Multilingual Sarcasm Detection for Enhancing Sentiment Analysis using Deep Learning Algorithms

Ahmed Derbala Yacoub, Amal Elsayed Aboutabl, and Salwa O. Slim

Original scientific article

***Abstract*—Recent years have seen a notable rise in online opinion-sharing, underscoring the demand for automated sentiment analysis tools. Addressing sarcasm in text is crucial, as it can significantly influence the effectiveness of sentiment analysis models. This research explores how sentiment analysis (SA) and sarcasm detection (SD) intersect, highlighting challenges in identifying how sarcasm influences sentiment polarity. Sarcasm, a type of irony, poses computational difficulties due to the lack of nonverbal cues in written texts. Users often express opinions in their preferred languages, underscoring the need for sentiment analysis tools that can adeptly handle sentiment and sarcasm across diverse languages. We propose the incorporation of sarcasm features into the architecture of sentiment analysis models, employing classifiers and embeddings, including BILSTM or LSTM alongside word embedding techniques such as Word2vec, FastText, Glove, and Bert. We conducted experiments using the ArSarcasm-v2 Dataset for Arabic, the IMDB Movie dataset and IsarcasmEval dataset for English, and the SentiMixArEn dataset for code-mixed language scenarios. The results demonstrated consistent accuracy enhancements ranging from 2% to over 10%, highlighting the positive impact of incorporating sarcasm-related information. Additionally, the Bi-LSTM model with GloVe embeddings achieved higher accuracy across all scenarios compared to other methods.**

***Index Terms*—machine learning, deep learning, natural language processing, sarcasm detection, sentiment analysis.**

I. Introduction

NATURAL language processing is a subfield of machine learning, that aims to enable computers to analyze and understand human language. This involves the development of models and algorithms that can effectively process and analyze data presented in natural language, whether in the form of speech or text [1]. For sentiment analysis, NLP plays a significant role in enabling the automatic analysis of sentiments conveyed in the text [2]. The sentiment analysis's goal is to determine the expressed sentiment polarity (i.e., negative, neutral, or positive) in a text by developing models that can analyze people's feelings or beliefs expressed in texts such as emotions, opinions, attitudes, and appraisals, among others [3-6]. Sentiment analysis is used for many applications such as modeling user preferences, monitoring consumer behaviors, digital marketing, product review analysis, customer feedback, social media monitoring, etc. [7]. Existing research primarily concentrates on sentiment classification or sarcasm detection separately, overlooking the strong correlation between the two tasks [7]. Challenges within sentiment analysis encompass difficulties in precise polarity classification (positive, negative, or neutral), addressing sentiment at different text levels, identifying reader sentiment, sentiments related to the mentioned entities, and parsing semantic roles within the text [8]. Sentiment analysis currently focuses on English, overlooking the rise of native language use on platforms like Facebook and Twitter, underscoring the necessity for adopting multilingual approaches to improve accuracy in interpreting diverse linguistic expressions across social media [9]. Additionally, users often use sarcasm to convey sentiments, requiring the identification of sarcasm and subsequent adjustment of polarity based on its presence in the sentence [10].

Sarcasm is a type of irony where the stated meaning is contrary to the intended meaning, and its detection in text poses challenges for computers [11]. Identifying sarcasm in text poses a challenge for computers as it involves recognizing the contrast between the stated and intended meanings, and the absence of nonverbal cues like tone and gestures in written communication adds to the difficulty [11]. Detecting sarcasm is crucial for improving sentiment analysis. The challenge arises from sarcasm devices that can reverse expressed sentiment, emphasizing the need for developing sentiment analysis tools that understand and detect sarcasm [12]. Identifying sarcasm is a complex task involving determining if a sentence contains sarcasm, complicated by challenges in classifying textual data and extracting information to discern the often negative sentiments expressed through positive language [13]. Identifying sarcasm is challenging due to its varied expressions, the contextual importance of text, and the influence of tone and style on detection, highlighting the complexity of sarcasm detection [14]. Recent developments in deep learning leverage neural networks to learn these features, eliminating the need for handcrafted features [14]. Recognizing sarcasm in text is challenging, demanding a high level of intelligence, and machine learning, particularly deep learning, has proven effective in this process by learning contextual and lexical features [15]. This research aims to identify challenges in sentiment analysis and understand the role of sarcasm detection. It also seeks to develop a system for analyzing

Manuscript received August 9, 2024; revised August 24, 2024. Date of publication October 25, 2024. Date of current version October 25, 2024.

Authors are with the Computer Science Department, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt (e-mails: ahmed_drbala_1073@fci.helwan.edu.eg, amal.aboutabl@fci.helwan.edu.eg, salwaosama@fci.helwan.edu.eg).

Digital Object Identifier (DOI): 10.24138/jcomss-2024-0071

sentiments, focusing on sarcasm detection to improve accuracy in determining emotional polarity. This research will look into current methods and future directions in sentiment analysis and sarcasm detection. Additionally, it will explore the use of sentiment analysis in natural language processing and examine its various applications. Finally, the research will test and evaluate the proposed model, showing why integrating sarcasm detection with sentiment analysis is important. In this research, we explore three key questions that shape our investigation: What are sentiment and sarcasm, and how do they relate to each other? How can sarcasm be detected, and how might this detection be integrated into sentiment analysis without relying on specific contextual cues? Finally, how can we develop an efficient model for sentiment analysis and sarcasm identification, particularly in the context of written text, whether in a single language or across multiple languages?

Below are the key challenges in sentiment analysis:

- Difficulty in determining sentiment in sarcastic sentences, impacting intended meaning.
- Challenges in detecting sarcasm due to context absence, diverse writing styles, data classification limitations, and the implicit nature of sarcasm.
- Complexities in classifying sentiment and sarcasm in multilingual contexts.
- Challenges in sentiment analysis include dealing with implicit text that lacks contextual information and addressing issues in written communication caused by the absence of non-verbal cues such as gestures and facial expressions.
- Limitations in word vectors focusing on word meaning, potentially leading to similar vectors for different sentiments.
- Exclusion of non-English text affects accuracy and may introduce bias.

Sarcasm detection and sentiment analysis methods can be categorized into three types: machine learning, lexical-based, and hybrid approaches. In machine learning, algorithms are employed to find patterns in large sets of data. Deep learning, especially recurrent neural networks (RNNs), is a flexible framework for understanding written information. Machine learning techniques include supervised learning, which uses labeled training data for accurate predictions, and unsupervised learning, which discovers patterns without labeled data. Common machine learning algorithms like Naive Bayesian, Support Vector Machine, Artificial Neural Networks, and the K-means algorithm help analyze sentiments. Hybrid approaches combine aspects of machine learning and lexical-based methods to improve accuracy in identifying subtle sentiments and sarcasm in written content. Researchers have employed a variety of sophisticated methods in sarcasm detection and sentiment analysis. Notably, Shah et al. (2) addressed sarcasm detection in Arabic text by introducing a modified switch transformer architecture, achieving an 83% accuracy in ArSarcasm-v2dataset. Yin et al. (7) introduced a multi-tasks deep neural network, MT SS, incorporating bidirectional gated recurrent units (Bi-GRU) and attention mechanisms for simultaneous sarcasm and sentiment analysis. Rao et al. (8) devised a robust sarcasm detection methodology utilizing a Twitter headlines dataset, employing meticulous data preprocessing and training a recurrent neural network (RNN) with long short-term memory (LSTM) to achieve a remarkable 92% accuracy. Shrivastava et al. (11) organized a comprehensive sarcasm detection model into data preparation, fine-tuning, and classification modules, incorporating the Next Sentence Prediction (NSP) module and BERT, achieving 68% accuracy on the SemEval 2018 Task 3 dataset. Rahaman Wahab Sait et al. (14) integrated N-gram feature extraction and a Multi-Head Self-Attention-based gated recurrent unit model, achieving an impressive accuracy rate of 97.61% on the Twitter dataset.

The key contributions of this paper are as follows:

- Development of a novel architecture that integrates sarcasm detection into sentiment analysis to improve the accuracy of the sentiment analysis model and explicitly demonstrate the influence of identifying sarcasm on the accuracy of sentiment analysis through systematic comparisons.
- This architecture is capable of handling multilingual contexts, focused on Arabic, English, and code-mixed Arabic-English datasets.
- Transfer learning is employed through pre-trained word embedding techniques instead of training embedding models from scratch. This approach saves time and enhances accuracy by leveraging existing representations learned from extensive text data.
- Generating a new code-mixed Arabic-English dataset customized for sentiment analysis tasks represents a significant contribution, especially considering the limited availability of datasets in this domain.

The rest of the paper is organized in this way: Section II offers a review of related work. Section III describes the proposed system. Section IV discusses the dataset. Section V outlines the experimental results. Section VI discusses our findings. Section VII compares the proposed research to recent studies. Section VIII concludes with a summary and future research directions.

