Soft Computing
https://doi.org/10.1007/s00500-023-08368-6

FOCUS

Identifying sarcasm using heterogeneous word embeddings: a hybrid and ensemble perspective

Ravi Teja Gedela[1] · Pavani Meesala[1] · Ujwala Baruah[1] · Badal Soni[1]

Accepted: 29 April 2023

Abstract
Past research suggests pre-trained word embedding strategies to assess and determine feelings conveyed in various text documents. However, using a single word embedding strategy makes it difficult to grasp the whole spectrum of intricate inter-dependencies among words in texts. This article presents hybrid and stacking-based ensemble approaches for sarcasm detection using heterogeneous word embeddings to tackle this issue. The proposed approaches accomplish the sarcasm recognition task by combining three heterogeneous word embedding techniques created by state-of-the-art strategies: Global Vectors for Word Representation (GloVe), fastText, and Bidirectional Encoder Representations from Transformers (BERT). The pre-trained BERT-base model produces the first pair of 768-dimensional word vectors. GloVe and fastText models generate two pairs of word vectors, each with 300-dimensions. The proposed models have been evaluated on two corpora from different domains, namely news headlines and the self-annotated reddit corpus (SARC), both of which consist of English language texts. The former contains 28,619 news headlines from two notable websites, and the latter contains 533 million Reddit comments. The results obtained indicate the effectiveness of using heterogeneous word embeddings with the aid of BiLSTM-CNN for the sarcasm detection task. Experiments show that our proposed hybrid model achieves an accuracy of 95.7% on news headlines, which is an improvement of 3.8% over the state-of-the-art approaches and 80.64% on the SARC. Also, the proposed stacking ensemble-based model achieves 96.76% on the news headlines, which is an improvement of 4.86%, and 81.46% on the SARC, which is a gain of 0.46% over other state-of-the-art approaches.

Keywords Sarcasm detection · Word embedding · Hybrid approach · Stacking-based ensemble approach

1 Introduction

Social media has risen to prominence as the most popular platform for voicing one's thoughts, feelings, and facts. Due to this, social media platforms like Twitter, Facebook, Instagram, and others generate tremendous volumes of data daily (Srinivasarao and Sharaff 2020). Many entrepreneurs use this information to understand and analyze public perception about a specific individual, product, idea, or entity. As an outcome, sentiment analysis of social media content has sparked a lot of interest. Sentiment analysis identifies the emotional feelings conveyed in textual data, including online conversation forums, product reviews, social media posts, etc. (Jindal and Aron 2021). Organizations employ tools that mine social media information to retrieve people's views and analyze the demand for goods and services. Stock trading enterprises also use emotion analysis strategies to acquire information and assess their impression of various news essays (Zhao et al. 2020). Though sentiment analysis has had fantastic success in a wide variety of fields, a few aspects still need to be investigated further, one of which is sarcasm identification. Finding sarcasm is tricky, requiring a thorough understanding of the language, dialogue system, and skills such as understanding context and content (Kumaran and Chitrakala 2022). Not only that, but correctly verifying the presence of sarcasm in a sentence is difficult for human beings too. Consequently, training a machine to differentiate between non-sarcastic and sarcastic comments is tricky and an emerging research challenge (Joshi et al. 2017).

The presence of sarcasm is felt when encouraging words and feelings in tweets have slang, unfavorable, or undesirable interpretations (Sarsam et al. 2020). Take the following line as an example: "It smells good. How long did you leave

✉ Ravi Teja Gedela
ravi_rs@cse.nits.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India

Published online: 22 May 2023

it to marinate?". Most people can infer a negative emotion from this sentence with a basic knowledge of sarcasm. We all recognize that the preceding phrase means that you have put on too much fragrance and is, therefore, not acceptable. However, machines struggle to identify the figurative nature of sarcasm in the text due to the presence of the positive phrase "good." Therefore, if not appropriately handled, sarcasm can alter the polarity of a statement from negative to positive (Liebrecht et al. 2013).

Three methods can be used to illustrate sarcasm detection (Yaghoobian et al. 2021): (a) a machine learning (ML)-based classification approach that leverages learning models; (b) a rule-based system that utilizes corpus-based sentiment lexicons, lexical dictionaries, or publicly accessible sentiment lexicons; and (c) a hybrid strategy that blends machine learning and rule-based systems. Consequently, deep learning (DL) approaches have proven their significance in text classification (Ghayoomi and Mousavian 2022), speech recognition (Nassif et al. 2019), and computer vision (He et al. 2022) as part of various studies. It has been observed that DL algorithms outperform conventional machine learning techniques when it comes to detecting sarcasm. Furthermore, integrating CNN and BiLSTM has been seen to yield more reliable and effective results in detecting sarcasm. BiLSTM is effective for gathering long-term dependencies, while CNN is excellent for retrieving local characteristics (Ay Karakuş et al. 2018).

Figure 1 depicts the general architectural framework of the sarcasm identification task. Identifying sarcasm begins with data preprocessing, which entails tokenizing the data points into individual phrases (tokens). The word tokens in each data point are then uniquely mapped to their indexes in the corpus's vocabulary database. This mapping is accomplished by assigning each token to an integer. Finally, the embedding layer translates the tokens with integer encoding into feature vectors of real values with fixed dimensions. Several classifier models then processes these real-valued feature vectors to classify the raw documents into sarcastic or non-sarcastic.

For many NLP applications, textual representation is essential. For this, researchers have been seen to use word embedding strategies to convert raw textual data into numeric word vectors that various ML-based sarcasm identification frameworks can process. Word embedding generates dense feature vectors with the appropriate dimensions, which can preserve the contextual and semantic relations between words in text documents. Four extensively used word embedding strategies are Word2Vec (Mikolov et al. 2013), fastText (Bojanowski et al. 2017), GloVe (Pennington et al. 2014), and BERT (Devlin et al. 2018). The first three word embedding schemes are static, whereas the fourth is contextual. Previous research has focused solely on static or contextual embeddings, with just one embedding approach for transforming raw text input into numeric vectors. Almost no research has been reported on developing a hybrid deep learning model that incorporates multiple kinds of embeddings (a combination of static and contextual embeddings).

An ensemble of models is a collection of learning models merged in fruitful ways to yield a more evident outcome (Rahman and Verma 2013). Ensemble learning approaches aimed at developing more generalizable models are also gaining attraction in sarcasm recognition. Furthermore, much of the ensemble learning work focuses on homogeneous ensembles. Only a few studies have been reported on heterogeneous ensembles (merging diverse models). Still, almost all of these use different deep learning models (Base CNN, RNN variants) with the same or varied embeddings. However, none of the work focuses on designing an efficient deep-learning model and applying the same to varied word embeddings (including static and contextual embeddings).

The following are the primary contributions of the proposed investigation:

1. We propose a novel hybrid deep learning strategy by concatenating the features from heterogeneous word representations (including static and contextual) with the aid of the BiLSTM-CNN.
2. We propose a novel stacking-based ensemble strategy with heterogeneous word embeddings (including static and contextual) utilizing the BiLSTM-CNN.
3. Our proposed models enhance sarcasm recognition and generate improved insights after training and evaluation on two publicly available datasets.

The remaining portion of the paper is presented as follows. We commence with a review of the literature in Sect. 2, followed by Sect. 3, which presents the framework of the proposed fusion and stacking ensemble approaches. Section 4 includes the experimental setup, results, and comparison with earlier research. Finally, Sect. 5 wraps up the paper and suggests future research.

2 Related work

This section examines current state-of-the-art methodologies pertinent to the proposed work, primarily focusing on sarcasm detection using hybrid and ensemble approaches. The literature review is discussed in four subheadings: the first and second subheadings concentrate on existing sarcasm recognition strategies that use static and contextual word embeddings, respectively. The third and fourth subheadings focus on existing hybrid-based and ensemble-based text classification works.

