabic corpus was collected for sarcasm detection in [62]. The dataset is based on Arabic news headlines. It was collected manually using the Scrapy python library from two Arabic news websites, namely الحدود and مصرس. This dataset consists of 5998 news headlines and is divided equally between sarcastic and non-sarcastic labels, with 2999 headlines for each of both labels. The dataset was written in MSA and included different domains such as sports, politics, and religion. Furthermore, the DAICT [34] corpus was collected from Arabic tweets also based on ironic hashtags only without any domain-specific keywords to contain different topics. The collected tweets were written during the period from 2012 to 2019. After preprocessing, the resultant data

consists of 5,358 tweets written in MSA, dialectal Arabic, and a mix of both in some tweets. The researchers hired two specialists in linguistics to annotate the data manually. The linguists were from different Arabic regions with different Arabic dialects to understand the contexts of the tweets. Further, the annotators disagreed on some tweets, which led to adding a new label for ''ambiguous'' tweets that are not classified as ironic nor non-ironic.

The ArSarcasm dataset was introduced in [10]. It is based on sentiment analysis datasets that were re-annotated to be suitable for sarcasm detection. These datasets are SemEval's 2017 [65] and ASTD [66]. Regarding the annotation, it was done using the CrowdFlower platform for crowd-Sourcing. The new dataset, Arsarcasm, consists of 10,543 tweets, most of which were taken from SemEval's dataset. Researchers labelled the data as sarcastic and non-sarcastic. In addition, SA labels of the original datasets were added; which are positive, negative, and neutral. Moreover, they added dialectal labels for MSA, Egyptian, Gulf, Levantine and Maghrebi dialects. The Egyptian dialect has the highest percentage of sarcastic tweets with 34% out of the entire sarcastic tweets. Regarding the Arsarcasm-v2 [19], it is an extension of the Arsarcasm dataset that was expanded by adding tweets from the DAICT dataset, which is mostly sarcastic. Furthermore, additional random tweets were collected within the period of November-December 2020 then they were used to balance the DAICT dataset. Regarding the annotation process, the same procedure used to annotate the ArSarcasm dataset was followed to annotate the new portion of the data. Since the DAICT was annotated only for sarcasm, sentiment and dialectal labels were added and the data was manually annotated by Arab annotators. The resultant dataset consists of 15,548 tweets, 2989 out of which are sarcastic.

Regarding the SARQA dataset [63], we give it this title, which stands for Sarcasm Quantification in the Arabic language. This dataset was collected through Twitter API filtered to Arabic only. It has multiple domains such as politics, entertainment, products, sports, and services. In addition, the dataset was written in MSA and dialectal Arabic. Each tweet was annotated by eleven different native Arabic speakers and labelled as sarcastic and non-sarcastic. The final dataset consists of 1554 tweets; 1165 of them are sarcastic. The majority of tweets belong to the politics category, which is one of the most frequently discussed topics in social media.

We assigned the name of ASAT to the dataset introduced in [5], which stands for Automatic Sarcasm Detection in Arabic Tweets. This dataset consists of 20,000 tweets collected within the period from 2010 to 2020; 50% are sarcastic tweets and the remaining are non-sarcastic. It was automatically and manually annotated using hashtags, including political and sports hashtags for non-sarcastic tweets. The dataset includes different variations of Arabic such as MSA, dialectal Arabic, and a mix of both in some tweets. Finally, the DIAM dataset has been collected recently by the authors in [64]. We designated the name of DIAM to the dataset referring to Detecting Irony in Arabic Microblogs. The data was collected using a Tweeter scraper and annotated manually by two Arabic speakers. This dataset consists of 11,240 tweets divided equally between ironic and non-ironic labels.

VI. LITERATURE REVIEW

The main purpose of this section is to review Arabic sarcasm and irony detection studies. The papers are divided into the upcoming four distinct categories based on the classification approaches they adapt. On the other hand, we do not classify the studies on the basis of the preprocessing or feature extraction methods they use.

The published research studies for Arabic sarcasm detection do not exceed thirty-seven studies, as depicted in Figure 3. In addition, a brief summary of the studies is itemized in Table 3, where the studies are ordered based on the year of publication, used dataset and the F1-score value measured for each experiment.

1. **Classical Machine learning approach**: the first collected dataset for irony detection in the Arabic language named Soukhria was introduced in [37]. Moreover, the researchers experimented with RF algorithm on the Soukhria corpus for Arabic irony detection. They employed previously used features with other languages, such as sentiment, contextual, shifters and surface features, which include lexical and stylistic features. Precision, recall, and F1-score were used for performance measurement with values of 0.724, 0.736, and 0.730, respectively. In [8], another model was experimented and trained on the authors' collected AST dataset for Arabic sarcasm detection. This model was trained using the WEKA classification tool and the NB algorithm. Furthermore, the dataset was split into 60-40% for training and test sets, respectively. Regarding the performance, it was measured using precision, recall and F1-score with the values of 65.9%, 71% and 67.6% correspondingly. A different approach was introduced in [41] applying the AraBERT model for contextual word embedding extraction, and the RF algorithm to classify the data after preprocessing. The used datasets are Ar-Sarcasm-V2, the emoji dataset, stop words dataset, and an unseen dataset used to test the results [19]. The dataset was split into 80-20%for training and test sets respectively. They applied data augmentation to solve the imbalanced data issue, using over-sampling and under-sampling techniques to generate sarcastic tweets and remove some non-sarcastic tweets. F1-score, macro F1-score, precision, and recall were used to evaluate the performance with values of 0.5189, 0.6765, 0.6858, and 0.6700, respectively.
An emotion-based voted classifier was applied in [44]. They used a combination of emotion-based features and TF-IDF methods to generate features. The IDAT dataset [9] was split into 4024 tweets for training, of which 2091 were labelled as ironic, and the remaining were non-ironic. Besides, three classical ML algorithms were applied, which are multi-model NB, SVM,

TABLE 3. Summary of research studies on Arabic sarcasm detection.