II. Related Work

Natural language processing has advanced significantly in sentiment analysis and sarcasm detection. Sentiment analysis identifies emotional tones in text, while sarcasm detection focuses on understanding non-literal statements. This section reviews the literature on these topics, categorizing methods into three groups: sentiment analysis, sarcasm analysis, and combined sentiment and sarcasm approaches. The overview highlights recent developments from the last five years, clarifying the evolution and advancements in these methods.

A. Methods for Sentiment Analysis

Sentiment analysis involves the extraction of emotional tone from text, which is crucial for applications ranging from customer feedback analysis to social media monitoring. This subsection reviews methods specifically designed to tackle sentiment analysis, highlighting their techniques, datasets, and performance metrics.

Draskovic et al. [1] introduced a specialized sentiment analysis model for the Serbian language to address the challenges associated with sentiment classification in a linguistically resource-limited environment. The model takes into account diverse text attributes, including the number of attributes, stop words, number of n-grams, and attribute type. The researchers employed three common classification algorithms (naive Bayes, support vector machine, and logistic regression) individually and in combination to optimize performance. Training the models involved three distinct datasets, and the evaluation was conducted on a collection of Serbian music reviews. The NB-LR hybrid model achieved the highest accuracy, with 79% for binary classification and 58% for three-class classification. It performs well in binary tasks but struggles with neutral class separation in three-class scenarios.

Graff et al.[3] presented evolutionary multilingual sentiment analysis (EvoMSA), a sentiment analysis system designed to provide a domain-independent and multilingual approach for sentiment analysis in written text. This model unifies multiple text classification methods and consistently achieves high rankings in sentiment analysis competitions across various languages. EvoMSA comprises two stages: the first stage involves five text models that transform the text into decision function values, while the second stage employs an evolutionary direct acyclic graph (EvoDAG), a genetic programming-based classifier, to make the final sentiment prediction. EvoMSA achieved notable results, including first places in TASS 2018 and IberEval 2018 tasks, with strong performances in the HAHA task (F1 score of 79.9%) and the S2 task (macro-F1 score of 72.5%).

Pragati Goel et al. [4] developed a language-independent system for efficient sentiment analysis of multilingual Twitter data to overcome the limitations of prior research, which focused only on English tweets. They utilize the Google Translator API to transform tweets in multiple languages into English and employ pre-processing, data modeling, feature extraction, and a Naïve Bayes classifier. A distinguishing feature of their approach is the integration of Stanford NLP to model the English language comprehensively, enhancing the training process. The results indicate an accuracy of 79.4% and a precision of 0.78 with a dataset of 2000 tweets.

Zahedi et al. [5] developed a model for sentiment classification on a multilingual dataset of tweets in English, Urdu, and Roman Urdu. They used machine learning techniques including Logistic Regression, Linear Support Vector Classifier, Stochastic Gradient Descent, Multinomial Naïve Bayes, and Complement Naïve Bayes. The dataset consisted of 200k tweets in English and Urdu, and 11k in Roman Urdu. Data preprocessing involved cleaning, stop word removal, and normalization, with feature extraction using the TF-IDF vectorizer. An ensemble technique is used to merge predictions from multiple models, with Logistic Regression achieving the highest accuracy of 75%.

Arun et al. [6] introduce the Multilingual Twitter for sentiment analysis (MLTSA) algorithm as a solution to challenges in sentiment analysis of non-English tweets. Classified into two parts, the algorithm first detects and utilizes Natural Language Processing (NLP) to convert tweets in languages other than English into English and then employs NLP-supported pre-processing to mitigate data sparsity. Utilizing Support Vector Machines (SVM), MLTSA achieves a commendable accuracy of up to 95%. The dataset, sourced from Twitter's API, centers around tweets related to the Chief Minister of Andhra Pradesh in various languages, predominantly English, Hindi, and Telugu.

Londhe et al. [9] developed a sentiment analysis system for predicting sentiment from multilingual user comments, overcoming the challenge of limited language dictionaries. Their hybrid model combines the Social Eagle Algorithm (SoEo) with a deep Bidirectional LSTM classifier. The process involves transliterating languages into a common format, standardizing slang, emojis, and abbreviations, and using TF-IDF for feature extraction. The SoEo algorithm, inspired by eagle hunting and coyote adaptation, improves solution search through phases of hand-pick, scavenge, and descend. This model achieves high performance with 91.572% accuracy, 89.196% precision, 91.551% recall, and 89.019% F1 measure.

B. Methods for Sarcasm Detection

Sarcasm detection is a specialized task that requires understanding the subtleties of language to identify statements meant to convey irony or humor. This subsection reviews methods focused on sarcasm analysis, detailing their techniques, datasets, and performance outcomes.

Rao et al. [8] developed for sarcasm detection using Twitter data involved selecting the 'Twitter headlines dataset' from Kaggle, containing about 30,000 tweets. Data was preprocessed through tokenization, stemming, lemmatization, and word embeddings, and then used to train an LSTM-based recurrent neural network (RNN). The model achieved up to 92% accuracy after multiple training epochs.

Gupta et al. [10] developed a sentiment analysis method for social network interactions that focuses on emoticons and sarcasm. The process begins with uploading unfiltered text, followed by removing stop words and refining the text. A neural network is then trained to classify text as positive or negative, using a generator to aid in this classification. The system also identifies emoticons and retrains the neural network to detect sarcasm. The method achieved a notable 96.34% accuracy in distinguishing sentiments in both sarcastic remarks and emoticons.

Shrivastava et al. [11] developed a sarcasm detection model with three main components: Data Preparation, Model Fine Tuning, and Classification. The Data Preparation Module focused on handling unstructured data by preprocessing and organizing it into structured columns, including adapting the Next Sentence Prediction (NSP) into a classification task with new labels for sarcasm. The Fine-Tuning Module involved classifying data using a BERT tokenizer and pretrained model. Finally, the Classification Module used a feedforward neural network to identify sarcasm in the processed data. Their model, tested on the SemEval 2018 Task 3 dataset, achieved a 68% accuracy, demonstrating its effectiveness in detecting sarcasm.

Rahaman Wahab Sait et al. [14] presented a comprehensive approach for the detection and classification of sarcasm in text data. This method incorporates multiple stages, beginning with text pre-processing involving tasks like lemmatization, stopword removal, and tokenization. It continues with N-gram feature extraction, combining neighboring tokens into n-grams, followed by a classification step employing a Multi-Head Self-Attention-based Gated Recurrent Unit (MHSA-GRU) model. This study achieved an impressive accuracy rate of 97.61% on the Twitter dataset and compared it to other state-of-the-art methods.

Murali Krishna et al. [15] introduced a model that utilizes a recurrent neural network for sarcasm detection in text. This method incorporates a two-level classification system that considers both emotional and semantic dimensions within the text. The first level focuses on finding sentiment hard attention from each word within the sentence, while the second level is focused on identifying the underlying semantics present in it. The semantic representation of sentiment analyzed from tweets is obtained through mathematical representations involving tanh and soft-max functions. For manually annotated datasets, it achieves an accuracy level of up to 78.42%, while for automatically annotated datasets, the accuracy is even higher, reaching up to 97.06%.

Ghanem et al. [16] developed a multilingual and multicultural system for irony detection across English, French, and Arabic. They compared monolingual approaches using feature-based methods and neural architectures with monolingual word embeddings. The findings show that semantic-based embeddings outperform lexicon-based and surface features. Cross-lingual performance varied due to dialect words and the need for text-based features for weak semantic detection. CNN models achieved F1 scores between 34.1% and 62.4%, with the best results in French-Arabic and English-Arabic pairs. RF models showed F1 scores ranging from 42.9% to 74.6%, performing best in Arabic-English and Arabic-French contexts.

Cignarella et al. [17] investigate the efficacy of features based on syntax dependency in multilingual irony detection. They conduct three distinct experimental settings, examining the impact of these features when combined with classical machine learning models, the use of various word embeddings including fastText and dependency-based embeddings, and the integration of syntactic features into the state-of-the-art BERT model for irony detection. The results demonstrate that the inclusion of syntax features can enhance model performance. The integration of dependency-based syntactic features into M-BERT generally improved irony detection, with notable gains in F1-scores: English (68.2%), Spanish (66.8%), French (78.5%), and Italian (70.3%).

Han et al. [18] developed the X-PuDu system for the ISarcasmEval and SemEval-2022 Task 6, aimed at detecting sarcasm in English and Arabic. The system uses a processing step: pre-training, and fine-tuning with large-scale language models like ERNIE-M and DeBERTa. For single-sentence tasks, it fine-tunes pre-trained Transformers, while for sentence-pair tasks, it employs multi-layer Transformer blocks. It enhances performance by fine-tuning on both Arabic and English data and uses ensemble learning with k-fold cross-validation to optimize model selection. The ERNIE-M model achieved 82.50% accuracy in English and 90.50% in Arabic.