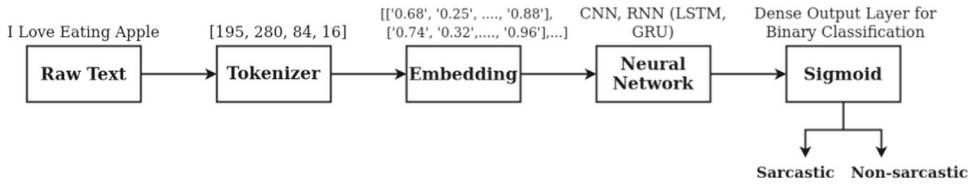

Fig. 1 General Flow of Sarcasm Identification Framework

2.1 Sarcasm detection with static word embeddings

Azwar et al. (2020) use a six-layer MCAB-BLSTM (BiLSTM based on multi-channel attention) network to recognize sarcasm in the news headlines. Two BiLSTM networks based on attention run concurrently—one utilizing GloVe word embeddings, while the other utilizes fastText. The reported accuracy is 96.64%. Kumar et al. (2019) present the sAtt-BLSTM convNet, an eight-layer model hybridizing sAtt-BLSTM (BiLSTM plus soft attention) and convNet (CNN). Their proposed method uses GloVe to produce meaningful word embeddings. The experimentation has been carried out with two datasets, and produces accuracies of 97.87% on Twitter's balanced corpus and 93.71% on an unbalanced random-tweet corpus.

Misra and Arora (2019) bring a novel dataset containing news headlines to confront Twitter data's failings. This study presents an attention-based hybrid neural architecture that uses Word2vec embeddings as input and improves accuracy by approximately 5% over the baseline. On the SARC dataset, Mehndiratta and Soni (2019) investigated and presented the behaviors of several hyper-parameters, including epochs, data size, and dropout for each approach (CNN, LSTM, and CNN/LSTM blend). This study also examines the influence of word embeddings (fastText and GloVe).

2.2 Sarcasm detection with contextual word embeddings

Bhardwaj and Prusty (2022) present a novel strategy in which BERT is used to preprocess the sentences before it is fed into a blended deep-learning model for training and classification purposes. They got 99.63% accuracy with tenfold cross-validation on the news headlines dataset. The LMTweets encoder model, introduced by Ahuja and Sharma (2022), is used to capture the dataset's features after training on 500000 tweets crawled from various social media sites. The CNN model uses the retrieved features to identify the sentence as ironic/non-ironic and sarcastic/non-sarcastic. Six transformer models, six deep learning models, and five machine learning techniques were used in the study. According to the data, the LMTweets + CNN model outperforms all other models tested with accuracies of 88.3% on SemEval 2018 Task 3.A corpus, 95.9% on Riloff corpus, and 80.9% on SARC(political) corpus.

Shrivastava and Kumar (2021) suggest an innovative strategy based on BERT to represent the text and assess if it is sarcastic or otherwise. In this study, several hyperparameters were examined, and the F1-score of the proposed model is 69.64%, which is compared against a set of benchmarks. Potamias et al. (2020) presents RCNN-RoBERTa, a hybrid of Recurrent CNN and RoBERTa evaluated on four standard repositories. Furthermore, this model surpasses all other state-of-the-art strategies examined, including XLnet, BERT, USE, and ELMo, on all criteria, some by a considerable margin.

2.3 Hybrid-based text classification works

Pandey and Singh (2023) proposes a hybrid model consisting of BERT stacked with LSTM. For a code-mixed English-Hindi dataset, the authors used BERT to create embeddings, and then an LSTM network used the resulting vectors. The proposed model achieved an accuracy of 92%, an improvement of nearly 6% over the other baseline methods. Eke et al. (2021) use three strategies on three benchmark repositories to present a context-based feature solution for sarcasm detection. BiLSTM on GloVe embeddings is used in the first strategy, whereas BERT is used in the second. The third model, on the other side, employs a feature fusion strategy that combines BERT, GloVe embedding features, sentiment-related, and syntactic with traditional machine learning. To design a hybrid model for discovering rumors in a given tweet, Albahar (2021) combine SVM and RNN with BiGRU. RNN with BiGRU is utilized in the first stage of the proposed hybrid approach for learning features, and SVM is used in the second step for classification.

To handle mixed inputs, Yuan et al. (2020) designed a generic framework of hybrid deep neural networks (HDNNs), which is an aggregation of multiple networks (CNN and MLP), indicating its versatility and adaptability. The suggested HDNN model outperforms the MLP and CNN models in terms of accuracy and generalization. Using FastText and character-level embeddings with CNN and LSTM algorithms, Salur and Aydin (2020) introduce an innovative hybrid strategy for opinion analysis. This article utilizes a dataset of 17,289 Turkish tweets and gives an accuracy of 82.14%.

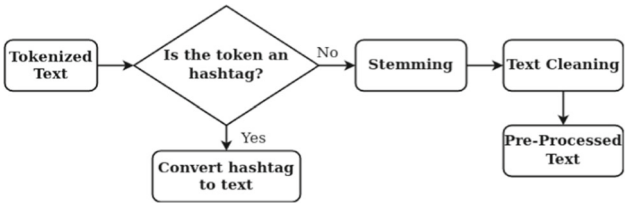

Fig. 2 Preprocessing Pipeline

2.4 Ensemble-based text classification works

Praseed et al. (2023) presents an ensemble approach for the identification of fake news in Hindi, comprised of three transformer models, including XLM-RoBERTa, ELECTRA, and mBERT. The model suggested enhanced efficacy by overcoming the shortcomings of the individual transformer models. Goel et al. (2022) suggest an ensemble model that employs baseline as CNN, BiLSTM, and GRU. They obtained word embeddings for the model using Word2vec, GloVe, and fastText. They used two standard repositories to train and validate their model. They examined three ensemble models and reported that the (Glove + weighted average ensemble) performed well on the datasets tested. Subba and Kumari (2022) propose a computationally efficient stacking ensemble-based sentiment classification strategy that uses several word embeddings (Glove, BERT, and Word2vec), multiple deep base-level classifiers (LSTM, GRU, and BiGRU), and an LSTM-based meta-level classifier. They tested their approach on four standard corpora and claimed that it outperformed the strategies reported in the literature.

Gundapu and Mamidi (2021) provide an ensemble method for detecting fake news that combines three transformer architectures (BERT, ALBERT, and XLNET). This model achieved an F1-score of 0.9855 after being trained and assessed in the ConstraintAI 2021 shared task (Patwa et al. 2021). To identify idioms and literals on the in-house dataset of 1470 data points, Briskilal and Subalalitha (2022) recommends an ensemble model using BERT and RoBERTa models. The suggested model has 2% greater accuracy than the benchmarks.

It has been observed that several hybrid and ensemble approaches have been developed for the sarcasm detection task. Most of these approaches made use of a single-word embedding strategy for converting raw text into numerical vectors. However, none of the work focuses on developing an efficient hybrid model with multiple word embeddings. Also, only a few works have concentrated on stacking ensembles, and almost all works use homogeneous ensembles or a single word embedding strategy with multiple deep learning models. To bridge this gap, the hybrid and stacking ensemble models proposed in this article use three state-of-the-art heterogeneous word embeddings (including static and contextual) and BiLSTM-CNN.

3 Methodology

3.1 Data preprocessing

The ultimate aim of this submodule is to utilize NLP techniques for preprocessing the raw text sentences and arranging them for the subsequent step of extracting valuable features. Figure 2 depicts the brief preprocessing pipeline. The tokenized data point is sent through the pipeline in the preprocessing step to remove or normalize the useless tokens in the sarcasm recognition dataset. The following are the subparts of the preprocessing pipeline:

- *Handling Hashtags:* After identifying hashtags with the pound (#) sign, we separated # from the hashtag. Example: #TheProudFamily → TheProudFamily.
- *Stemming:* We stripped off the inflectional morphemes such as "est", "ed", "ing", and "s" from their token stems using Snowball stemmer (Usually called Porter2). Example: Words like "interested," "interesting," and "interests" are reduced to their base form, "interest".
- *Text cleaning:* We perform this step to discard the irrelevant data. We removed URLs, punctuation marks, html tags, digits, non-ASCII glyphs, and special characters from each data point.