Reference	Contribution	Year	ML/DL model	Dataset	Statistical results
[37]	1. First Arabic corpus for irony detection 2. Introduce the first irony detection system in Arabic	2017	random forest	Soukhria	Precision:0.724 Recall: 0.736 F1-score:0.730
[8]	Sarcasm classification system using WEKA	2017	Naive Bayes multi-nominal	ART	Precision: 65.9% Recall: 71% F1-score: 67.6%
[62]	1. Collect Arabic sarcastic data from news headlines. 2. Apply hybrid approach of CNN and RNN-based models	2019	CNN LSTM CNN-LSTM LSTM-CNN and CNN-BILSTM	Arabic headlines	Precision: 0.895 Recall: 0.892 F1-score: 0.895
[21]	Build three ensemble-based models for irony detection in Arabic	2019	classical: XGboost, Multi-layer perceptron and RF Ensemble DL: 8 BiLSTM networks hybrid: classical ML+ DL	IDAT	F1-score:86.5%
[67]	1. Introduce a multitask classifier for irony detection in Arabic 2. Train a BERT-based model on dialectal Arabic data	2019	GRU, single task, and BERT multitask BERT	IDAT	F1-score: 82.4%
[15] [68]	Build classical ML and ensemble classifiers for Arabic irony detection	2019	SVM, linear regression with SGD, ensemble: SVM+ Linear regression+ multi-nominal NB+ Random forest	IDAT	F1-score: 82.1%
[35]	A Neural network-based model for Arabic irony detection	2019	pooled GRU lSTM GRU with attention 2D convolution with pooling GRU with capsule LSTM with capsule and Attention	IDAT	F1-score: 0.818
[47]	A neural network-based model for irony detection in Arabic	2019	FastText: 1 n-gram for model1 and 2 n-gram for model2	IDAT	F1-score: 81.7%
[69]	Compare Transformer-based, RNN and classical approaches for irony detection in Arabic	2019	BERT, LSTM, GRU, Multi- layer perceptron, SVM, KNN, Decision trees, RF, Adaboost, quadratic DA and Gaussian NB	IDAT	F1-score: 0.816.
[44]	Build an emotion-based voted classifier for irony detection in Arabic	2019	Ensemble: SVM Multi-model Naive Bayes Logistic Regression	IDAT	Precision:0.77 Recall:0.76 F1-score: 0.75
[43]	Build a classical ML classifier for Arabic irony detection	2019	word2vec+ SVM	IDAT	Precision:0.77 Recall: 0.77 F1-score:0.689
[70]	Build a hybrid model for Arabic irony detection	2019	Classical ML: KNN, RF, Decision Tree, Adaboost, and SVM. DL model: 1. LSTM with 1D-convolution layer. 2. LSTM with TF-IDF features.	IDAT	F1-score:43%

TABLE 3. *(Continued.)* **Summary of research studies on Arabic sarcasm detection.**

Reference	Contribution	Year	ML/DL model	Dataset	Statistical results
[10]	1.Re-annotate existing dataset to detect sarcasm in Arabic 2.Experiment SA system for sarcasm detection 3.Build RNN-based model for Arabic sarcasm detection	2020	BILSTM	ArSarcasm	Precision:62% Recall: 38% F1-score: 46%
[5]	Arabic sarcasm corpus build a classical ML model for Arabic sarcasm detection	2021	Naive Bayes, Logistic Regression and Random Forest	ASAT	Precision: 89.18% Recall: 89.17% F1-score: 89.17%
[12]	Build BERT-based models for sarcasm and sentiment detection	2021	AraBERT and AraElectra	Arsarcasm-v2	Precision:72.64% Recall:71.47% F1-score: 72%
[14]	Build a BERT-based models for sarcasm and sentiment detection	2021	MARBERT	Arsarcasm-v2	Precision: 0.706 Recall: 0.702 F1-score: 0.704
[40]	Build a BERT-based model with data augmentation for Arabic sarcasm and sentiment detection	2021	MARBERT, ARBERT, AraBERT-v02, QARIB, ArabicBERT and mBERT	ArSarcasm-v2	F1-score: 0.647
[13]	Build a multi-task model using static and contextualized embeddings for Arabic sarcasm and sentiment detection	2021	MAR- BERT+ static word embeddings in CNN-LSTM network	ArSarcasm-v2	F1-score: 0.623
[18]	Build an ensemble-based system combining the static and contextualized representations for Arabic sarcasm detection	2021	AraBERT+ CNN-BILSTM	Arsarcasm-v2	Precision: 0.7031 Recall: 0.7447 F1-score: 0.6140 Macro F1-score: 0.7096
[33]	Build a deep learning ensemble model for Arabic sarcasm detection and sentiment analysis	2021	XLM-R and AraBERT	Arsarcasm-v2	F1-score:0.6127 Macro F1-score: 0.7310
[71]	Build a deep learning model for Arabic sarcasm detection and sentiment analysis.	2021	MARBERT, attention layer+ CLS embedding,	Arsarcasm-v2	Precision: 0.7268 Recall: 0.7122 F1-score: 0.6000
[32]	Build a BERT-based model with ensemble technique for Arabic sarcasm detection	2021	AraBERT-v01, AraBERT-v02 and XLM-R	Arsarcasm-v2	Precision: 0.7268 Recall:0.7235 F1-Score: 0.5989 Macro F1-score: 0.7251
[28]	Build an ensemble BERT-based model for sarcasm detection in Arabic	2021	AraBERT -v02+ sentence BERT	ArSarcasm-v2	Precision: 72.68 Recall: 72.35 F1-score: 59.89 Macro F1-score: 72.51
[48]	Build a classical ML-based model for Arabic sarcasm detection	2021	SVM, NB, complementary NB, lR and SGD	ArSarcasm-v2	Precision: 0.7048 Recall: 0.5602 F1-score:0.5936 Macro F1-score: 0.5457

TABLE 3. *(Continued.)* **Summary of research studies on Arabic sarcasm detection.**

Reference	Contribution	Year	ML/DL model	Dataset	Statistical results
[59]	Compare the effectiveness of Arabic transformer-based models for Arabic sarcasm detection and sentiment analysis	2021	Arabic variants of BERT, ELECTRA and GPT	ArSarcasm-v2	F1-score: 0.584
[39]	Build a multi-headed with BERT-based model for irony detection in Arabic	2021	Multi-headed CNN-LSTM-GRU and MARBERT	ArSarcasm-v2	Precision: 0.7231 Recall: 0.7004 F1-score: 0.5662 Macro F1-score: 0.7095
[72]	Build two fine tuned systems to study the impact of offensive language on sarcasm and sentiment detection in Arabic.	2021	AraBERT and SalamBERT	Arsarcasm-v2 and 9 offensive language datasets	Precision: 0.7128 Recall: 0.6807 F1-score: 0.5348 Macro F1-score: 0.6922
[41]	Build a classical ML model using contextual word embedding for sarcasm detection in Arabic	2021	RF, AraBERT for word embedding	Arsarcasm-v2	Precision: 0.6858 Recall: 0.6700 F1-score: 0.5189 Macro F1-score: 0.6765
[56]	Build a BERT-based model for Arabic sarcasm detection and sentiment analysis	2021	mBERT, AraBert, ARBERT and MARBERT	Arsarcasm-v2	Precision: 89.7% Recall: 42.5% F1-score: 48.6%
[22]	Build a classical ML and deep learning-based models for Arabic sarcasm detection using word and character level features	2021	Classical ML: SVM, Multinominal NB, decision trees, RF, SGD and Logistic Regression. Deep learning:LSTM+Aravec	ArSarcasm-v2	F1-score: 41.09%.
[73]	Study the impact of applying preprocessing steps on the classification of Arabic sarcasm detection	2021	SVM BILSTM	Arsarcasm-v2	F1-score: 33.71%.
[63]	1. Introduce a new corpus for Arabic sarcasm detection 2. Approach sarcasm detection as regression problem	2021	multi- dialect Arabic BERT	SARQA	Final loss: 0.011631458
[64]	1. Collect Arabic irony detection corpus. 2.Build a deep neural network for Arabic irony detection	2022	CNN BILSTM	DIAM	Precision: 90% Recall: 84% F1-score: 87%