C. Methods for Both Sentiment and Sarcasm Detection

Combining sentiment and sarcasm analysis within a single model offers a more integrated approach to understanding complex emotional content. This subsection reviews methods that address both sentiment and sarcasm, focusing on their integrated techniques and overall performance.

Shah et al. [2] developed a new approach for detecting irony and sarcasm in Arabic text using a modified switch transformer architecture. They introduce Variational Enmesh Expert's Routing (VEeR) and Probabilistic Projections to enhance embedding generation. By dynamically controlling the flow of information with variational techniques, their model adapts to different input paths. Their method achieved an 83% accuracy in sarcasm detection and 51% in sentiment classification, using the ArSarcasm-v2 dataset of Arabic tweets.

Yin et al. [7] introduced MT_SS, a deep neural network designed for multitasking to detect sarcasm and perform sentiment analysis. The model utilizes bi-directional gated recurrent units (BiGRU) with attention mechanisms to capture both local and global sentence features. It incorporates GloVe embeddings for word representation, BiGRU for contextual encoding, and task-specific fully connected layers for sarcasm and sentiment classification. Additionally, it uses CNNs for spatial features and a bilinear tensor layer for intertask communication. The final classifications are made with concatenated representations and softmax classifiers. MT_SS outperformed existing methods, achieving 90.03% and 92.01% accuracy for sarcasm and sentiment analysis on Dataset 1, and 91.04% and 92.28% on Dataset 2.

Mahdaouy et al. [12] developed a multi-task learning (MTL) model that integrates a Transformer-based BERT encoder with a multi-task attention module and two classifiers for sarcasm detection (SD) and sentiment analysis (SA). The BERT encoder, using pre-trained MARBERT, generates contextualized word embeddings from tweets. The model features a multi-task attention module with task-specific layers and a Sigmoid task-interaction layer, along with classifiers Fsarc for sarcasm and Fsent for sentiment. This MTL model, named ATTINTER, outperforms single-task models, achieving 76.80% accuracy in sarcasm detection and 71.07% in sentiment analysis, by leveraging the relationship between negative sentiment and sarcasm.

Naski et al. [13] introduced a method that utilizes pre-trained contextualized text representation models for natural language understanding, with a particular focus on Arabic language comprehension. They extensively explored various BERT models, including a multilingual cased BERT (mBERT) and specialized Arabic BERT variants (AraBERT, ARBERT, and MARBERT). The study's primary objectives encompassed sarcasm detection and sentiment analysis in Arabic text. They achieved an accuracy of 68.8% in sarcasm detection, and for sentiment analysis, they achieved an accuracy of 72%.

A summary of the results from these studies is provided in Table I, offering a concise comparison of their performance metrics.

TABLE I
SUMMARY OF RELATED WORK PERFORMANCE METRICS

Study	Task	Accuracy(%)	
		Sarcasm	sentiment
Draskovic[1]	Sentiment	-	79
Graff[3]	Sentiment	-	79.9
Pragati[4]	Sentiment	-	79.4
Zahedi[5]	Sentiment	-	75
Arunet[6]	Sentiment	-	95
Londhe[9]	Sentiment	-	91.572
Rao[8]	Sarcasm	92	-
Gupta[10]	sarcasm	96.34	-
Shrivastava[11]	Sarcasm	68	-
Rahaman[14]	Sarcasm	97.61	-
Murali[15]	Sarcasm	78.42	-
Ghanemet[16]	Sarcasm	F1:74.6	-
Cignarellaet[17]	Sarcasm	F1:78.5	-
Hanet[18]	Sarcasm	90.50	-
Shahet[2]	Sarcasm, Sentiment	83	51
Yin[7]	Sarcasm, Sentiment	92.01	90.03
Mahdaouy][12]	Sarcasm, Sentiment	76.8	71.07
Naski][13]	Sarcasm, Sentiment	68.8	72

III. PROPOSED SYSTEM

The proposed system aims to effectively handle multilingual sarcasm detection and sentiment analysis tasks, leveraging pre-trained word embedding and deep learning models for robust results and evaluated using k-fold cross-validation. The proposed system combines sarcasm detection and sentiment analysis using advanced deep learning methods, including Recurrent Neural Networks (RNNs) models. It pre-processes text data based on language detection, trains a sarcasm detection model, incorporates sarcasm features into a sentiment analysis model, and evaluates overall performance through k-fold cross-validation, showcasing its effectiveness in capturing linguistic nuances for both tasks as shown in Figure 1.

A. Language Identification

In this study, which covers two languages, Arabic and English, the proposed system is designed to identify the language of the dataset, differentiating between Arabic content, English content, and code-mixed content. This task is accomplished using the Python Langdetect library. The outcomes from this language identification step are essential for guiding the subsequent processes. During training, both sentiment and sarcasm datasets are utilized as inputs, whereas during classification, only the testing samples are processed.

B. Translation

The proposed system uses the Google API Translator to convert the text in the code-mixed dataset into English text.

C. Pre-processing

The proposed system involves two languages, Arabic and English. According to the dataset language recognized from the language identification phase, the pre-processing phase loads the dedicated language processing functions. The pre-processing phase encompasses the following steps:

- Common Pre-processing Steps for Both Languages:
  - Eliminate noise and insignificant elements like usernames, special characters, numbers, html tags, and urls.
  - Remove hashtags, as this study operates under the assumption of an unknown context.
  - Address repeated characters.
  - Substitute emojis with their corresponding text in the detected language.
  - remove stop-words according to detected language.
- For the Arabic language, remove non-Arabic text and perform text normalization.

D. Feature Extraction

This phase consists of the following actions:

- Generate tokens for all words in sarcasm and sentiment datasets
- Utilize a pre-existing word embedding model and generate an embedding representation encompassing the embeddings for all words from both datasets.

E. Sarcasm Detection

The proposed system employs a deep learning architecture that uses Recurrent Neural Networks (RNNs) to address both sarcasm detection and sentiment analysis.
The steps involved are as follows:

- Train a classifier using the word embeddings from the sarcasm detection dataset, which were obtained during feature extraction phase, and refer to it as the sarcasm detection model.
- Evaluate the sarcasm detection model's performance through k-fold cross-validation.
- Utilize the sarcasm model to extract sarcasm-related features from the embedding of the sentiment analysis dataset, which will be used in the subsequent sentiment analysis phase.

F. Sentiment Analysis

The following steps are involved:

- Combine the extracted sarcasm features with the sentiment analysis dataset embeddings.
- Train a classifier using this combined data and refer to it as the sentiment analysis model.
- Evaluate the overall performance of the sentiment analysis model, which encompasses both sarcasm detection and sentiment analysis components using K-fold cross-validation.

IV. DATASET

The proposed system makes use of two separate datasets: one for the sentiment analysis task and another for the sarcasm detection task. Alternatively, the model has the flexibility to utilize a single dataset for both tasks. This requires the dataset to include labels for both sentiments (positive, negative, or neutral) and sarcasm (sarcastic or not sarcastic). The proposed system employed established benchmark datasets, with samples for each dataset illustrated in Table II.