We carried out each step of preprocessing for the SARC dataset. The first step was not carried out in the case of news headlines because the headlines do not include hashtags.

3.2 Word embedding

Quantitative data are necessary for computer algorithms to function. Therefore, data must be expressed quantitatively for algorithms to handle text-based data. There are numerous ways to do this, which can be categorized into three groups: Count-based methods (Count Vectorization, TF-IDF), Static word embeddings (Word2vec, GloVe, fastText), and Contextual embeddings (ELMo, BERT). For a lot of NLP activities, word representations are necessary. Effective word representations can enhance text encoding and classification capabilities. The core premise behind Word2vec is that rather than expressing words as one-hot representations (Count Vectorization/TF-IDF) in high-dimensional space, we present words in dense low-dimensional space so that similar words receive similar word vectors, allowing them to be mapped to their nearest neighbors. Word2vec does not utilize the information in the entire document as a window-based paradigm and does not acquire sub-word

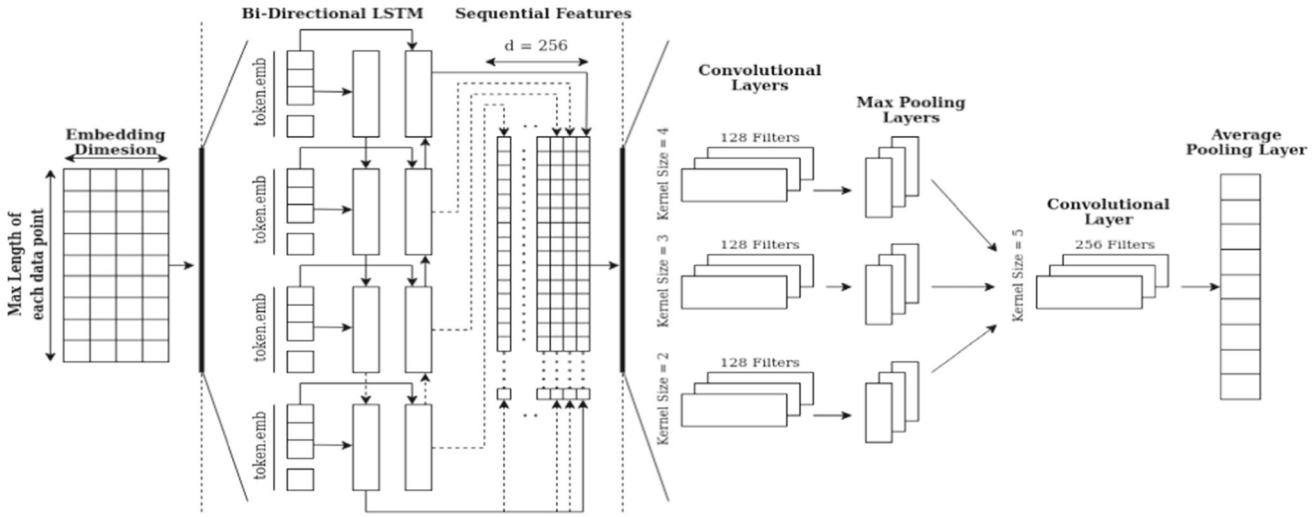

Fig. 3 BiLSTM-CNN Model

information. Disambiguation is another issue that Word2Vec does not address. The former problem is tackled with GloVe and FastText, while the latter is handled with BERT.

This module includes three pre-trained word embeddings: GloVe, fastText, and BERT. As a result, we get three different kinds of word embeddings by utilizing this module. We generated two pairs of 300-dimensional word embeddings using pre-trained GloVe[1] and fastText[2] models with 300-dimensional word vectors. Similarly, the third pair of 768-dimensional word embeddings is generated using the BERT-base,[3] a pre-trained base variant of BERT. The word vectors generated using these three particular embeddings will be given as input to BiLSTM-CNN.

3.3 BiLSTM-CNN model

The sequentially combined BiLSTM-CNN serves as the deep learning model. CNN and LSTM offer efficient outcomes in several applications as CNN is better at handling short phrases and capturing inter-dependencies between all conceivable word combinations. At the same time, LSTM can retrieve long-term correlations among word sequences. These techniques work well since they can contribute to the problem of categorizing sarcasm. The acquired categorization results have verified this effort. Combining the two yields considerably superior outcomes.

As shown in Fig. 3, utilizing word embeddings as input, our presented BiLSTM-CNN begins with a BiLSTM layer having a hidden dimension of 128. Three concurrent convolutional layers having kernel sizes of 2, 3, and 4 with 128 filters each are applied with this output. The yield from each convolutional layer feeds to max-pooling layers, and the outputs of all max-pooling layers combine to form a single feature vector comprising freshly extracted features. After that, the retrieved vector is fed into a convolutional layer comprising 256 filters and a kernel size of 5, followed by an average-pooling layer.

3.4 Proposed hybrid model

This section briefly presents the proposed hybrid-based sarcasm identification framework. As illustrated in Fig. 4, the overall framework comprises of several sub-modules, each of which performs a specific task. Initially, the data pre-processing module pre-processes the raw textual data by performing the steps described in Sect. 3.1. The cleaned data acquired after the pre-processing step has been given to three word embedding techniques (described in Sect. 3.2) separately, of which two are static and one is contextual. For the first two branches, GloVe and fastText use their pre-trained embeddings to generate the word vectors. On the other hand, the word vectors have been obtained from the sequence output of BERT in its third branch. The word vectors obtained from each embedding technique have been given to the BiLSTM-CNN model separately. For all three branches, we use the same BiLSTM-CNN architecture (described in Sect. 3.3) for acquiring three separate contextual features. From each branch, we now have the outcome of 256 features.

While we typically accept NumPy input for GloVe and fastText, the input type for BERT is a tensor. We employed the idea of Yuan et al. (2020) to deal with these mixed input types. We merged all the outcomes by applying this approach to get

[1] https://nlp.stanford.edu/data/glove.840B.300d.zip.

[2] https://dl.fbaipublicfiles.com/fasttext/vectors-english/crawl-300d-2M-subword.zip.

[3] https://huggingface.co/bert-base-uncased.