and logistic regression. Furthermore, cross-validation was conducted on 0.8 % of the data selected at random. Precision, recall and F1-score were used to measure the performance computing the majority voting of the classifiers. The results for the cross validation are 0.77%, 0.76% and 0.75% for precision, recall and F1-score, respectively. In contrast, the F1-score for the test data is 0.807%.

A classical machine learning algorithm using the SVM algorithm was applied in [48]. In addition to the TF-IDF method used to calculate the weight of terms, the unigram and bigram methods were utilized for feature extraction. Five-fold cross-validation was performed to train different algorithms including SVM, LR, NB, complementary Naïve Bayes (CNB), and stochastic gradient descent (SGD). The results

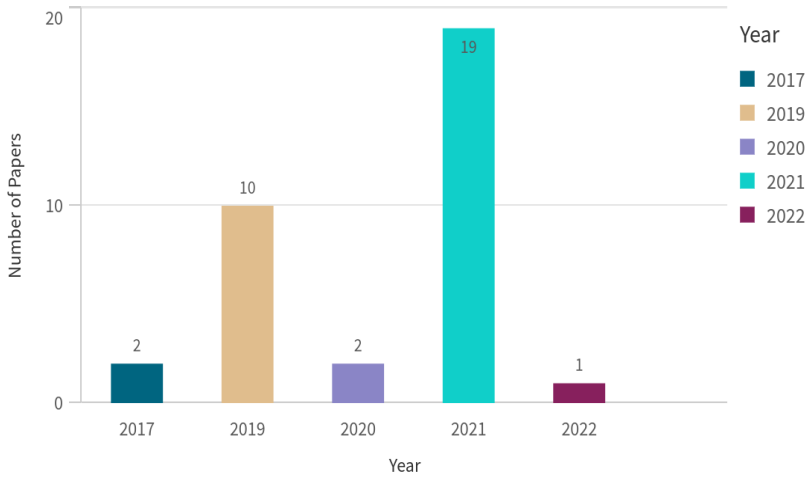

FIGURE 3. Number of published studies on Arabic sarcasm detection.

demonstrate that the SVM outperforms the other algorithms with values of 0.5457, 0.7048, and 0.5602 for macro F1-score, precision, and recall, respectively.

2. **Deep learning approach**: in [10] the first publicly accessible sarcastic dataset "Arsarcasm" was introduced. The dataset was created based on previously published sentiment analysis datasets which are SemEval's 2017 [65] and ASTD [66]. In addition, the study analyzes the annotator's subjectivity in sentiment annotation and how SA systems perform on sarcastic content. Regarding the dataset, it was split into 1682 sarcastic tweets and 8865 non-sarcastic tweets. Furthermore, two experiments were implemented. The first experiment was dedicated for sarcasm detection to study the impact of sarcasm on sentiment analysis and to affirm the need for building sarcasm-specific classifiers. They used the Mazajak sentiment analyzer that applies a recurrent neural network (RNN) as a feature extractor followed by a Long Short Term Memory (LSTM) model [74]. The performance was evaluated using the F1-score metric based on original and new labels. It achieved the values of 0.43 for new labels and 0.44 for original labels of sarcastic tweets. F1-scores for new and original labels of non-sarcastic tweets were 0.64 and 0.61, respectively. The second experiment was to build a baseline system to detect Arabic sarcasm using the BILSTM model. The dataset was split into 80-20% training and test sets. This experiment was measured using precision, recall, and F1-score with values of 62%, 38%, and 0.46, respectively.
   In [47], another neural network was applied using the fastText library, adjusting the parameters of the algorithms to build two different models. Further, the first model has 40 epochs, learning rate value of 0.2, and 1-gram features. While the second model has 50 epochs, learning rate value of 0.1, and 2-gram features. The performance was measured using the F1-score metric with values of 81.7% and 79.4% for the first model and the second model, respectively. Moreover, researchers in [56] applied a pretrained contextualized representation model and fine-tuned it on the ArSarcasm-v2 dataset [19]. Initially, the mBert model was employed. Furthermore, different models were applied, such as the MARBERT, ARBERT and AraBERT models. Due to its pretraining on dialectal data, the MARBERT model outperformed the other models. Precision, recall, and F1-score were used for measurement with values of 89.7%, 42.5% and 57.7%, respectively. Another experiment was conducted in [72] to demonstrate the impact of offensive language on sarcasm detection. Different nine datasets for offensive languages were used with no filtering or preprocessing. In addition, the main dataset of ArSarcasm-v2 [19] was used. This dataset was split into 80-20% for training and test sets, respectively. They applied the AraBERT model to the offensive language datasets and the ArSarcasm-v2 dataset separately. Furthermore, an additional experiment was implemented using the salamBERT model which was trained on MADAR corpus and achieved the best results for F1-score, precision, recall, and macro F1-score with the values of 0.5348, 0.7128, 0.6807, 0.6922, respectively. Besides, four classifiers were built in [14] using the MARBERT transformer-based model which was trained on four dialectal sets of ArSarcasm-v2 [19], where the Levantine and Maghrebi dialects are grouped together in one dataset. The performance was measured using precision, recall and F1-score with the values of 0.706, 0.702 and 0.704, respectively.
   A new approach was introduced in [63] where sarcasm detection is handled as a regression problem rather than classification. In addition, researchers provided a new corpus collected from Arabic tweets, SARQA dataset, which is described in detail in section V.