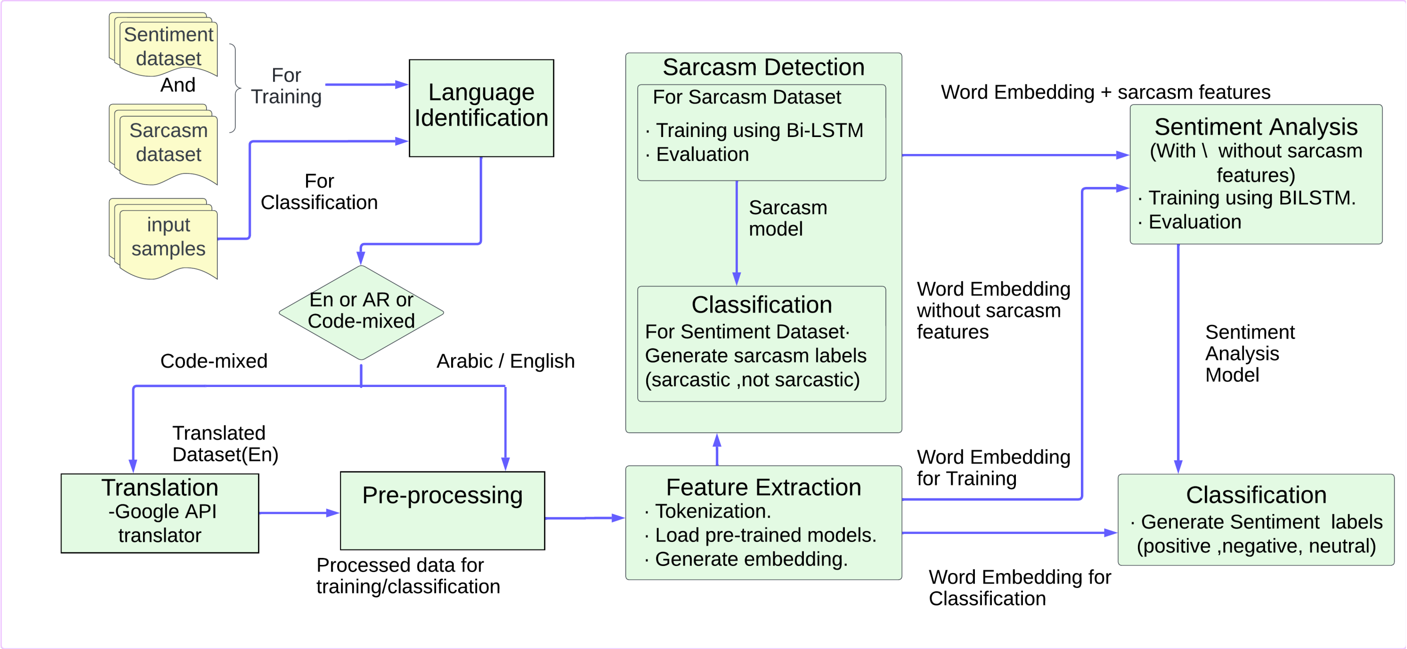
Sentiment dataset
And
Sarcasm dataset
For Training
input samples
For Classification
Language Identification
En or AR or Code-mixed
Code-mixed
Arabic / English
Translated Dataset(En)
Translation -Google API translator
Pre-processing
Processed data for training/classification
Sarcasm Detection
For Sarcasm Dataset
· Training using Bi-LSTM
· Evaluation
Sarcasm model
Classification
For Sentiment Dataset· Generate sarcasm labels (sarcastic ,not sarcastic)
Feature Extraction
· Tokenization.
· Load pre-trained models.
· Generate embedding.
Word Embedding + sarcasm features
Word Embedding without sarcasm features
Word Embedding for Training
Word Embedding for Classification
Sentiment Analysis (With \ without sarcasm features)
· Training using BILSTM.
· Evaluation
Sentiment Analysis Model
Classification
· Generate Sentiment labels (positive ,negative, neutral)

Fig. 1. Proposed system

- ArSarcasm-v2 Dataset [19]: The ArSarcasm-v2 dataset is an expansion from the initial ArSarcasm dataset created as part of a shared task. It comprises 15,548 tweets sourced from ArSarcasm dataset (SemEval 2017 and ASTD). The dataset has been annotated with sarcasm and dialect labels.
- iSarcasmEval dataset [20]: The iSarcasmEval dataset, customized for SemEval 2022 Task 6, showcases self-labeled sarcasm data in English and Arabic, with authors assigning sarcasm labels directly. This dataset comprises 4,867 texts.
- IMDB Movie Reviews dataset [21]: The IMDB dataset, designed for sentiment analysis in English, comprises 50,000 movie reviews and is intended for binary sentiment classification.
- The SentiMixArEn Dataset: is a new code-mixed dataset in Arabic-English tailored for sentiment analysis tasks, comprising 1,007 instances. We created the SentiMixArEn dataset from the SentMixA-3L [22] dataset by translating non-English text into Arabic.

V. Experimental Results

Our experiment aims to explore the impact of incorporating sarcasm in sentiment analysis. We hypothesized that adding sarcasm detection would improve sentiment analysis accuracy. For word embedding, we utilized pre-trained models to represent textual features as described in Table III. This step is crucial as deep learning models require numerical input. Transfer learning was employed to leverage pre-learned features, accelerating model training, and enhancing performance. The model is trained with a dynamic learning rate and early stopping to optimize convergence and prevent over-fitting. Training occurs over a maximum of 100 epochs, utilizing a batch size of 256. In the performance evaluation phase, we used accuracy as the primary metric, measured through k-fold cross-validation (k=10). Accuracy calculates the ratio of accurately classified instances to the overall number of instances within the dataset. This metric provides a straightforward assessment of the model's classification performance, aligning with commonly used evaluation practices in machine learning, particularly for classification tasks. In this experiment, we present a comparative analysis of two sentiment analysis models: the proposed system, referred to as (SA+SD), and a baseline sentiment analysis model without the incorporation of a sarcasm detection feature, referred to as (SA). (SA+SD) represents our proposed system which integrates both sentiment analysis and sarcasm detection.

TABLE II
Text sample within the dataset

Dataset	Text	Labels
ArSarcasm-V2 [19]		Negative, Sarcastic
ArSarcasm-V2 [19]		Negative, Not sarcastic
Isarcasm Eval [20]	Maccies are becoming too regular	Not sarcastic
Isarcasm Eval [20]	The only thing I got from college is a caffeine addiction	Sarcastic
IMDB [21]	A wonderful little production. The filming technique is very unassuming- very old-time-BBC fashion ..	Positive
IMDB [21]	This movie was a failure as a comedy and a film in general. It was a very slow paced movie, …	Negative
SentiMix ArEn [22]	This morning's sunrise was absolutely stunning	Positive
SentiMix ArEn [22]	but now I feel let down and helpless	Negative

A. Experiment 1: Arabic-language Dataset

1) Goal: In Arabic-language dataset, the goal of this experiment section is to compare various classifiers, such as LSTM and BiLSTM, along with various embedding techniques, to

TABLE III
USED PRE-TRAINED WORD EMBEDDING MODELS

Model	Source
Arabic Word2Vec	Full Grams CBOW 300 Twitter
Arabic GloVe	Arabic Corpus (1.75B tokens, 1.5M vocab, 256d vectors)
Arabic Fast-Text	Common Crawl/Wikipedia using fastText (CBOW, dim-300, char n-grams 5)
Arabic Bert	AraBERT v2 aubmindlab/bert-base-arabertv02
English Word2Vec	GoogleNews-vectors-negative300
English GloVe	Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download)
English FastText	2 million word vectors generated through training on Common Crawl, encompassing 600 billion tokens
English Bert	bert-base-uncased

determine the optimal classifier and embedding method for sarcasm detection in sentiment analysis. Additionally, the experiment seeks to assess how sarcasm-detected features influence the performance of sentiment classification models using various classifiers and embedding techniques.

2) Setup: This experiment section employs the ArSarcasm-v2 dataset for the sarcasm detection and sentiment analysis task. Each entry in the dataset is annotated with labels for both sarcasm and sentiments.

3) Results: The results presented in Figure 2 and Table IV demonstrate a notable performance improvement when incorporating sarcasm features alongside sentiment analysis in the models. Across all classifiers and feature extraction methods, the sentiment analysis combined with sarcasm detection consistently outperforms sentiment analysis alone in terms of accuracy. For instance, BILSTM models with GloVe embeddings exhibit the highest accuracy, reaching 94.55% for sarcasm detection, 80.87% for sentiment analysis, and 90.53% when both features are combined. This suggests that integrating sarcasm detection features enhances the model's ability to discern subtle nuances in language, contributing to a more comprehensive sentiment analysis. The impact of sarcasm features is particularly evident in models utilizing bidirectional LSTM architecture, emphasizing the importance of context understanding for accurate sentiment interpretation, especially in scenarios where sarcasm is prevalent. It is worth highlighting that the incorporation of sarcasm features into the model of sentiment analysis resulted in a noticeable improvement in the sentiment analysis task accuracy as evidenced in Table V and Table VI. For the BILSTM classifier, there is a consistent improvement in sentiment analysis accuracy when sarcasm features are added, with the joint Sarcasm Detection and sentiment analysis outperforming the sentiment analysis without joint sarcasm detection features across all k-folds. Notably, the accuracy improvement ranges from approximately 2% to over 10%, showcasing the substantial positive impact of considering sarcasm-related information. Similarly, the LSTM classifier demonstrates consistent accuracy enhancements with the inclusion of sarcasm features, with the joint Sarcasm Detection and sentiment analysis consistently surpassing the sentiment analysis without joint sarcasm detection features. The results highlight the significance of incorporating sarcasm features to enhance sentiment analysis models, providing valuable insights for model optimization in scenarios involving sarcasm detection in the ArSarcasm-v2 Dataset. This finding demonstrated that for datasets with sarcastic expressions, it is crucial to use a sentiment analysis model that incorporates sarcasm-related features. This is important because sarcasm can alter sentiment polarity, making accurate sentiment analysis challenging. By integrating sarcasm detection, our model can better interpret the nuanced meanings of sarcastic remarks.