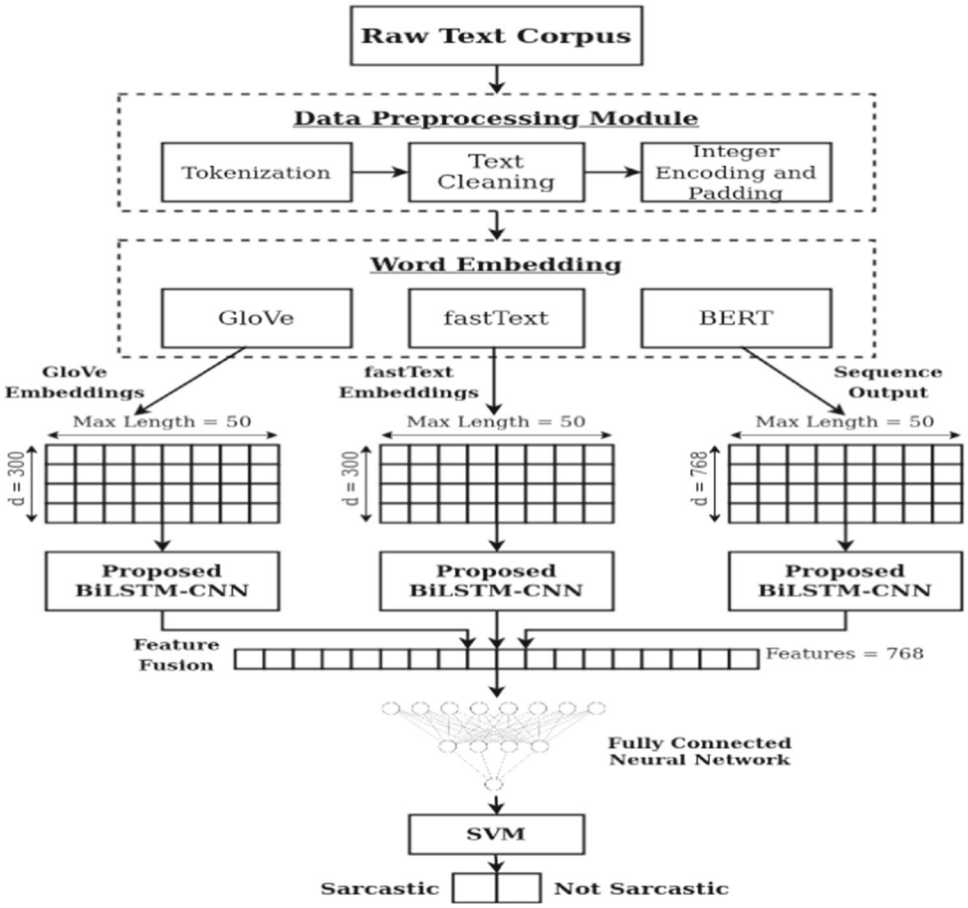

Fig. 4 Overview of the architecture of Proposed Hybrid Model

768 features. Consequently, these obtained features are sent to fully connected dense layers. Instead of Sigmoid activation function at the output layer, we use SVM for classifying the acquired features from hidden layers. Here, we employ SVM since it is a robust approach for resolving real-world binary classification tasks (Cervantes et al. 2020).

3.5 Proposed ensemble model

Figure 5 illustrates the structure of the suggested stacking ensemble-based sarcasm recognition framework that uses BiLSTM-CNN with heterogeneous word embeddings. The entire framework comprises various modules, just like the hybrid approach. We follow precisely the data preprocessing procedures outlined in Sect. 3.1, uses the word embedding strategies outlined in Sect. 3.2, and the BiLSTM-CNN architecture outlined in Sect. 3.3.

The stacking ensemble module consists of BiLSTM-CNN with three kinds of heterogeneous embeddings as the three base-level classifiers and a GRU-based meta classifier. Because SVM is successful in binary classification, we utilize it as the classifier at the output layer instead of sigmoid in all the base-level classifiers. The data points from the data preprocessing module that have been processed and integer-encoded are given as input to the base-level classifiers.

To avoid updating the embedding layer's weights after they have already been learnt, the parameter "trainable" of the first two base-level classifiers' embedding layers is set to False. The BERT's parameters are kept fixed to prevent its weights from changing, while the BiLSTM-CNN model is being trained. Figure 6 presents a schematic illustration of the process flow for converting the text documents into the appropriate word embedding matrices.

The integer-encoded data points obtained by the data preprocessing module are divided into the train (DS_{train}) and test (DS_{test}) sets at a ratio of 80%:20% to evaluate the suggested ensemble model. The base classifiers are trained on 90% of the data in the training set, and the leftover 10% is utilized to validate the trained base classifiers. Each learned base classifier is then appraised using the data from the testing set (DS_{test}). A single data frame (DS^f) contains four columns, out of which the first three columns have outcomes from the base classifiers on the test data (DS_{test}), and the fourth column includes labels from the test data (DS_{test}). Later, the DS^f is partitioned into training (DS^f_{train}) and testing (DS^f_{test}) sets with an 80%:20% ratio. Thereafter, 90% of the

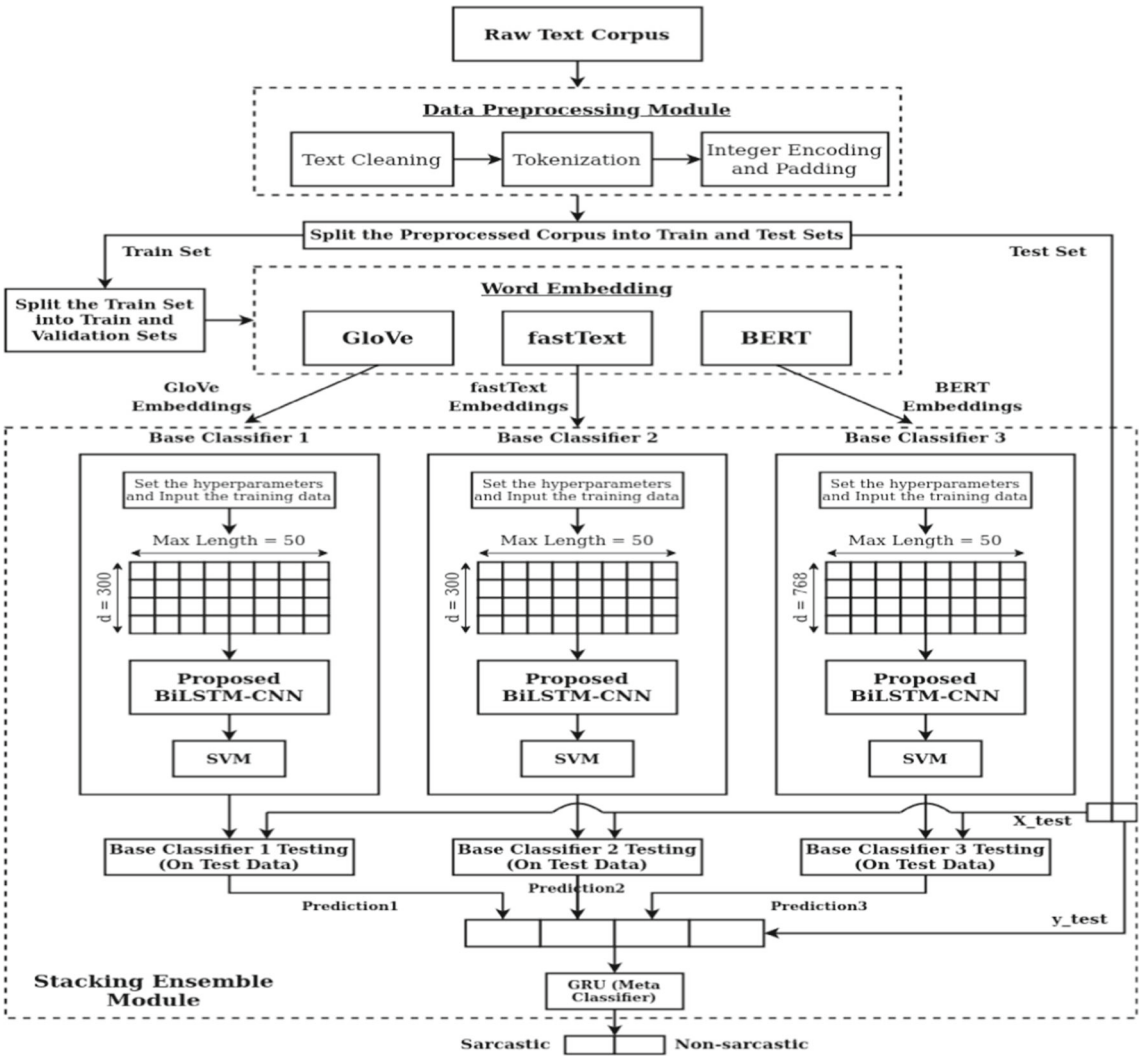

Fig. 5 Overview of the architecture of proposed stacking ensemble model

training (DS^{f}_{train}) data are used to train the GRU-based meta classifier, whereas the remaining 10% are used to validate the meta classifier. The trained meta-classifier is assessed on the test (DS^{f}_{test}) set to determine how well the recommended stacking ensemble-based classifier succeeds overall.