Multi-dialectal Arabic BERT was used for contextual word embedding extraction, and then the Arabic BERT model was fine-tuned on SARQA dataset. The experiment was measured using a loss function with a final value of 0.011631458. A CNN and BILSTM-based model were introduced in [64]. Moreover, the authors collected a new corpus for Arabic irony detection, namely the DIAM. For word embedding representations, the Aravec and word2vec tools were used. The CNN with two layers outperformed the BILSTM with 90%, 84%, 87% values of precision, recall, and F1-score respectively.

3. **Hybrid Approaches**: a brief overview of The WANLP-2021 Shared Task2 was stated in [19]. The gaol of this shared task was to improve the performance of Arabic sarcasm detection and sentiment analysis. However, we take only the sarcasm detection experiments and results into consideration. There were 27 submissions to the competition for sarcasm detection, with the highest F1-score value of 0.6225. On the other hand, a hybrid classical ML and DL-based approach with word embedding was proposed in [43] for irony detection in Arabic tweets. They applied the SVM algorithm as a baseline for the classification process on the IDAT dataset [9]. Furthermore, two additional models were implemented using the CNN with word embedding for the first model and with sub-word embedding for the second model, which were extracted using GloVe and word2vec tools. The performance of each model was measured using precision, recall and F1-score which are 0.77, 0.77 and 0.689 for SVM; 0.77, 0.74, 0.687 for the first model and 0.81, 0.81, 0.695 for the second model, respectively. It is worth mentioning that IDAT@FIRE2019 was the first shared task for irony detection in the Arabic language, which is briefly described in [9]. The aim of this shared task is to determine whether or not a tweet is ironic using binary classification. The competition had 10 submissions, and the greatest F1-score value of 0.844 indicated that traditional feature-based models outperformed neural ones.

   A deep learning-based model was experimented in [35] using six different neural networks comparing among them to select the network with the highest F1-score. These neural networks are pooled GRU, LSTM, GRU with Attention, 2D Convolution with Pooling, GRU with Capsule, and LSTM with Capsule and Attention. Regarding feature extraction, they applied word embedding using the FastText tool. Furthermore, a 10-fold cross-validation was performed in addition to reducing the learning rate by 0.6 when the model was not improving. the best result was in favour of the 2D convolution with pooling model, with the F1-score value of 0.818. To alleviate the issue of the need for a large dataset used with supervised learning, multi-tasking Transformer-based models were built in [67]. The involved tasks are deception detection, emotion and sentiment analysis, sarcasm detection, and author profiling, including age and gender detection; using different datasets for each NLP task. The IDAT dataset [9] was split into 90-10% for training and test sets respectively. GRU was the baseline network for irony detection. While a BERT-based multilingual model was trained separately for each one of the aforementioned six tasks. A second in-domain BERT model was trained on dialectal Arabic to enhance the performance. The multitask model with specific domain BERT ranked fourth in the competition with an F1-score value of 0.8434.

   A pretrained fine-tuned model was used in [40] with seven variants of BERT-based models. Data augmentation was applied to solve the problem of imbalanced data. Further, the experiment was divided into three phases. First, the authors applied classical machine learning algorithms such as SVM, XGBoost, and RF; along with TF-IDF method for feature extraction. Second, they applied deep neural networks along with character and word-level features. Finally, the BERT-based models were applied, showing that the MARBERT model with the data augmentation process gave the best performance and improved the results by 15%. The BERT-based models are MARBERT and ARBERT [55], QARiB[1], AraBERTv02 [20], GigaBERTv3, [75], Arabic BERT [76], and mBERT [53]. Regarding the results, the MARBERT model outperformed the other models with F1-score value of 0.647 followed by the QARIB model with F1-score value of 0.597. A hybrid approach was adapted by authors in [70] where they built a multilingual model to detect irony in three different languages using three different approaches which are: classical ML including SVM, Decision Tree, RF, Adaboost, Linear SVM, Sigmoid SVM, KNN and SVM RBF. Second, LSTM was used with two types of features: first, a 1D convolutional layer was used to generate sub-word embeddings that are fed then into an LSTM layer. Second, TF-IDF along with the unigram and bigram methods were calculated and then fed the resultant vectors into the neural network. They used IDAT dataset [9], which was split into 2091 ironic and 1933 non-ironic. The performance was measured by using F1-score with the greatest value of 43% in favour of the KNN, NB and SVM algorithms. While the other ML algorithms received an F1-score within the range 35% to 43%. Regarding LSTM, it performed better on the non-ironic Arabic tweets, where it achieved an F1-score of 73% and 69% for non-ironic and ironic tweets, respectively. The weighted average of the F1-score for both labels was 71%. For the sub-word approach, it performed better also with the non-ironic tweets with an F1-score value of 79%.

[1] https://github.com/qcri/QARiB

In [59], authors compared 24 Transformer-based models in terms of sarcasm detection and sentiment analysis in the Arabic language. The experiment showed that the pretrained models on dialectal Arabic are more efficient regarding these tasks. The MARBERT model achieved the best results with an F1-score value of 0.584. Moreover, the Arabic-specific models achieved better performance than their multilingual counterparts. Concerning the computational cost, ElECTRA-based models were the most efficient. Furthermore, the preprocessing was an important factor for the effectiveness of the classifiers. As an illustration, the AraBERT model performed better than the Arabic BERT although they have the same architecture and are pretrained on the same dataset. Another hybrid approach was applied in [73], where the authors focused on the impact of preprocessing on the model's performance. They applied traditional steps such as removal of punctuation, diacritics, emojis, repeated letters, etc. Two models were experimented on the Arsarcasm-v2 dataset [19]. Firstly, they applied the linear SVM algorithm along with the TF-IDF method. In addition, the BILSTM network was used with an embedding layer. The BILSTM and the Linear SVM achieved an F1-score values of 86.05% and 98.83%, respectively.

4. **Ensemble-based Approach**: in [62] a new data source, scraped from Arabic news headlines, was introduced. The data was split into 75-25% for training and test sets, respectively. Furthermore, a 20% of the training data was split for the validation. The authors applied a CNN, RNN and a combination of both. Firstly, the word embedding layer was added to the CNN and RNN models. Moreover, the CNN model used three 1D convolution layers with two filters. Whereas the RNN model applied two LSTM layers and a BILSTM layer for two different models. To represent the word embedding, they mainly used the FastText tool with the CNN model. In addition, the AraVEC tool was used and resulted in decreasing the performance of the classification because of missing some word representations. Three models were applied to represent the hybrid approach, which are a CNN on the top of the LSTM architecture, an LSTM on the top of the CNN architecture; and a BILSTM layer on the top of the CNN architecture. Regarding the results, the CNN-LSTM and the CNN-BILSTM models achieved the best. They were measured using the F1-score, precision and recall with the values of 0.883, 0.891, and 0.885 for CNN-LSTM; and 0.895, 0.895 and 0.892 for the CNN-BILSTM model, respectively.