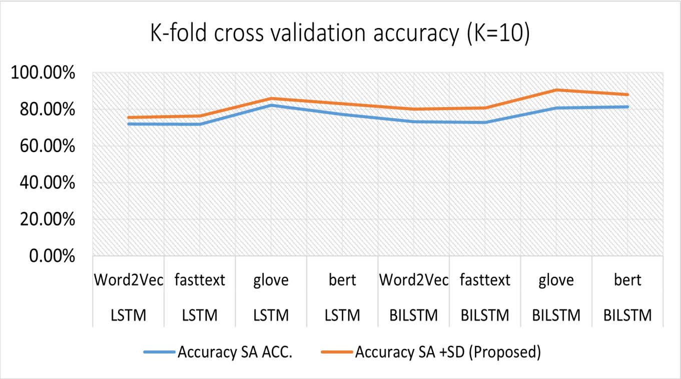

Fig. 2. K-fold cross-validation accuracy (k=10) on ArSarcasm-v2 dataset

TABLE IV
ACCURACY OF 10-FOLD CROSS-VALIDATION FOR ARSARCASM-V2 DATASET

Classifier	Feature Extractor	Accuracy (%)		
		SD	SA	SA+SD (Proposed)
LSTM	Word2Vec	86.22	72.07	75.64
LSTM	fasttext	84.86	71.84	76.31
LSTM	glove	91.0	82.26	85.96
LSTM	bert	87.03	77.16	83.14
BILSTM	Word2Vec	85.80	73.26	80.24
BILSTM	fasttext	85.75	72.92	80.81
BILSTM	glove	94.55	80.87	90.53
BILSTM	bert	88.85	81.32	88.15

TABLE V
ACCURACY OF 10-FOLD CROSS-VALIDATION: ARSARCASM-V2 WITH BILSTM

K	BiLstm+Word2Vec		BiLstm+Fasttext		BiLstm+Glove		BiLstm+Bert	
	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed
1	70.68	71.32	68.42	67.85	66.17	68.04	68.10	68.94
2	73.18	74.79	72.15	73.50	76.78	76.59	78.97	76.59
3	70.93	72.60	72.09	73.12	77.43	85.40	76.85	78.97
4	74.28	77.81	73.89	77.17	81.54	93.12	81.80	87.91
5	71.64	78.52	72.41	79.16	79.87	95.88	81.54	90.29
6	73.76	81.16	73.38	81.80	82.44	96.40	82.32	93.76
7	75.56	84.63	74.98	87.07	85.27	97.56	85.53	95.05
8	75.05	86.17	73.05	87.91	86.11	97.17	84.24	96.53
9	73.23	86.36	74.39	88.87	86.10	97.17	86.87	97.04
10	74.26	89.06	74.39	91.63	86.94	98.01	86.94	96.46
AVG	73.26	80.24	72.92	80.81	80.87	90.53	81.32	88.15

B. Experiment 2: English-language Dataset

1) Goal: In English-language dataset, this experiment aims to find the best classifier and embedding technique for sarcasm detection and sentiment analysis tasks. It also investigates how sarcasm-detection features impact sentiment analysis models across different classifiers and embedding techniques.

TABLE VI
ACCURACY OF 10-FOLD CROSS-VALIDATION: ARSARCASM-V2 WITH LSTM

K	Lstm+Word2Vec		Lstm+Fasttext		Lstm+Glove		Lstm+Bert	
	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed
1	70.48	69.97	69.45	70.10	67.07	67.91	68.94	69.00
2	73.44	73.89	73.44	74.21	80.51	74.66	75.69	75.24
3	69.52	71.06	70.42	69.84	81.09	80.45	74.02	73.63
4	72.99	74.08	71.90	74.73	83.34	85.66	77.17	80.84
5	71.38	74.08	70.93	73.70	82.64	87.52	77.17	82.83
6	71.45	75.37	71.90	76.78	84.95	89.26	76.46	86.50
7	74.86	78.97	73.05	80.96	85.92	92.15	80.39	90.10
8	73.12	78.78	71.32	78.84	84.69	93.57	78.78	90.35
9	71.43	79.60	72.72	81.79	86.62	93.82	82.05	91.25
10	72.07	80.63	73.29	82.18	85.71	94.59	80.89	91.70
AVG	72.07	75.64	71.84	76.31	82.26	85.96	77.16	83.14

2) Setup: In this experiment, the IsarcasmEval dataset is utilized for sarcasm detection, while the IMDB Movie Reviews dataset is used for sentiment analysis. Each entry in the sarcasm dataset is labeled with a sarcasm label, and each entry in the sentiment dataset is labeled with a sentiment category.

3) Results: Figure 3 and Table VII present notable variations in the accuracy of sarcasm detection models, depending on the feature extraction and classifier algorithm. The highest accuracy, reaching 89.65%, is achieved using the glove feature extraction method and BILSTM classifier for the IsarcasmEval dataset. In terms of sentiment analysis, the models without incorporating sarcasm detection features exhibit accuracies ranging from 87.09% to 94.83%. The incorporation of sarcasm detection features into sentiment analysis models yields an improvement in accuracy, as evidenced by the highest values ranging from 90.90% to 96.63%. This indicates that leveraging sarcasm detection features enhances the overall effectiveness of sentiment analysis models, suggesting a positive effect in discerning sentiment nuances in text, especially when sarcasm is considered. The BILSTM classifier with glove feature extraction stands out with the highest combined accuracy of 96.63% for IMDB Movie Reviews dataset, emphasizing the effectiveness of this particular configuration in joint sarcasm detection and sentiment analysis tasks. Table VIII and Table IX present the accuracy results of k-fold cross-validation for sentiment analysis (SA) and joint sarcasm detection with sentiment analysis (Joint SD and SA) using different classifiers (BILSTM and LSTM) with various word embeddings (Word2vec, Fasttext, Glove, Bert) on the IMDB dataset for sentiment model and IsarcasmEval Dataset for sarcasm model. The inclusion of sarcasm features in the sentiment analysis models has a notable impact on performance. In general, the accuracy improvement ranges from approximately 2% to over 10%, highlighting the substantial positive impact of incorporating sarcasm-related information. The joint models tend to slightly underperform compared to the sentiment-only models for lower values of k (number of folds) but demonstrate significant improvement as k increases. This suggests that the joint models benefit from more extensive cross-validation, likely due to the nuanced nature of sarcasm detection. The BILSTM model demonstrates an average accuracy ranging from 92.89% to 96.63%, while the LSTM model shows a range of 90.90% to 94.71%. The average accuracy across all settings indicates that joint models consistently outperform the sentiment-only model, as sarcasm can alter sentiment polarity and significantly affect overall understanding.

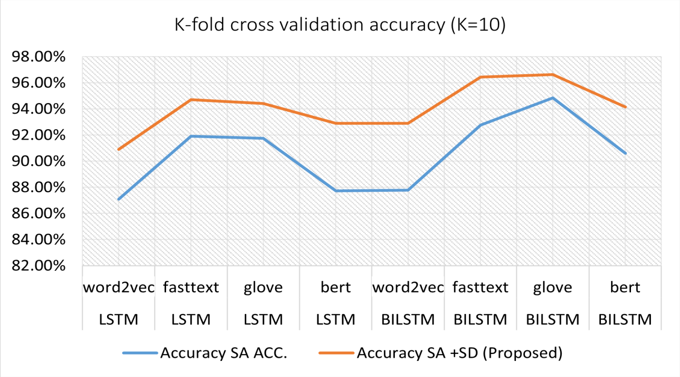

Fig. 3. K-fold cross-validation accuracy (k=10) on IMDB Movie Reviews dataset and IsarcasmEval Dataset

TABLE VII
ACCURACY OF 10-FOLD CROSS-VALIDATION: IMDB, AND ISARCASMEVAL DATASETS

Classifier	Feature Extractor	Accuracy (%)		
		SD	SA	SA+SD (Proposed)
LSTM	Word2Vec	79.10	87.09	90.90
LSTM	fasttext	85.21	91.90	94.71
LSTM	glove	87.00	91.76	94.40
LSTM	bert	79.41	87.72	92.90
BILSTM	Word2Vec	80.44	87.78	92.89
BILSTM	fasttext	82.83	92.77	96.44
BILSTM	glove	88.52	94.83	96.63
BILSTM	bert	87.10	90.61	94.14

TABLE VIII
CCURACY OF 10-FOLD CROSS-VALIDATION: IMDB, AND ISARCASMEVAL DATASETS WITH BILSTM

K	BiLstm+Word2Vec		BiLstm+Fasttext		BiLstm+Glove		BiLstm+Bert	
	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed
1	83.18	83.10	84.30	83.56	83.54	83.32	81.76	81.84
2	87.50	86.86	91.14	90.84	93.92	91.78	89.44	88.42
3	87.38	89.92	92.04	94.88	94.58	95.32	90.18	89.90
4	87.64	92.04	92.52	97.48	95.38	98.46	90.48	94.62
5	87.54	93.48	93.22	98.82	95.72	98.74	91.12	95.82
6	88.88	94.78	93.94	99.64	96.10	99.64	91.84	97.60
7	87.80	95.54	93.54	99.66	96.04	99.66	91.44	97.40
8	89.00	96.72	95.28	99.76	97.38	99.76	92.58	98.16
9	89.40	98.12	95.60	99.90	97.50	99.68	93.48	98.28
10	89.48	98.36	96.12	99.86	98.14	99.92	93.82	99.40
AVG	87.78	92.89	92.77	96.44	94.83	96.63	90.61	94.14

C. Experiment 3: Code-Mixed Arabic-English Dataset

1) Goal: In this experiment utilizing an Arabic-English code-mixed dataset, the primary objective is to identify optimal methods for detecting sarcasm in sentiment analysis. It investigates different classifiers and embeddings, taking into account how sarcasm-detection features impact sentiment analysis models

2) Setup: In this experiment, the IsarcasmEval dataset is utilized for sarcasm detection, while the SentiMixArEn dataset is used for sentiment analysis. Each entry in the sarcasm dataset is labeled with a sarcasm label, and each entry in the sentiment dataset is labeled with a sentiment category.