4 Experimental results and discussion

Numerous tests were carried out to demonstrate the viability of the suggested models. The efficacy of the suggested models is presented in this subheading utilizing a variety of evaluation criteria, including accuracy, F1-score, precision, and recall. We performed experiments in a Jupyter notebook running Python 3.9 from the Anaconda distribution on Ubuntu 20.04.4 LTS with an Intel Xeon (R) CPU (W-2133), 64GB of RAM, and a Quadro RTX 4000 GPU. Section 4.1 describes the datasets used for experimentation. Hyperparameters, Evaluation metrics, Baselines, and Results were all covered in sections 4.2, 4.3, 4.4, and 4.5, respectively.

4.1 Dataset

We have made use of two independent corpora in this work, one containing news headlines and the other with Reddit comments. In this subheading, we present an overview of these two corpora.

4.1.1 News headlines dataset

To overcome the noise limits in Twitter datasets, Misra and Arora (2019) built a news headlines dataset from two famous news sources: TheOnion and HuffPost. The dataset is in

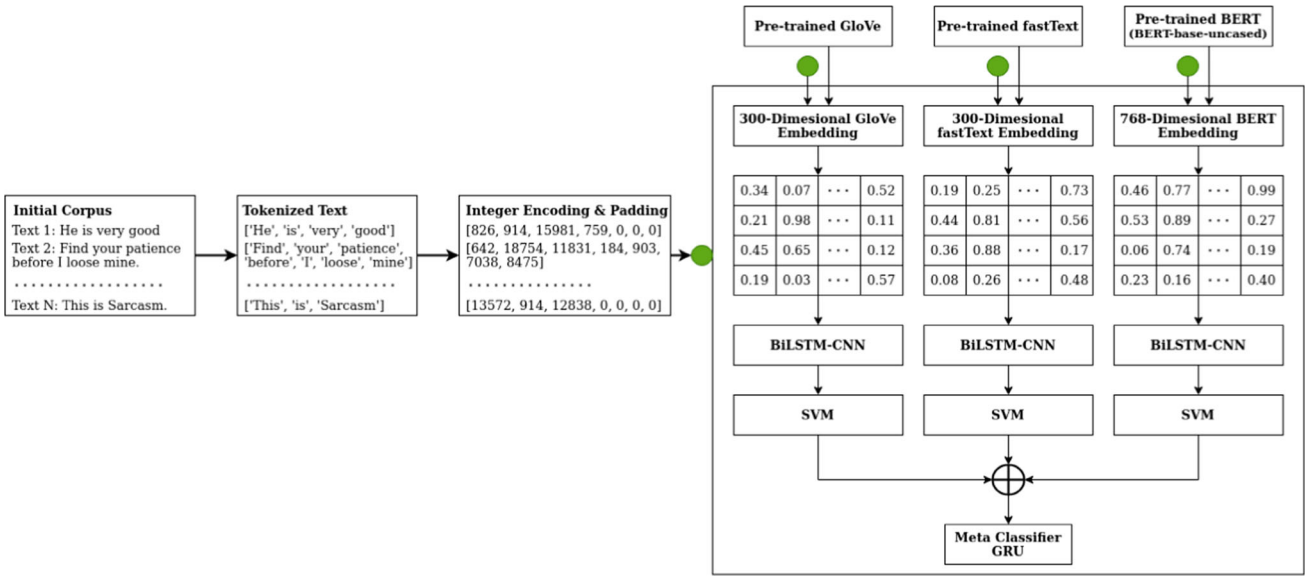

Fig. 6 A schematic diagram showing the transformation of textual documents into word-embedded matrices

JSON format and can be acquired from Kaggle.[4] There are 28,619 news headlines in the collection. Out of which, 47.6% of are sarcastic, whereas 52.4% are not. Each data point has three attributes. One is a binary variable that expresses whether or not the headline is sarcastic. The other is the news headline. The final one is the headline's URL. We excluded the URL because the objective is to judge whether a headline is sarcastic or otherwise.

4.1.2 SARC dataset

Khodak et al. (2017) developed the Self-Annotated Reddit Corpus (SARC), an enormous dataset for sarcasm identification. A portion of Reddit reviews between January 2009 and April 2017 comprise the dataset, which we can download from Kaggle.[5] The dataset comprises a sarcastic label, author, subreddit where the comment first appears, user-voted comment score, date of the comment, and parent comment. The corpus contains 1.3 million sarcastic utterances and many more non-sarcastic utterances. We considered the comments, each of which had at least ten words. The total number of comments is 4,41,637, with 2,25,974 being sarcastic and 2,15,663 being non-sarcastic.

4.2 Hyperparameters

Hyperparameters are the specific settings the user makes to regulate the learning process. The best/optimal hyperparameters must be chosen in the training step for learning algorithms to produce the most significant results. Table 1 presents the hyperparameters employed in our suggested approaches. After examining how the suggested methods performed with various sets of hyperparameters, we zeroed in on these values.

Table 1 Hyperparameters used for all the experiments

Parameter	Value
Activation Function	Sigmoid
Optimizer	Adam
Loss Function	binary_crossentropy
Learning Rate	2e-5
Batch Size	32/64
Number of Epochs	20
Dropout	0.2
ModelCheckpoint	Yes
EarlyStopping	Yes
Patience	5

It has been observed that the majority of headline/comment lengths vary from 10 to 50 words in both datasets. As a result, we set the maximum text length at 50. For texts with less than 50 words, we have included a sufficient number of zeros at the end of the headline/comment to guarantee a uniform length of 50. Similarly, headlines/comments with more than 50 words were condensed to just the first 50 words. With the incorporation of these changes, the total performance was unaffected because there were fewer headlines/comments with longer word counts. For each data point in both datasets, GloVe,

[4] https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection.

[5] https://www.kaggle.com/datasets/danofer/sarcasm.

fastText, and BERT produced embeddings with the following dimensions:

- *GloVe and fastText*: 2-D embedding matrix of dimension "50×300".
- *BERT*: 2-D embedding matrix of dimension "50×768".

We have used the features ModelCheckpoint and EarlyStopping of Keras library to save the best model during training. Different learning rates, including 2e-5, 3e-5, 4e-5, and 4e-5, were tested. The optimal value arrived at is 2e-5. Additionally, we experimented with dropouts of 0.2, 0.3, and 0.4. In this instance, 0.2 is the optimal value. For the news headlines dataset, we used a batch size of 32, while for the SARC dataset, we used a batch size of 64. Table 2 displays the complete count of the train, validation, and test sets.

4.3 Evaluation metrics

Quantifiable metrics have been devised as evaluation tools to assess the results of the categorization algorithms. Accuracy and F1-score were utilized as the assessment measures because the task required binary classification (1: Sarcastic, 0: Non-sarcastic). The percentage of accurately anticipated data points is the accuracy, whereas the F1-score combines precision and recall into a single statistic, which is a harmonic mean of both. The formulas for computing accuracy, precision, recall, and F1-score are shown in equations (1)–(4).

$$\text{Accuracy} = \frac{\text{Tr.P.} + \text{Tr.N.}}{\text{Tr.P.} + \text{Tr.N.} + \text{Fal.P.} + \text{Fal.N.}} \tag{1}$$

$$\text{Precision} = \frac{\text{Tr.P.}}{\text{Tr.P.} + \text{Fal.P.}} \tag{2}$$

$$\text{Recall} = \frac{\text{Tr.P.}}{\text{Tr.P.} + \text{Fal.N.}} \tag{3}$$

$$\text{F1} - \text{Score} = \frac{2 * \text{Tr.P.}}{2 * \text{Tr.P.} + \text{Fal.P.} + \text{Fal.N.}} \tag{4}$$

where

- The text that is correctly identified as being sarcastic is known as a Tr.P. (True Positive) because it originally belonged to that category.
- Tr.N. (True Negative) is the text originally belonging to the non-sarcastic class, and the model predicts the same.
- The text initially categorized as non-sarcastic but anticipated to be sarcastic is marked as Fal.P. (False Positive).
- Fal. N. (False Negative) is the text mistakenly classified as non-sarcastic, while it falls under the sarcastic category.