   Furthermore, an ensemble approach was introduced in [15] while the authors reported its technical details in [68]. For feature extraction, the TF-IDF method was applied along with the unigram and bigram methods. The SVM, LR, and an ensemble classifiers were applied. The ensemble technique included RF, multinominal Bayes, SVM, and linear classifier with Stochastic Gradient Descent (SGD) optimizer. For performance measurement, The F1-score metric was used and resulted in 82.1%, 81.6% and 81.1% for the ensemble classifier, SVM and linear algorithms respectively. Moreover, a transformer-based model was applied in [12] to the ArSarcasm-V2 dataset [19]. Further, the dataset was split into 90-10% for training and test sets respectively. Regarding the experiment, they applied eight variants of two transformer-based models, which are AraElECTRA [58] and AraBERT [20]. Finally, the models were stacked to get the best performance, which was measured using precision, recall and macro F1-score score giving the following results: 72.64, 71.47 and 72.00, respectively. A study introduced in [33] by the team that ranked third in the sarcasm detection task for the WANLP-2021 shared task [19]. They used CLS embedding and BERT-based models for classification, where the pre-trained XLM-R [77] and AraBERT [20] models were fine-tuned on the training data. Finally, an ensemble technique was applied using LR algorithm and adjusting the threshold value to 0.41. The official F1-score and macro F1-score were 0.6127 and 0.7310 respectively. Another Ensemble-based model was experimented in [28] applyingg the AraBERT-v02 and Sentence-BERT models. Both models were fine-tuned on the ArSarcasm-v2 dataset [19], with 80-20% split for training and test sets correspondingly. The Experiment's metrics have the values of 59.89, 72.51, 72.68 and 72.35 for F1-score, macro F1-score, precision and recall, respectively. Moreover, the AraBERT model was applied in [32] and achieved the best accuracy for sarcasm detection task in the WANLP-2021 shared task. The AraBERT-v02 performed better than v01 with an F1-score value of 0.5650. To enhance the results, an ensemble technique was applied with hard-voting. In the testing phase, the model achieved F1-score value of 0.5989. Moreover, the authors used other deep learning models such as XLM-R [77], which performed less than the other models, and mBERT model, which was one of the best algorithms. The authors in [18]implemented an ensemble-based model of CNN-BiLSTM in addition to the AraBERT model to combine the advantages of using static and contextualized word embeddings. The hybrid approach had better performance than the standalone AraBERT model. The performance was measured using precision, recall, F1-score, and macro F1-score with values of 0.7031, 0.7447, 0.6140, and 0.7096 respectively.

   A multitask learning approach was applied in [13]. The model used a combination of static word-level and character-level embeddings as a single task model with the CNN-LSTM network. Then, this model will be concatenated with the MARBERT multi-task model to enhance the performance. F1-score was used to

measure the performance with value of 0.623. This experiment ranked first in the WANLP-2021 shared task [19]. Furthermore, a multi-headed LSTM-CNN-GRU model was applied along with the MARBERT model in [39], as the baseline models. The word embeddings were fed to the deep learning model which consists of BILSTM-CNN, Bidirectional GRU-CNN and CNN-LSTM models while the MARBERT was fine-tuned separately on ArSarcasm-v2 dataset [19]. Additional experiments were conducted and achieved lower results than the baseline model. MARBERT achieved the best performance which was measured using precision, recall, F1-score, and macro F1-score with the values of 0.7231, 0.7004, 0.5662, and 0.7095, respectively. Additionally, an ensemble-based model was experimented in [21]. A set of features was extracted such as topic modeling features, sentiment features, n-gram, bag of words and TF-IDF features. Furthermore, three ensemble classifiers were applied. The first consists of classical ML algorithms, which are XGboost, RF and Multi-layer perceptron (MLP). In addition, eight variants of BILSTM network were experimented. Finally a hybrid approach combining both classical and deep neural models was implemented. The classical ensemble technique achieved the best performance with F1-score value of 84.1%. Moreover, a classical ensemble ML model and an RNN model were applied in [22]. The dataset was a combination of AraSarcasm [19] and DAICT [34] corpora. The ensemble classifier consists of Multinominal NB, Linear SVM, and Ridge classifiers. It used the union feature in SKlearn to combine word-level and character-level features. In addition, they applied LSTM model with the Aravec tool that represents the word embeddings. The performance was measured using F1-score with the value of 41.09 in favor of the ensemble classifier.

Table 3 summarizes the main contributions and the performance of the research studies conducted on Arabic sarcasm detection. Concerning measurement metrics, we consider precision, recall, F-score, macro-f, and loss function.

VII. CHALLENGES AND FUTURE DIRECTIONS

Sarcasm detection is a challenging task in its nature [10], [37], [78], especially for the Arabic language [18], for a variety of reasons that will be explained in this section. The following subsections enumerate the most common challenges in Arabic sarcasm detection. In addition, we recommend some future directions that could be considered to help solving these challenges. More precisely, the challenges can be categorized into Arabic-specific and general challenges, which are related to sarcasm detection as a sub-task of sentiment analysis.

A. ARABIC-SPECIFIC CHALLENGES

1. **Lack of irony/sarcasm corpora**: it is a natural result of the immaturity of this topic of research in the Arabic language. For the moment, a limited number of datasets were collected with small sizes [32], [41], [44], [79]. As a future contribution, we recommend collecting datasets using domain-specific keywords from multiple domains. This method will not only increase the number of the available Arabic sarcasm corpora, but also increase the extracted features that captures contextual differences of the same word in different domains.
2. **Lack of freely dedicated tools**: to process Arabic social media content [37], [41] or any Arabic NLP task in general [4], [23].
3. **Absence of diacritics**: this results in ambiguity for some similar words, which affects the text analysis in general and ironic/sarcastic texts in particular [37], [44]. This informal Arabic writing has high syntactic and semantic ambiguity [72]. Therefore, handling this issue requires sufficient general knowledge of Arabic in addition to the dialects in which the text is written. In this case, automatic data annotation may lead to incorrect data labels. Thus, we recommend hiring linguistic specialists for the annotation process to guarantee high accuracy. This recommended approach was applied once in [34]
4. **Diversity of dialects in Arabic**: other than the modern standard Arabic, there are more dialects such as Egyptian, Levantine, Maghrebi…etc. Even if some dialects share a common vocabulary, their meanings vary by country. This is because each dialect has its own peculiarities, which affect the task's performance [15], [37], [41], [44]. On the other hand, in some cases, sentences may contain contradictory sentiments at the beginning and at the end. For example, in [56]: شعور الإهانة لما أقول لسواق التاكس قصر حلو مشوحلي بإيده وسايق كأن خدت لطشة على أوي وشي كده فعلا أبدعتي يقوم which means: How nice feeling humiliated when I asked the taxi driver if he was going to "Al-Qasr" and he waved me off, driving away, It was like I had taken a slap in the face. You really got the point!