3) Results: The results presented in Figure 4 and Table X illustrate that integrating sarcasm detection task in sentiment analysis task (SA+SD) significantly improves accuracy compared to sentiment analysis alone (SA). For instance, using LSTM with the fasttext feature extractor, accuracy increases from 96.43% (SA) to 98.51% (SA+SD). Similarly, when considering the LSTM classifier with word2vec (cbow) embeddings, the sentiment analysis model achieves an accuracy of 93.25%. However, when sarcasm features are integrated into the model, the accuracy substantially increases to 95.54%. Additionally, with glove, and bert embeddings, the addition of sarcasm features leads to even more substantial improvements, reaching 97.72%, and 92.97% respectively. Overall, models that include sarcasm detection consistently outperform those relying solely on sentiment analysis, demonstrating the importance of accounting for sarcasm in text classification. it is evident that the utilization of fasttext embeddings consistently yields the highest accuracy, achieving 96.43% for sentiment analysis, and 98.51% for sentiment analysis incorporating sarcasm features. Additionally, the results indicate that Bi-LSTM with GloVe embeddings achieves the second-highest accuracy in this scenario, scoring 97.72% for sentiment analysis alone and 98.02% when incorporating sarcasm. Table XI and Table XII present the detailed accuracy results of k-fold cross-validation for sentiment analysis (SA) and joint sarcasm detection and sentiment analysis (Joint SD and SA) models, using different embeddings and classifiers (BILSTM or LSTM) across various values of k (1 to 10). The addition of sarcasm detection features (Joint SD and SA) to sentiment analysis models shows a consistent improvement in accuracy across different embeddings and classifiers. In both BILSTM and LSTM architectures, the Joint SD and SA models consistently outperform their SA counterparts, indicating that incorporating sarcasm detection features enhances the overall effectiveness of sentiment analysis models. The improvement is more pronounced in certain scenarios, such as BILSTM with Fasttext embeddings. The average accuracy for Joint SD and SA models is consistently higher than that of SA models alone, demonstrating the positive impact of sarcasm features on sentiment analysis tasks.

TABLE IX
ACCURACY OF 10-FOLD CROSS-VALIDATION: IMDB, AND ISARCASMEVAL DATASETS WITH LSTM

K	Lstm+Word2Vec		Lstm+Fasttext		Lstm+Glove		Lstm+Bert	
	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed
1	83.52	82.70	84.24	83.98	83.68	84.06	82.02	81.82
2	87.12	86.40	91.20	90.22	90.90	88.52	87.06	88.58
3	86.96	88.80	91.76	92.06	91.06	91.88	86.86	90.92
4	87.58	89.94	92.14	94.52	92.02	94.00	87.66	92.40
5	87.08	90.70	92.94	96.00	92.38	95.60	87.44	93.30
6	87.90	92.52	92.66	97.14	92.92	96.52	88.94	95.08
7	86.80	92.42	92.30	96.80	92.22	97.76	87.90	95.08
8	87.76	95.06	94.00	98.46	94.00	98.38	89.32	96.84
9	87.96	94.98	93.28	98.92	94.00	98.64	89.96	97.26
10	88.24	95.52	94.52	98.98	94.38	98.68	90.04	97.68
AVG	87.09	90.90	91.90	94.71	91.76	94.40	87.72	92.90

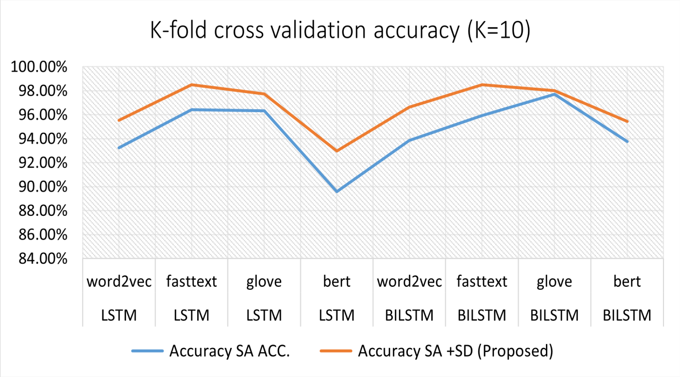

Fig. 4. K-fold cross-validation accuracy (k=10) on SentiMixArEn and IsarcasmEval Datasets

TABLE X
ACCURACY OF 10-FOLD CROSS-VALIDATION: SENTIMIXAREN, AND ISARCASMEVAL DATASETS.

Classifier	Feature Extractor	Accuracy (%)		
		SD	SA	SA+SD (Proposed)
LSTM	Word2Vec	80.03	93.25	95.54
LSTM	fasttext	82.54	96.43	98.51
LSTM	glove	87.04	96.33	97.72
LSTM	bert	79.54	89.58	92.97
BILSTM	Word2Vec	79.93	93.85	96.63
BILSTM	fasttext	84.00	95.93	98.51
BILSTM	glove	89.24	97.72	98.02
BILSTM	bert	84.10	93.76	95.44

VI. DISCUSSION

In our experiments, we utilized pre-trained models for word embedding, as detailed in Table II. This approach, known as transfer learning, allowed us to leverage existing knowledge and resources rather than training models from scratch. By harnessing pre-trained models, we accelerated model training and enhanced performance, demonstrating the efficiency and effectiveness of transfer learning in our deep learning framework. Our experiments revealed that bidirectional LSTM models indeed exhibit longer processing times compared to unidirectional LSTM models. This observation aligns with existing literature, which commonly reports that bidirectional models require more computational resources due to their ability to capture context in both the forward and reverse directions. Despite the increased processing time, the BiLSTM model showcased enhanced accuracy in our experiments, as evidenced in Table III, Table VI, and Table IX. This can be attributed to its ability to capture richer contextual information by considering both past and future states of the input sequence. Using BERT embeddings adds computational overhead and increases processing time, but it improves performance by capturing detailed semantic features, enhancing both sarcasm detection and sentiment analysis. Moreover, our proposed system involves a joint task of sarcasm detection followed by sentiment analysis. The inclusion of the sarcasm detection phase adds an extra layer of complexity to the overall processing pipeline. Consequently, the proposed model's execution time is further prolonged compared to models solely dedicated to sentiment analysis. However, our experiments demonstrate that the joint model achieves superior accuracy

TABLE XI
ACCURACY OF 10-FOLD CROSS-VALIDATION: SENTIMIXAREN, AND ISARCASMEVAL DATASETS WITH BILSTM

K	BiLstm+Word2Vec		BiLstm+Fasttext		BiLstm+Glove		BiLstm+Bert	
	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed
1	83.17	86.14	87.13	87.13	88.12	84.16	73.27	72.28
2	97.03	95.05	95.05	99.01	99.01	96.04	95.05	93.07
3	96.04	97.03	95.05	99.01	98.02	100.00	95.05	97.03
4	88.12	94.06	95.05	100.00	98.02	100.00	95.05	98.02
5	91.09	99.01	98.02	100.00	98.02	100.00	97.03	98.02
6	97.03	97.03	96.04	100.00	97.03	100.00	94.06	99.01
7	97.03	99.01	99.01	100.00	100.00	100.00	94.06	98.02
8	99.00	100.00	99.00	100.00	100.00	100.00	99.00	100.00
9	96.00	99.00	96.00	100.00	99.00	100.00	98.00	100.00
10	94.00	100.00	99.00	100.00	100.00	100.00	97.00	99.00
AVG	93.85	96.63	95.93	98.51	97.72	98.02	93.76	95.44