4.4 Baselines

Representing the texts accurately is one of the primary goals of a text classification task. GloVe, FastText, and BERT are the three word embedding strategies used in this work to represent the dataset. The efficacy of the proposed fusion and ensemble models was compared with 18 deep learning techniques. We name the neural network models M-1, M-2,..., M-18, and they are either CNN or RNN forms or a combination of both. The three deep neural techniques, CNN, BiLSTM, and BiGRU-also known as feature extractors and classifiers-make up the 18 deep learning models. We adopted the notations M-4-A, M-10-B, and M-16-C since M-4, M-10 and M-16 are the stems of our proposed approaches, as shown in Fig. 4 and 5. Further, the proposed fusion model is M-Hybrid, and the proposed ensemble model is M-Ensemble.

We tried six combinations with each of the word embedding approaches. For the CNN and BiLSTM, we used the same structure discussed in Sect. 3.3, whereas for the BiGRU, we used the hidden dimension of 128 in both directions.

4.5 Results and discussion

Tables 3, 4, 5, 6, and 7 detail the classification performance of the presented hybrid and ensemble models against the baselines. Compared to models with GloVe and fastText embeddings, those with BERT embeddings are observed to produce better outcomes. The models that employ GloVe and fastText have pretty comparable results. From Tables 3, 4, and 5, we can see that the BiLSTM-CNN model produces superior outcomes for all the static and contextual word embeddings that have been taken into account. The combinations M-4-A, M-10-B, and M-16-C prove the same. Hence, we integrated the same three pairs in the M-Hybrid and M-Ensemble frameworks. We presented a framework with two inputs and binary output using the Keras functional API to assess the classification performance of the M-Hybrid and M-Ensemble frameworks with that of the M-4-A, M-10-B, and M-16-C.

The comparative results of the suggested models on the news headline and SARC datasets are shown in Tables 6 and 7. On the news headlines dataset, Table 6 demonstrates that the M-Hybrid and M-Ensemble models, with 95.70% and 96.76% accuracy, respectively, clearly surpass all other standalone models. Table 7 shows that the suggested models also perform well on the SARC dataset, with accuracy values of 80.64% and 81.46%, respectively. Figure 7, 8 picture the M-Hybrid and M-Ensemble confusion matrices on both datasets. The number of data points in the confusion matrix is much less for M-Ensemble than M-Hybrid. The basis behind this is that initially, we used 20% of the data as test data for the base classifier to forecast. Again, we have picked 20% of the data from that to test the meta-classifier. The graphical

Table 2 Dataset statistics

Dataset	Training set		Validation set		Testing Set	
	non-sarc	sarc	non-sarc	sarc	non-sarc	sarc
News Headlines	10788	9818	1199	1090	2997	2727
SARC	155277	162702	17253	18078	43133	45194

Table 3 Results(%) of deep learning models with GloVe embeddings and SVM as the last layer for classification

Model name	Model combination	News headlines		SARC	
		Accuracy	F1-Score	Accuracy	F1-Score
M-1	CNN	85.95	85.60	72.98	73.27
M-2	BiLSTM	86.23	85.56	73.65	73.53
M-3	BiGRU	86.79	86.0	71.60	71.31
M-4-A	**BiLSTM-CNN**	**88.01**	**87.65**	**74.71**	**74.74**
M-5	BiGRU-CNN	86.93	86.23	73.65	73.63
M-6	BiLSTM-BiGRU	87.14	86.43	73.99	73.49

The best results are shown in bold

Table 4 Results(%) of Deep Learning Models with fastText embeddings and SVM as the last layer for classification

Model name	Model combination	News headlines		SARC	
		Accuracy	F1-Score	Accuracy	F1-Score
M-7	CNN	86.19	85.53	74.12	73.92
M-8	BiLSTM	86.89	86.17	74.06	73.56
M-9	BiGRU	86.09	85.59	71.87	72.02
M-10-B	**BiLSTM-CNN**	**87.66**	**87.02**	**75.05**	**75.18**
M-11	BiGRU-CNN	86.93	86.29	74.19	73.95
M-12	BiLSTM-BiGRU	86.82	86.18	74.16	74.35

The best results are shown in bold

Table 5 Results(%) of deep learning models with Sequence output from BERT and SVM as the last layer for classification

Model name	Model combination	News headlines		SARC	
		Accuracy	F1-Score	Accuracy	F1-Score
M-13	CNN	93.39	92.87	77.04	77.43
M-14	BiLSTM	93.43	92.98	78.10	78.94
M-15	BiGRU	92.24	91.85	77.70	78.08
M-16-C	**BiLSTM-CNN**	**94.51**	**94.16**	**79.62**	**79.67**
M-17	BiGRU-CNN	93.74	93.41	78.47	79.30
M-18	BiLSTM-BiGRU	93.88	93.50	78.73	79.33

The best results are shown in bold

Table 6 Results(%) of proposed M-Hybrid, M-Ensemble, and baselines on news headlines with BiLSTM-CNN as the model combination

Model name	Word embedding	Accuracy	Precision	Recall	F1-Score
M-4-A	GloVe	88.01	87.12	88.19	87.65
M-10-B	fastText	87.66	88.35	85.73	87.02
M-16-C	BERT	94.51	93.78	94.55	94.16
M-Hybrid	Glove + fastText + BERT	**95.70**	**95.38**	**95.66**	**95.52**
M-Ensemble	Glove + fastText + BERT	**96.76**	**96.89**	**96.37**	**96.63**

The best results are shown in bold

Table 7 Results(%) of proposed M-Hybrid, M-Ensemble, and baselines on SARC with BiLSTM-CNN as the model combination

Model name	Word embedding	Accuracy	Precision	Recall	F1-Score
M-4-A	GloVe	74.71	75.71	73.78	74.74
M-10-B	fastText	75.05	75.85	74.52	75.18
M-16-C	BERT	79.62	80.98	78.39	79.67
M-Hybrid	Glove + fastText + BERT	**80.64**	**81.69**	**79.84**	**80.75**
M-Ensemble	Glove + fastText + BERT	**81.46**	**82.04**	**81.28**	**81.66**