B. GENERAL SARCASM RELATED ISSUES

1. **Contextual and cultural dependency**: sarcasm is highly dependent on culture, gender and context [15], [17], [29], [40], [56]. Thus, it is more likely that ironic content will be correctly annotated if the annotator has a common cultural background with the authors of the tweets.
2. **SA Polarity contradiction**: is the result of using positive emotions in irony or sarcasm to express negative ones. This contradiction produces incorrect polarity of sentiments which lead to poor performance of SA systems [6], [15], [29], [33], [47], [63], [79]. In this regard, presence of sarcasm should be considered in all SA tasks in order to obtain the correct polarity of sentiments.

3. **Absence of conversational context**: occasionally, the ironic content seems to be cut off from its context or that it is part of a sequence of interactions [34]. This is due to focusing on each tweet separately without considering the situation and the context where it appears.
4. **Implicit irony**: this conforms to the definition of irony with no humorous indicators. Consequently, it is more difficult to be captured than the explicit one [34].
5. **Imbalanced datasets**: most of the currently available datasets have a noticeable bias in their data in favour of one of the labels; ironic or non-ironic, which also could lead to inaccurate models classifying Arabic Sarcasm or irony [32], [40], [41]. In the future, we recommend merging the currently existing datasets, which almost all are extracted from twitter to fill this gap in some labels.
6. **The challenging nature of writing on social platforms**: since the writing style tends to be informal, unstructured, and sometimes limited to a specific number of characters, such as in Twitter [7], [18], [27], [79]. For further work, we recommend investigating new social media platforms such as Facebook, which has different characteristics and style of writing. This recommendation would go further with building a generalized model that could classify Arabic sarcasm or irony whether in Facebook, twitter or any other social media platform.

VIII. CONCLUSION

The emergence of social media platforms has promoted opinion and thought to share. The accelerated spread of these social networks shows the need to build systems that handle this data type. Sarcasm is one of the most frequently used figurative devices by social media users to express with ridicule their ideas. Therefore, sarcasm detection interested researchers in languages such as English. However, Arabic sarcasm detection is still in its early stages. Thus, this survey reviews the state-of-the-art research studies conducted on sarcasm detection in Arabic. It is the first survey paper on this topic. We provide a detailed description of the experiments, collected Arabic sarcasm corpora, preprocessing and feature extraction methods, the applied AI techniques, performance metrics, and the challenges associated with Arabic sarcasm detection. In summary, research on sarcasm detection in Arabic is recent, as the first introduced corpus and experiment were in 2017. Further, it was followed by a period of stagnation that lasted until the IDAT@FIRE2019 shared task was held in 2019. Most published studies were in 2021, which depends primarily on the participant teams in the WANLP-2021 shared task competition. It is important to note that there is no available corpus for the public except Arsarcasm-v1 [10], Arsarcasm-v2 [19], and the Arabic headlines datasets [62], which indicates the challenge of scarcity in the Arabic resources for sarcasm. In addition, all the collected corpora for Arabic sarcasm detection are small-sized and mostly imbalanced datasets. Regarding the annotation process, either manual or automatic, both have advantages and drawbacks. Therefore, we suggest using the hybrid approach, which is more efficient, although it requires more effort. Further, sarcasm challenges are quite associated with the context and culture in addition to language structure and morphology, which are more complex in Arabic. Regarding the classification phase in Arabic sarcasm detection, it is noteworthy that Transformer-based models are gaining more interest. Specifically, the Arabic versions of BERT-based models performed better than their multilingual counterparts. Concerning performance, it depends on multiple factors. For instance, the corpus size, the model's parameters, preprocessing, and features extraction methods in addition to the machine resources. This indicates that each experiment is standalone and that we cannot assert that one model outperforms the others unless we assume that all factors remain constant.

REFERENCES

[1] S. A. Salloum, A. Q. AlHamad, M. Al-Emran, and K. Shaalan, "A survey of Arabic text mining," in *Intelligent Natural Language Processing: Trends and Applications*. Cham, Switzerland: Springer, 2018, pp. 417–431.
[2] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowl. Inf. Syst.*, vol. 60, pp. 617–663, Jul. 2018.
[3] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013.
[4] S. L. Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali, and I. Abunadi, "Arabic natural language processing and machine learning-based systems," *IEEE Access*, vol. 7, pp. 7011–7020, 2019.
[5] M. M. Abuteir and E. S. Elsamani, "Automatic sarcasm detection in Arabic text: A supervised classification approach," *Int. J. New Technol. Res.*, vol. 7, no. 8, pp. 1–11, 2021.
[6] A. Kamath, R. Guhekar, M. Makwana, and S. N. Dhage, "Sarcasm detection approaches survey," in *Advances in Computer, Communication and Computational Sciences*. Singapore: Springer, 2021, pp. 593–609.
[7] J. Godara, I. Batra, R. Aron, and M. Shabaz, "Ensemble classification approach for sarcasm detection," *Behavioural Neurol.*, vol. 2021, pp. 1–13, Nov. 2021.
[8] D. Al-Ghadhban, E. Alnkhilan, L. Tatwany, and M. Alrazgan, "Arabic sarcasm detection in Twitter," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, May 2017, pp. 1–7.
[9] B. Ghanem, J. Karoui, F. Benamara, V. Moriceau, and P. Rosso, "IDAT@FIRE2019: Overview of the track on irony detection in Arabic tweets," in *Proc. 11th Forum Inf. Retr. Eval.*, 2019, pp. 10–13.
[10] I. A. Farha and W. Magdy, "From Arabic sentiment analysis to sarcasm detection: The ArSarcasm dataset," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, Shared Task Offensive Lang. Detection*, 2020, pp. 32–39.
[11] P. Rosso, F. Rangel, I. H. Farías, L. Cagnina, W. Zaghouani, and A. Charfi, "A survey on author profiling, deception, and irony detection for the Arabic language," *Lang. Linguistics Compass*, vol. 12, no. 4, Apr. 2018, Art. no. e12275.
[12] A. Wadhawan, "AraBERT and Farasa segmentation based approach for sarcasm and sentiment detection in Arabic tweets," 2021, *arXiv:2103.01679*.
[13] A. I. Alharbi and M. Lee, "Multi-task learning using a combination of contextualised and static word embeddings for Arabic sarcasm detection and sentiment analysis," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 318–322.
[14] A. Israeli, Y. Nahum, S. Fine, and K. Bar, "The IDC system for sentiment classification and sarcasm detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 370–375.
[15] H. A. Nayel, W. Medhat, and M. Rashad, "BENHA@IDAT: Improving irony detection in Arabic tweets using ensemble approach," in *Proc. FIRE*, 2019, pp. 401–408.