TABLE XII
ACCURACY OF 10-FOLD CROSS-VALIDATION: SENTIMIXAREN, AND ISARCASMEVAL DATASETS WITH LSTM

K	Lstm+Word2Vec		Lstm+Fasttext		Lstm+Glove		Lstm+Bert	
	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed	SA	SA+SD proposed
1	85.15	86.14	86.14	88.12	85.15	83.17	75.25	67.33
2	97.03	96.04	96.04	98.02	97.03	97.03	93.07	95.05
3	95.05	94.06	98.02	100.00	98.02	99.01	89.11	88.12
4	90.10	94.06	94.06	99.01	97.03	99.01	93.07	93.07
5	94.06	96.04	98.02	100.00	97.03	100.00	82.18	94.06
6	94.06	97.03	96.04	100.00	96.04	100.00	95.05	99.01
7	96.04	98.02	100.00	100.00	100.00	100.00	92.08	96.04
8	95.00	100.00	99.00	100.00	98.00	100.00	89.00	99.00
9	96.00	98.00	97.00	100.00	97.00	99.00	95.00	99.00
10	90.00	96.00	100.00	100.00	98.00	100.00	92.00	99.00
AVG	93.25	95.54	96.43	98.51	96.33	97.72	89.58	92.97

compared to models focusing solely on sentiment analysis. This emphasizes the importance of incorporating contextual cues, such as sarcasm, into sentiment analysis frameworks to enhance their performance and robustness. In our experiment, for validation, we utilized k-fold cross-validation with k=10. The average accuracy of k-fold cross-validation increases significantly when k exceeds 4. For k values up to 4, both sentiment-only and combined sentiment with sarcasm detection models perform equally well in terms of accuracy. However, beyond k=4, there's a noticeable improvement in the average accuracy of k-fold cross-validation, particularly in the combined model that incorporates both sentiment and sarcasm detection . In our experiment, adding sarcasm detection to a simple Bi-LSTM or LSTM model significantly improved sentiment analysis accuracy. Across various datasets, including Arabic, English, and code-mixed texts, the model with sarcasm features outperformed traditional sentiment analysis models by 2% to over 10% in accuracy. This shows that even with a simple model structure, integrating sarcasm detection can notably enhance performance in understanding nuanced language

VII. COMPARISON WITH RECENT STUDIES

Our model combines a simple Bi-LSTM or LSTM structure with integrated sarcasm detection, achieving high accuracy, particularly on datasets with sarcastic text. Focusing specifically on sarcasm outperforms more complex models and enhances the effectiveness of sentiment analysis. In comparison to recent studies on the ArSarcasm-v2 dataset as presented in Table XIII , the various models reviewed use a range of techniques, but most focus heavily on the model architecture rather than the integration of sarcasm features. For instance, Song [23] and Wadhawan [25] achieved similar sarcasm detection accuracies (78.3%) by using transformer-based models like BERT and AraBERT, fine-tuned with different strategies, but their sentiment analysis results were lower (70.37% and 69.83%, respectively). Mahdaouy [12] employed a multi-task learning (MTL) model with a BERT encoder and multi-task attention module, slightly improving sentiment accuracy (71.07%). Abdel-Salam [26] used a hybrid of LSTM-CNN-GRU with MARBERT, attaining modest sarcasm detection (78.03%) and sentiment (69.57%) results. In contrast, the proposed model emphasizes sarcasm integration across architectures, leveraging Bi-LSTM with embeddings like GloVe and Word2Vec, leading to significantly higher performance in both sarcasm detection (94.55%) and sentiment analysis (90.53%). For comparison, as presented in Table XIV, the proposed Bi-LSTM model with GloVe embeddings, achieving an accuracy of 96.63% on the IMDB Movie Reviews dataset, demonstrates a significant edge over recent state-of-the-art models. The proposed model, achieving an accuracy of 96.63%, excels by specifically addressing sarcasm in sentiment analysis, an aspect often overlooked by recent studies. Unlike Yang [33]'s XLNet (96.21%), which improves bidirectional context learning, our model enhances sentiment interpretation through dedicated sarcasm detection. Heinsen [34]'s routing algorithm (96.2%) and Wang [35]'s EFL (96.1%) focus on computational efficiency and few-shot learning, respectively, without addressing sarcasm. Lu [36]'s GraphStar (96%) advances graph-based neural networks but lacks sarcasm detection, and Zhang [37]'s study (95.94%) critiques simpler models without specifically tackling sarcasm. Our model's integration of sarcasm detection into a simple Bi-LSTM or LSTM structure achieves superior accuracy, particularly on datasets containing sarcastic text, setting it apart from more complex models lacking this specific focus.

TABLE XIII
ARSARCASM-V2 DATASET COMPARISON OF THE PROPOSED MODEL WITH RECENT STUDIES

Study	Sarcasm Accuracy	Sentiment Accuracy
Mahdaouy [12]	0.768	0.7107
Song [23]	0.783	0.7037
Alharbi [24]	0.770	0.6753
Wadhawan [25]	0.783	0.6983
Abdel-Salam [26]	0.7803	0.6957
Hengle [27]	0.741	0.6840
Husain [28]	0.7727	0.6073
Abuzayed [29]	0.698	0.6853
Elagbry [30]	0.7533	0.6053
Gaanoun [31]	0.7797	0.6817
Proposed system	**94.55**	**90.53**

VIII. CONCLUSION AND FUTURE WORK

In conclusion, the proposed model, featuring a simple Bi-LSTM or LSTM architecture with integrated sarcasm detection, demonstrates high accuracy, especially with sarcastic text. This method not only outperforms more complex models but

TABLE XIV
IMDB MOVIE REVIEWS DATASET COMPARISON OF THE PROPOSED MODEL WITH RECENT STUDIES.

Rank	Study	Sentiment Accuracy%
1	**Proposed system**	**96.63**
2	Yang [33]	96.21
3	Heinsen [34]	96.20
4	Wang [35]	96.10
5	Lu [36]	96.00
6	Zhang [37]	95.94

also significantly boosts overall sentiment analysis effectiveness. The incorporation of sarcasm features leads to substantial accuracy improvements, with embedding methods like Glove and BERT enhancing performance by over 7%. Both Bi-LSTM and LSTM classifiers benefit from sarcasm detection, showing accuracy gains of 2% to over 10%. Future work involves extending the model to support additional languages beyond Arabic and English by acquiring labeled datasets, adapting architecture for linguistic variations, and fine-tuning parameters. Techniques such as multilingual learning and transfer learning could enhance the model's applicability, facilitating sentiment analysis in diverse linguistic contexts and fostering cross-cultural communication. Additionally, improving the sarcasm detection accuracy integrated into sentiment analysis models could involve leveraging ensemble learning, adversarial training, and attention mechanisms.

REFERENCES

[1] D. Draskovic, D. Zecevic, and B. Nikolic, "Development of a Multilingual Model for Machine sentiment analysis in the Serbian Language," Mathematics, Vol. 10., pp. 3236, 2022 doi:10.3390/math10183236

[2] S. M. A. H. Shah, S. F. H. Shah, A. Ullah, A. Rizwan, G. Atteia, and M. Alabdulhafith, "Arabic sentiment analysis and Sarcasm Detection Using Probabilistic Projections-Based Variational Switch Transformer," IEEE Access, Vol. 11, pp. 1-1, 2023, doi:10.1109/ACCESS.2023.3289715.

[3] M. Graff, S. Miranda, E. Tellez, and D. Moctezuma, "EvoMSA: A Multilingual Evolutionary Approach for sentiment analysis," IEEE Computational Intelligence Magazine, Vol. 15, pp. 76-88, 2020, doi:10.48550/arXiv.1812.02307.

[4] P. Goel, V. Goel, and A. Gupta, "Multilingual Data Analysis to Classify sentiment analysis for Tweets Using NLP and Classification Algorithm," Vol. 10, pp. 12, 2020, doi:10.1007/978-981-15-0694-9_26.

[5] R. Zahedi, N. Alajlan, H. Zahedi, and T. Rabczuk,"Multilingual Sentiment Mining System to Prognosticate Governance," Computers, Materials & Continua, Vol. 71, pp. 389-406, 2022 doi:10.32604/cmc.2022.021384.

[6] K. Arun, and A. Srinagesh, "Multi-lingual Twitter sentiment analysis using machine learning," International Journal of Electrical and Computer Engineering (IJECE), Vol. 10, pp. 5992-6000, 2020,doi:10.11591/ijece.v10i6.pp5992-6000.

[7] C. Yin, Y. Chen, and W. Zuo,"Multi-Task Deep Neural Networks for Joint Sarcasm Detection and sentiment analysis," Pattern Recognition and Image Analysis, Vol. 31, pp. 103-108, 2021, doi:10.1134/S105466182101017X.