The best results are shown in bold

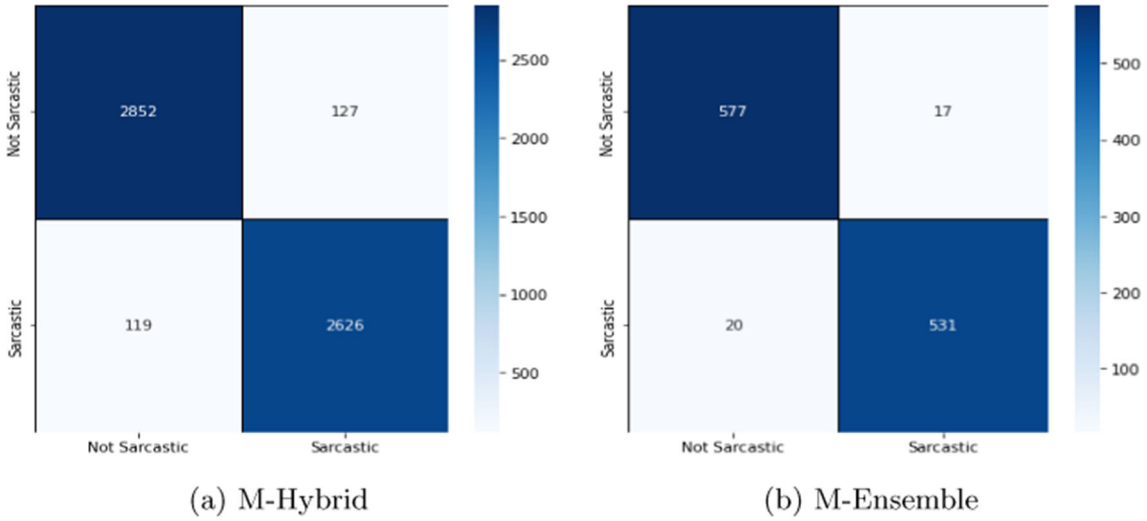

Fig. 7 Confusion matrices of M-Hybrid and M-Ensemble on News Headlines Dataset

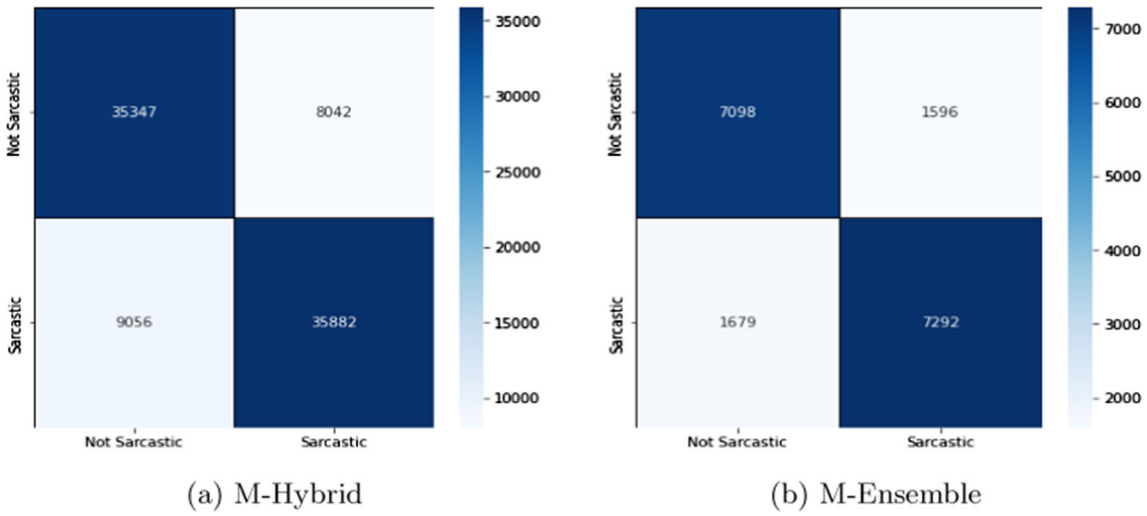

Fig. 8 Confusion matrices of M-Hybrid and M-Ensemble on SARC Dataset

depiction of suggested approaches and their stems for both datasets is shown in Fig. 9. Table 8 compares the suggested techniques with a few relevant state-of-the-art frameworks reported in the literature. In the case of the news headlines dataset, it can be seen from the table that the recommended frameworks perform noticeably better than previous frameworks. The M-Ensemble technique outperforms all others on the SARC dataset.

As a result of using three distinct kinds of word representations supplied by GloVe, fastText, and BERT, the proposed approaches perform better than the earlier ones. This makes it possible for the suggested frameworks to accurately represent the contextual dependencies between words in the

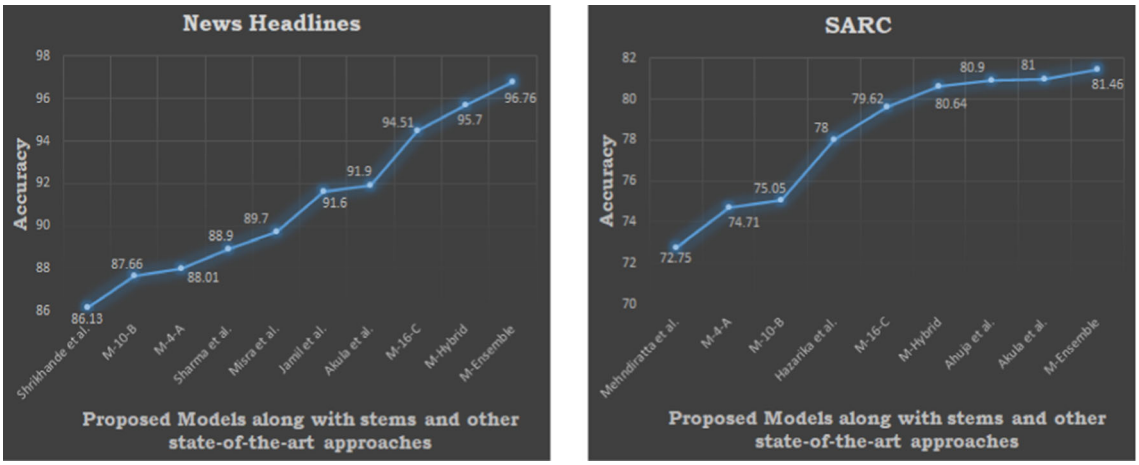

Fig. 9 Comparative performance of proposed models with their stems and other state-of-the-art approaches on both datasets

Table 8 Performance(%) comparison of the methods proposed with state-of-the-art methods

Research paper	Dataset	Accuracy	Precision	Recall	F1-Score
Misra and Arora (2019)	News Headlines	89.7	–	–	–
Shrikhande et al. (2020)		86.13	84.4	86.8	85.6
Pandey et al. (2021)		–	88	88	88
Jamil et al. (2021)		91.6	91	91	91
Akula and Garibay (2021)		91.9	91.8	91.8	91.6
Sharma et al. (2022)		88.9	91.1	88.67	89.87
M-Hybrid (Proposed)		95.70	95.38	95.66	95.52
M-Ensemble (Proposed)		96.76	96.89	96.37	96.63
Hazarika et al. (2018)	SARC	78	–	–	77
Mehndiratta and Soni (2019)		72.75	–	–	–
Savini and Caragea (2020)		–	–	–	76.3
Akula and Garibay (2021)		81	–	-	81
Ahuja and Sharma (2022)		80.9	81.4	81.3	81.3
Savini and Caragea (2022)		–	–	–	77.53
M-Hybrid (Proposed)		80.64	81.69	79.84	80.75
M-Ensemble (Proposed)		81.46	82.04	81.28	81.66

texts, improving their ability to anticipate outcomes. Another aspect is that the proposed frameworks use an effective deep learning model, a sequential combination of BiLSTM and CNN, rather than the base forms of CNN and RNN, like in previous studies.

5 Conclusion

In this paper, we have proposed novel hybrid and stacking-based ensemble models for sarcasm identification with heterogeneous word embeddings and BiLSTM-CNN. The proposed hybrid model has been employed to successfully extract three sets of features by running the BiLSTM-CNN with three heterogeneous word embeddings: GloVe, fastText, and BERT. The extracted features from the three sets were fused and sent to an SVM classifier for classification. On the other hand, the BiLSTM-CNN model with three types of heterogeneous word embeddings forms the three base-level classifiers of the proposed stacking ensemble-based framework. The base classifier probabilities are then combined into a single data frame, which is subsequently used to train the meta classifier, GRU. The hybrid model produced promising results, with 95.70% accuracy on the news headline repository and 80.64% on the SARC repository. In addition, very promising results have been attained with the stacking ensemble-based model. With this model, the accuracy obtained is 96.76% on the news headlines and 81.46% on the SARC. The proposed stacking ensemble-based model has outperformed all reported works in this area with superlative outcomes on both datasets. The experiments suggest that

combining heterogeneous word embeddings enhances sarcasm identification performance.

As sarcasm detection is a vast and fascinating field, there is much work to explore in the future. The popularity of typo-graphic images-text that is portrayed as an image-shows off the expressiveness of online social data even more, and sarcasm detection in them is a fascinating area for future research. Additionally, the usage of code-mix and code-switch languages, intentional ambiguity, unique vocabulary, "crowd-sourced" or "self-tagging" datasets, and other factors make it a vibrant field of study with many research challenges.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data and material availability The datasets used in the current study are publicly available datasets and are taken from Kaggle.