[16] I. Guellil, F. Azouaou, and M. Mendoza, "Arabic sentiment analysis: Studies, resources, and tools," *Social Netw. Anal. Mining*, vol. 9, no. 1, pp. 1–17, Dec. 2019.

[17] M. S. Razali, A. A. Halin, N. M. Norowi, and S. C. Doraisamy, "The importance of multimodality in sarcasm detection for sentiment analysis," in *Proc. IEEE 15th Student Conf. Res. Develop. (SCOReD)*, Dec. 2017, pp. 56–60.

[18] A. Hengle, A. Kshirsagar, S. Desai, and M. Marathe, "Combining context-free and contextualized representations for Arabic sarcasm detection and sentiment identification," 2021, *arXiv:2103.05683*.

[19] I. A. Farha, W. Zaghouani, and W. Magdy, "Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 296–305.

[20] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," 2020, *arXiv:2003.00104*.

[21] M. Khalifa and N. Hussein, "Ensemble learning for irony detection in Arabic tweets," in *Proc. FIRE*, 2019, pp. 433–438.

[22] D. Ghoul and G. Lejeune, "Sarcasm and sentiment detection in Arabic: Investigating the interest of character-level features," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 329–333.

[23] Y. Jaafar and K. Bouzoubaa, "A survey and comparative study of Arabic NLP architectures," in *Intelligent Natural Language Processing: Trends and Applications*. Cham, Switzerland: Springer, 2018, pp. 585–610.

[24] A. Mehta, Y. Parekh, and S. Karamchandani, "Performance evaluation of machine learning and deep learning techniques for sentiment analysis," in *Information Systems Design and Intelligent Applications*. Singapore: Springer, 2018, pp. 463–471.

[25] M. Alruily, "Classification of Arabic tweets: A review," *Electronics*, vol. 10, no. 10, p. 1143, May 2021.

[26] A. Avvaru, S. Vobilisetty, and R. Mamidi, "Detecting sarcasm in conversation context using transformer-based models," in *Proc. 2nd Workshop Figurative Lang. Process.*, 2020, pp. 98–103.

[27] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, "Sarcasm detection using machine learning algorithms in Twitter: A systematic review," *Int. J. Market Res.*, vol. 62, no. 5, pp. 578–598, Sep. 2020.

[28] L. Bashmal and D. AlZeer, "ArSarcasm shared task: An ensemble BERT model for SarcasmDetection in Arabic tweets," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 323–328.

[29] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis," in *Proc. LREC*, 2014, pp. 1–13.

[30] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 1–22, 2017.

[31] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. Gelbukh, "Sentiment and sarcasm classification with multitask learning," *IEEE Intell. Syst.*, vol. 34, no. 3, pp. 38–43, May/Jun. 2019.

[32] D. Faraj and M. Abdullah, "SarcasmDet at sarcasm detection task 2021 in Arabic using AraBERT pretrained model," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 345–350.

[33] B. Song, C. Pan, S. Wang, and Z. Luo, "DeepBlueAI at WANLP-EACL2021 task 2: A deep ensemble-based method for sarcasm and sentiment detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 390–394.

[34] I. Abbes, W. Zaghouani, O. El-Hardlo, and F. Ashour, "DAICT: A dialectal Arabic irony corpus extracted from Twitter," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 6265–6271.

[35] T. Ranasinghe, H. Saadany, A. Plum, S. Mandhari, E. Mohamed, C. Orasan, and R. Mitkov, "RGCL at IDAT: Deep learning models for irony detection in Arabic language," Tech. Rep., 2019.

[36] J. Iranzo-Sánchez and R. Ruiz-Dolz, "VRAIN at Irosva 2019: Exploring classical and transfer learning approaches to short message irony detection," in *Proc. IberLEF@SEPLN*, pp. 322–328, 2019.

[37] J. Karoui, F. B. Zitoune, and V. Moriceau, "SOUKHRIA: Towards an irony detection system for Arabic in social media," *Proc. Comput. Sci.*, vol. 117, pp. 161–168, Jan. 2017.

[38] A. Khattri, A. Joshi, P. Bhattacharyya, and M. Carman, "Your sentiment precedes you: Using an author's historical tweets to predict sarcasm," in *Proc. 6th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2015, pp. 25–30.

[39] R. Abdel-Salam, "WANLP 2021 shared-task: Towards irony and sentiment detection in Arabic tweets using multi-headed-LSTM-CNN-GRU and MaRBERT," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 306–311.

[40] A. Abuzayed and H. Al-Khalifa, "Sarcasm and sentiment detection in Arabic tweets using BERT-based models and data augmentation," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 312–317.

[41] H. Elgabry, S. Attia, A. Abdel-Rahman, A. Abdel-Ate, and S. Girgis, "A contextual word embedding for Arabic sarcasm detection with random forests," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, pp. 340–344, 2021.

[42] R. Rani and D. K. Lobiyal, "Automatic construction of generic stop words list for Hindi text," *Proc. Comput. Sci.*, vol. 132, pp. 362–370, Jan. 2018.

[43] L. Moudjari and K. Akli-Astouati, "An embedding-based approach for irony detection in Arabic tweets," in *Proc. FIRE*, 2019, pp. 409–415.

[44] N. Kanwar, R. K. Mundotiya, M. Agarwal, and C. Singh, "Emotion based voted classifier for Arabic irony tweet identification," in *Proc. FIRE*, 2019, pp. 426–432.

[45] M. Avinash and E. Sivasankar, "A study of feature extraction techniques for sentiment analysis," in *Emerging Technologies in Data Mining and Information Security*. Cham, Switzerland: Springer, 2019, pp. 475–486.