[8] R. Rao, S. Dayanand, K. Varshitha, and K. Kulkarni, "Sarcasm Detection for sentiment analysis: A RNN-Based Approach Using Machine Learning," High Performance Computing and Networking. Lecture Notes in Electrical Engineering, Singapore, vol. 853, 2022, doi:10.1007/978-981-16-9885-9_4.

[9] D. Londhe, and A. Kumari, "Multilingual sentiment analysis Using the Social Eagle-Based Bidirectional Long Short-Term Memory," International Journal of Intelligent Engineering and Systems, Vol. 15, pp.479-493, 2022, doi:10.22266/ijies2022.0430.43.

[10] S. Gupta, R. Singh, and V. Singla, "Emoticon and Text Sarcasm Detection in sentiment analysis," First International Conference on Sustainable Technologies for Computational Intelligence, Singapore, Vol. 1045, pp.1-10, 2020, doi:10.1007/978-981-15-0029-9_1.

[11] M. Shrivastava, and S. Kumar, "A Pragmatic and Intelligent Model for Sarcasm Detection in Social Media Text," Technology in Society, Vol. 64, pp. 101489, 2020, doi:10.1016/j.techsoc.2020.101489.

[12] A. Mahdaouy, A. e. Mekki, K. Essefar, N. Mamoun, I. Berrada, and A. Khoumsi, "Deep Multi-Task Model for Sarcasm Detection and sentiment analysis in Arabic Language," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 334–339, 2021.

[13] M. Naski, A. Messaoudi, H. Haddad, M. BenHajhmida, C. Fourati, and A. B. Mabrouk, "iCompass at Shared Task on Sarcasm and Sentiment Detection in Arabic," Workshop on Arabic Natural Language Processing, Kyiv, Ukraine (Virtual), pp. 381-385, 2021.

[14] A. Rahaman W. Sait, and M. K. Ishak, "Deep Learning with Natural Language Processing Enabled Sentimental Analysis on Sarcasm Classification," Computer Systems Science and Engineering, Vol. 44, pp. 2553-2567,2023, doi:10.32604/csse.2023.029603.

[15] M. M. Krishna, M. Janarthanan , and J. Vankara, "Detection of Sarcasm Using Bi-Directional RNN Based Deep Learning Model in sentiment analysis," Journal of Advanced Research in Applied Sciences and Engineering Technology, Vol. 31, pp. 352-362,2023, doi:10.37934/araset.31.2.352362.

[16] B. Ghanem, J. Karoui, F. Benamara, P. Rosso, and V. Moriceau, "Irony Detection in a Multilingual Context," Advances in Information Retrieval, Cham, Vol. 12036, pp. 141-149, 2020, doi:10.48550/arXiv.2002.02427.

[17] A. Cignarella, V. Basile, M. Sanguinetti, C. Bosco, P. Rosso, and F. Benamara, "Multilingual Irony Detection with Dependency Syntax and Neural Models," Proceedings of the 28th International Conference on Computational Linguistics, pp. 1346-1358, 2020, doi:10.18653/v1/2020.coling-main.116.

[18] Y. Han, Y. Chai, S. Wang, Y. Sun, H. Huang, G. Chen, Y. Xu, and Y. Yang, "X-PuDu at SemEval-2022 Task 6: Multilingual Learning for English and Arabic Sarcasm Detection," Proceedings of the 16th International Workshop on Semantic Evaluation, pp. 999-1004, 2022, doi:10.18653/v1/2022.semeval-1.140.

[19] I. A. Farha, and W. Magdy, "From Arabic sentiment analysis to Sarcasm Detection: The ArSarcasm Dataset," The 4th Workshop on Open-Source Arabic Corpora and Processing ToolsAt, Marseille, pp. 32-39, 2020.

[20] I. A. Farha, S. V. Oprea, S. Wilson, and W. Magdy, "SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic," Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)At: Seattle, United States, pp. 802–814,2022, doi:10.18653/v1/2022.semeval-1.111.

[21] A. L. Maas, R. E. Daly, P. T. Pham, D.Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for sentiment analysis," Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologie, Portland, Oregon, USA, pp. 142-150, 2011.

[22] M. N. Raihan, D. Goswami, A. Mahmud, A. Anastasopoulos, and M. Zampieri, "SentMix-3L: A Bangla-English-Hindi Code-Mixed Dataset for sentiment analysis," Proceedings of the First Workshop in South East Asian Language Processing, pp 79–84, 2023, doi:10.48550/arXiv.2310.18023.

[23] B. Song, C. Pan, S. Wang, and Z. Luo, "A Deep Ensemble-based Method for Sarcasm and Sentiment Detection in Arabic," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 390-394, 2021.

[24] 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," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 318-322, 2021.

[25] A. Wadhawan, "Arabert and farasa segmentation based approach for sarcasm and sentiment detection in arabic tweets," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 395-400, 2021.

[26] R. Abdel-Salam, "Wanlp 2021 shared-task: Towards irony and sentiment detection in arabic tweets using multi-headed-lstm-cnn-gru and marbert," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 306-311, 2021.

[27] A. Hengle, A. Kshirsagar, S. Desai, and M. Marathe, "Combining context-free and contextualized word representations for arabic sarcasm detection and sentiment identification," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 357-363, 2021.

[28] F. Husain and O. Uzuner, "Leveraging offensive language for sarcasm and sentiment detection in arabic," Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 364-369, 2021.

[29] A. Abuzayed and H. Al-Khalifa, ”Sarcasm and sentiment detection in arabic tweets using bert-based models and data augmentation,” Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 312-317, 2021.

[30] H. E. Elagbry, S. Attia, A. Abdel-Rahman, A. Abdel-Ate, and S. Girgis, ”A contextual word embedding for arabic sarcasm detection with random forests,” Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 340-344, 2021.

[31] K. Gaanoun and I. Benelallam, ”Sarcasm and sentiment detection in arabic language: A hybrid approach combining embeddings and rule-based features,” Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), pp. 351-356, 2021.

[32] D. Draskovic, D. Zecevic, and B. Nikolic, ”LlamBERT: Large-scale low-cost data annotation in NLP,” Mathematics, Vol. 10, pp.3236, 2022, doi:10.48550/arXiv.2403.15938

[33] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, ”XLNet: Generalized Autoregressive Pretraining for Language Understanding,” Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS), No. 517, pp. 5753-5763, December 2019, doi:10.48550/arXiv.1906.08237

[34] F. A. Heinsen, ”An Algorithm for Routing Vectors in Sequences,” ArXiv, Vol. abs/2211.11754, 2022, doi:10.48550/arXiv.2211.11754

[35] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, ”Entailment as Few-Shot Learner.” ArXiv, Vol. abs/2104.14690, 2021, doi:10.48550/arXiv.2104.14690

[36] H. Lu, S. H. Huang, T. Ye, and X. Guo, ”Graph Star Net for Generalized Multi-Task Learning,” ArXiv, Vol. abs/1906.12330, 2019 , doi:10.48550/arXiv.1906.12330

[37] Bingyu Zhang, Nikolay Arefyev, ”The Document Vectors Using Cosine Similarity Revisited,” ArXiv, Vol. abs/2205.13357, 2022, doi:10.48550/arXiv.2205.13357

Ahmed Derbala Yacoub is currently pursuing a master’s degree in the Computer Science department while working as a technical manager at a software development company. He holds a bachelor’s degree in Computer Science from the Faculty of Computer and Artificial Intelligence at Helwan University in Cairo, Egypt. Passionate about software development, he is dedicated to advancing his knowledge and skills in computer science, aiming to contribute to innovative projects and solutions in the tech industry.

Amal Elsayed Aboutabl is currently a Professor of Computer Science at the Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt. She received her B.Sc. in Computer Science from the American University in Cairo and both of her M.Sc. and Ph.D. in Computer Science from Cairo University. She worked for IBM and ICL in Egypt for seven years. She was also a Fulbright Scholar at the Department of Computer Science, University of Virginia, USA. Her research interests include software engineering and natural language processing.

Salwa O. Slim is an accomplished academic and researcher with a Doctor of Philosophy in Computer Science from Helwan University, Egypt. Specializing in data analytics, machine learning, deep learning, and IoT systems, she developed a cutting-edge real-time system for human activity recognition via IoT in their Ph.D. thesis. This system utilized advanced Convolutional Neural Networks (CNNs) and was optimized using genetic algorithms, achieving high performance in activity detection. She is also a highly regarded educator with extensive experience teaching at various institutions, including Helwan University, Ain Shams University, MIU University, AOU University, and BST University. They have taught a broad range of subjects, including machine learning, artificial intelligence, algorithms, software engineering, data structures, and IoT systems. As a lecturer and coordinator for the Data Science Program at Helwan National University, She has played a key role in curriculum development and project supervision for undergraduate and postgraduate students.