Declarations

Conflict of Interest The authors have no relevant financial or non-financial interests to disclose.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Ahuja R, Sharma SC (2022) Transformer-based word embedding with cnn model to detect sarcasm and irony. Arab J Sci Eng 47(8):9379–9392. https://doi.org/10.1007/s13369-021-06193-3

Akula R, Garibay I (2021) Interpretable multi-head self-attention architecture for sarcasm detection in social media. Entropy 23(4):394. https://doi.org/10.3390/e23040394

Albahar M (2021) A hybrid model for fake news detection: Leveraging news content and user comments in fake news. IET Inform Secur 15(2):169–177. https://doi.org/10.1049/ise2.12021

Ay Karakuş B, Talo M, Hallaç İR et al (2018) Evaluating deep learning models for sentiment classification. Concurr Comput Pract Exp 30(21):e4783. https://doi.org/10.1002/cpe.4783

Azwar AS, et al (2020) Sarcasm detection using multi-channel attention based blstm on news headline https://doi.org/10.21203/rs.3.rs-63423/v1

Bhardwaj S, Prusty MR (2022) Bert pre-processed deep learning model for sarcasm detection. Nat Acad Sci Lett. https://doi.org/10.1007/s40009-022-01108-8

Bojanowski P, Grave E, Joulin A et al (2017) Enriching word vectors with subword information. Trans Assoc comput Linguist 5:135–146. https://doi.org/10.1162/tacl_a_00051

Briskilal J, Subalalitha C (2022) An ensemble model for classifying idioms and literal texts using bert and roberta. Information Processing & Management 59(1):102–756. https://doi.org/10.1016/j.ipm.2021.102756

Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L et al (2020) A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing 408:189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Eke CI, Norman AA, Shuib L (2021) Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and bert model. IEEE Access 9:48501–48518. https://doi.org/10.1109/access.2021.3068323

Ghayoomi M, Mousavian M (2022) Deep transfer learning for covid-19 fake news detection in persian. Expert Syst. https://doi.org/10.1111/exsy.13008

Goel P, Jain R, Nayyar A et al (2022) Sarcasm detection using deep learning and ensemble learning. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-12930-z

Gundapu S, Mamidi R (2021) Transformer based automatic covid-19 fake news detection system. arXiv preprint arXiv:2101.00180

Hazarika D, Poria S, Gorantla S, et al (2018) Cascade: Contextual sarcasm detection in online discussion forums. arXiv preprint arXiv:1805.06413

He B, Hu W, Zhang K et al (2022) Image segmentation algorithm of lung cancer based on neural network model. Expert Systems 39(3):e12.822. https://doi.org/10.1111/exsy.12822

Jamil R, Ashraf I, Rustam F et al (2021) Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model. PeerJ Comput Sci 7:e645. https://doi.org/10.7717/peerj-cs.645

Jindal K, Aron R (2021) A systematic study of sentiment analysis for social media data. Mater Today Proc. https://doi.org/10.1016/j.matpr.2021.01.048

Joshi A, Bhattacharyya P, Carman MJ (2017) Automatic sarcasm detection: a survey. ACM Comput Surv (CSUR) 50(5):1–22. https://doi.org/10.1145/3124420

Khodak M, Saunshi N, Vodrahalli K (2017) A large self-annotated corpus for sarcasm. arXiv preprint arXiv:1704.05579

Kumar A, Sangwan SR, Arora A et al (2019) Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access 7:23319–23328. https://doi.org/10.1109/ACCESS.2019.2899260

Kumaran P, Chitrakala S (2022) A novel mathematical modeling in shift in emotion for gauging the social influential in big data streams with hybrid sarcasm detection. Concurr Comput Pract Exp. https://doi.org/10.1002/cpe.6597

Liebrecht C, Kunneman F, van den Bosch A (2013) The perfect solution for detecting sarcasm in tweets #not. In: Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. Association for Computational Linguistics, Atlanta, Georgia, pp 29–37, https://aclanthology.org/W13-1605

Mehndiratta P, Soni D (2019) Identification of sarcasm using word embeddings and hyperparameters tuning. J Discret Math Sci Cryptogr 22(4):465–489. https://doi.org/10.1080/09720529.2019.1637152

Mikolov T, Chen K, Corrado G, et al. (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

Misra R, Arora P (2019) Sarcasm detection using hybrid neural network. arXiv preprint arXiv:1908.07414

Nassif AB, Shahin I, Attili I et al (2019) Speech recognition using deep neural networks: A systematic review. IEEE access 7:19143–19165. https://doi.org/10.1109/access.2019.2896880

Pandey R, Singh JP (2023) Bert-lstm model for sarcasm detection in code-mixed social media post. J Intell Inform Syst 60(1):235–254

Pandey R, Kumar A, Singh JP et al (2021) Hybrid attention-based long short-term memory network for sarcasm identification. Applied Soft Computing 106(107):348. https://doi.org/10.1016/j.asoc.2021.107348

Patwa P, Bhardwaj M, Guptha V, et al (2021) Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and

hindi hostile posts. In: International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer, pp 42–53, https://doi.org/10.1007/978-3-030-73696-5_5

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543, https://doi.org/10.3115/v1/d14-1162

Potamias RA, Siolas G, Stafylopatis AG (2020) A transformer-based approach to irony and sarcasm detection. Neural Comput Appl 32(23):17309–17320. https://doi.org/10.1007/s00521-020-05102-3

Praseed A, Rodrigues J, Thilagam PS (2023) Hindi fake news detection using transformer ensembles. Eng Appl Artif Intell 119(105):731

Rahman A, Verma B (2013) Cluster-based ensemble of classifiers. Expert Syst 30(3):270–282. https://doi.org/10.1111/j.1468-0394.2012.00637.x

Salur MU, Aydin I (2020) A novel hybrid deep learning model for sentiment classification. IEEE Access 8:58,080-58,093. https://doi.org/10.1109/ACCESS.2020.2982538

Sarsam SM, Al-Samarraie H, Alzahrani AI et al (2020) Sarcasm detection using machine learning algorithms in twitter: a systematic review. Int J Market Res 62(5):578–598. https://doi.org/10.1177/1470785320921779

Savini E, Caragea C (2020) A multi-task learning approach to sarcasm detection (student abstract). In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 13,907–13,908, https://doi.org/10.1609/aaai.v34i10.7226

Savini E, Caragea C (2022) Intermediate-task transfer learning with bert for sarcasm detection. Mathematics 10(5):844. https://doi.org/10.3390/math10050844

Sharma DK, Singh B, Garg A (2022) An ensemble model for detecting sarcasm on social media. In: 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, pp 743–748, https://doi.org/10.23919/INDIACom54597.2022.9763115

Shrikhande P, Setty V, Sahani A (2020) Sarcasm detection in newspaper headlines. In: 2020 IEEE 15th international conference on industrial and information systems (ICIIS), IEEE, pp 483–487, https://doi.org/10.1109/ICIIS51140.2020.9342742

Shrivastava M, Kumar S (2021) A pragmatic and intelligent model for sarcasm detection in social media text. Technol Soc 64(101):489. https://doi.org/10.1016/j.techsoc.2020.101489

Srinivasarao U, Sharaff A (2021) Sentiment analysis from email pattern using feature selection algorithm. Expert Syst. https://doi.org/10.1111/exsy.12867

Subba B, Kumari S (2022) A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. Comput Intell 38(2):530–559. https://doi.org/10.1111/coin.12478

Yaghoobian H, Arabnia HR, Rasheed K (2021) Sarcasm detection: A comparative study. arXiv preprint arXiv:2107.02276

Yuan Z, Jiang Y, Li J, et al (2020) Hybrid-dnns: Hybrid deep neural networks for mixed inputs. arXiv preprint arXiv:2005.08419

Zhao F, Zhang J, Chen Z et al (2020) Topic identification of text-based expert stock comments using multi-level information fusion. Expert Syst. https://doi.org/10.1111/exsy.12641