[46] V. Menger, F. Scheepers, and M. Spruit, "Comparing deep learning and classical machine learning approaches for predicting inpatient violence incidents from clinical text," *Appl. Sci.*, vol. 8, no. 6, p. 981, Jun. 2018.

[47] A. Allaith, M. Shahbaz, and M. Alkoli, "Neural network approach for irony detection from Arabic text on social media," in *Proc. FIRE*, 2019, pp. 445–450.

[48] H. Nayel, E. Amer, A. Allam, and H. Abdallah, "Machine learning-based model for sentiment and sarcasm detection," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 386–389.

[49] Y. Zheng, "An exploration on text classification with classical machine learning algorithm," in *Proc. Int. Conf. Mach. Learn., Big Data Bus. Intell. (MLBDBI)*, Nov. 2019, pp. 81–85.

[50] P. Gupta and M. Jaggi, "Obtaining better static word embeddings using contextual embedding models," 2021, *arXiv:2106.04302*.

[51] M. A. Di Gangi, G. Lo Bosco, and G. Pilato, "Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection," *Natural Lang. Eng.*, vol. 25, no. 2, pp. 257–285, Mar. 2019.

[52] W. Zhou and J. Bloem, "Comparing contextual and static word embeddings with small data," in *Proc. 17th Conf. Natural Lang. Process.*, 2021, pp. 253–259.

[53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[54] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification," 2020, *arXiv:2005.13012*.

[55] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," 2020, *arXiv:2101.01785*.

[56] M. Naski, A. Messaoudi, H. Haddad, M. BenHajhmida, C. Fourati, and A. A. B. E. Mabrouk, "iCompass at shared task on sarcasm and sentiment detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 381–385.

[57] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," 2020, *arXiv:2003.10555*.

[58] W. Antoun, F. Baly, and H. Hajj, "AraELECTRA: Pre-training text discriminators for Arabic language understanding," 2020, *arXiv:2012.15516*.

[59] I. A. Farha and W. Magdy, "Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection," in *Proc. 6th Arabic Natural Language Process. Workshop*, 2021, pp. 21–31.

[60] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," Tech. Rep., 2018.

[61] W. Antoun, F. Baly, and H. Hajj, "AraGPT2: Pre-trained transformer for Arabic language generation," 2020, *arXiv:2012.15520*.

[62] P. Mohammed, Y. Eid, M. Badawy, and A. Hassan, "Evaluation of different sarcasm detection models for Arabic news headlines," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.* Cham, Switzerland: Springer, 2020, pp. 418–426.

[63] B. Talafha, M. E. Za'Ter, S. Suleiman, M. Al-Ayyoub, and M. N. Al-Kabi, "Sarcasm detection and quantification in Arabic tweets," in *Proc. IEEE 33rd Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2021, pp. 1121–1125.

[64] L. Alhaidari, K. Alyoubi, and F. Alotaibi, "Detecting irony in Arabic microblogs using deep convolutional neural networks," *Int. J. Adv. Comput. Sci. Appl.*, pp. 1–14 vol. 13, no. 1, 2022.

[65] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 502–518.
[66] M. Nabil, M. Aly, and A. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2515–2519.
[67] C. Zhang and M. Abdul-Mageed, "Multi-task bidirectional transformer representations for irony detection," 2019, *arXiv:1909.03526*.
[68] H. A. Nayel, "Irony detection in Arabic tweets: Technical report," Tech. Rep.
[69] S. Kayalvizhi, D. Thenmozhi, B. S. Kumar, and C. Aravindan, "SSN_NLP@IDAT-FIRE-2019: Irony detection in Arabic tweets using deep learning and features-based approaches," in *Proc. FIRE*, 2019, pp. 439–444.
[70] Y. Sharma and A. V. Mandalam, "Irony detection in non-English tweets," in *Proc. 6th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2021, pp. 1–6.
[71] A. E. Mahdaouy, A. E. Mekki, K. Essefar, N. E. Mamoun, I. Berrada, and A. Khoumsi, "Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language," 2021, *arXiv:2106.12488*.
[72] F. Husain and O. Uzuner, "Leveraging offensive language for sarcasm and sentiment detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 364–369.
[73] M. Lichouri, M. Abbas, B. Benaziz, A. Zitouni, and K. Lounnas, "Preprocessing solutions for detection of sarcasm and sentiment for Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 376–380.
[74] I. A. Farha and W. Magdy, "Mazajak: An online Arabic sentiment analyser," in *Proc. 4th Arabic Natural Lang. Process. Workshop*, 2019, pp. 192–198.
[75] W. Lan, Y. Chen, W. Xu, and A. Ritter, "An empirical study of pre-trained transformers for Arabic information extraction," 2020, *arXiv:2004.14519*.
[76] A. Safaya, M. Abdullatif, and D. Yuret, "BERT-CNN for offensive speech identification in social media," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 1–16.
[77] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.
[78] J. Aboobaker and E. Ilavarasan, "A survey on sarcasm detection and challenges," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 1234–1240.
[79] D. H. Farias and P. Rosso, "Irony, sarcasm, and sentiment analysis," in *Sentiment Analysis in Social Networks*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 113–128.

ALAA RAHMA received the B.A. degree from Al-Azhar University, Egypt, in 2017, and the postgraduate Diploma and premaster's degrees in computer science from the Faculty of Graduate Studies for Statistical Researches, Cairo University, Egypt, in 2020 and 2022, respectively, where she is currently pursuing the master's degree with the Faculty of Graduate Studies for Statistical Researches. Her research interests include machine and deep learning applications and natural language processing.

SHAHIRA SHAABAN AZAB received the M.Sc. and Ph.D. degrees from Cairo University, Cairo, Egypt, in 2010 and 2018, respectively. She is currently an Assistant Professor with the Computer Science Department, Faculty of Graduate Studies of Statistical Research, Cairo University. Her research interests include machine learning, deep learning, optimization, and natural language processing.

AMMAR MOHAMMED received the bachelor's and master's degrees in computer science from Cairo University, Egypt, and the Ph.D. degree in computer science from the University of Koblenz-Landau, Germany, in 2010. He worked as a Researcher and a Research Fellow at the Artificial Intelligence (AI) Research Group, University of Koblenz-Landau. He is currently a Computer Science Professor with Cairo University and University for Modern Sciences and Arts. He supervised a group of Ph.D. and master's students. He has established the Machine/Deep Learning Research Group at the Department of Computer Science, Faculty of Graduate Studies for Statistical Research, Cairo University. His research interests include machine and deep learning techniques, methods, algorithms, and applications in several domains.