Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-025-02530-z

ORIGINAL RESEARCH

Encoder-decoder model with attention mechanism for sarcasm interpretation on social media text

Kartika Makkar[1] · Pardeep Kumar[1] · Monika Poriye[1] · Shalini Aggarwal[1,2]

Received: 20 December 2024 / Accepted: 28 March 2025

Abstract Sarcastic posts are common on social media, where user's express sentiments that differ from the literal meaning of their words. Accurately understanding the intended meaning behind these posts could greatly enhance the effectiveness of text analysis tools for social media, a topic that remains under-explored. This research introduces a bidirectional long and short term memory (BiLSTM) based encoder-decoder model incorporating an attention mechanism and SpaCy-based embeddings. The model processes sarcastic input text and generates their corresponding non-sarcastic interpretations. A key feature of the proposed model is its bidirectional architecture, enabling it to capture contextual information in both forward and backward directions. By utilizing an attention mechanism, the model focuses on relevant preceding and succeeding words to make accurate predictions. Additionally, external embeddings have been integrated into the model to enhance its performance, addressing limitations in existing sarcasm interpretation models. Which operate unidirectionally without leveraging external embeddings and attention mechanisms. The results demonstrate that the proposed model outperforms traditional unidirectional models in context generation, due to its attention-based bidirectional nature with external embedding for interpreting sarcasm.

Keywords Bidirectional long and short term memory · Gated neural network · Long and short term memory · Recurrent neural network

1 Introduction

As social media platforms continue to grow, they become vital spaces for individuals to share their opinions on various topics, including products, brands, and political entities. These platforms are not only a medium for personal expression but also a valuable resource for businesses and governments to gather insights about public opinion and consumer preferences for strategic planning [1, 2]. However, analyzing social media data is complicated by the frequent use of figurative language like sarcasm and irony, which can obscure the true sentiments expressed by users [3, 4]. Consequently, the development of methods that can identify sarcasm in text and interpret its underlying meaning of sarcasm is gaining momentum [4, 5]. These advancements are crucial for improving the accuracy of tools that analyze social media content for various applications, including sentiment analysis, retweets prediction, digital marketing, and text summarizing [5–7]. Although sarcasm interpretation holds considerable importance, it remains a relatively unexplored research area, with only a limited number of researchers addressing this topic [8, 9]. Most research in this field has largely been dedicated to

Pardeep Kumar, Monika Poriye and Shalini Aggarwal have contributed equally to this work.

✉ Kartika Makkar
sonikartika19@gmail.com

Pardeep Kumar
mittalkuk@gmail.com

Monika Poriye
monikaporiye@gmail.com

Shalini Aggarwal
aggshamit@gmail.com

[1] Department of Computer Science & Application, Kurukshetra University, Kurukshetra 136119, Haryana, India

[2] Department of Computer Science, S.U.S. Govt. College, Matak Majri (Indri), Karnal 132041, Haryana, India

Published online: 21 April 2025

developing techniques that determine whether a text exhibits sarcasm [4, 10–14]. Imagine a review has been encountered,"*Perfect shirt for job interviews…if you aspire to stay unemployed."* This sarcastic comment, due to its seemingly positive phrase"*Perfect shirt for job interviews*", could be mistakenly identified as a positive review by the recommendation system. This misinterpretation of sarcastic reviews can potentially mislead these systems, thereby compromising their reliability. For sarcasm interpretation, the incongruity between a negative situation and a positive word or vice-versa plays a significant role. In the above utterance"*perfect*" is a positive word but it is used with a negative situation"*if you aspire to stay unemployed.*". Such words require more attention and better context for accurate sarcasm interpretation.

Despite the significance of this issue, it appears that only a few studies have concentrated on interpreting the sarcastic text found in social media posts [8, 9]. To address this challenge, a BiLSTM based encoder-decoder model has been proposed for sarcasm interpretation. The key contributions of the proposed work are detailed as follows:

1. We propose a two-layered BiLSTM-based encoder-decoder model with an attention mechanism, to better understand the context of sarcastic utterances in the forward and backward directions for sarcasm interpretation.
2. Extensive experiments have been performed on the proposed model with three attention mechanisms such as Local, Global, and Bahdanau to find the best-fit attention for the proposed model.
3. To evaluate the impact of sentence length on the performance of the model, we derive four datasets from the original dataset and implement the proposed model on four distinct datasets, each featuring utterances of varying lengths.
4. Various experiments have been conducted to find the best-fit word embedding technique using three external embeddings such as Word2Vec, Global Vectors for Word Representation (Glove), and SpaCy-based embedding for the proposed architecture.
5. To investigate the impact of the number of BiLSTM layers and dropouts on the proposed model, we implement the proposed model using different layers (1, 2, 3) on different dropouts (0.5, 0.3, 0.1, and 0).

2 Related work

2.1 Sarcasm interpretation

Sarcasm interpretation refers to the process of discerning the underlying non-sarcastic meaning behind the sarcastic text. From the literature, we found most of the approaches focused only on sarcasm detection and there are limited approaches related to sarcasm interpretation and are discussed as follows.

Sarcasm interpretation model was proposed known as Sarcasm Sentimental Interpretation GeNerator (SIGN) [8] using monolingual machine translation based on sentiments. SIGN focuses on sentiment words within sarcastic statements. Initially, it groups these sentiment words based on their semantic similarities. Next, each occurrence of sentiment token is replaced by its corresponding cluster label, and this altered data is given to a Machine Translation (MT) model (Moses in this case) during the training as well as the testing phase. During testing, the MT system generates non-sarcastic statements where sentiment words are replaced by their clusters. Lastly, SIGN executes a declustering step on the MT system's output, where it replaces each sentiment cluster with appropriate words. This model was reported with 66.96% Bleu (Bilingual Evaluation Understudy) score, Rouge-1 (Recall-Oriented Understudy for Gisting Evaluation) 69.67%, 40.96% Rouge-2 and 69.98% Rouge-L. The main advantage of this work is that the proposed model works well even if there are a few sentiment word occurrences in a sentence. In another research [9], three different techniques were proposed for sarcasm interpretation, (1) Rule-Based technique (2) Statistical Machine Translation (SMT) Based technique (3) Neural Network (NN) Based techniques. For the Rule-based technique, this work only revolves around searching for the correct position to put a negation word like "*not*" in a sentence. The key idea of this work involves linking the verb with a negative word by utilizing a list of negative words. For instance, "*headaches are fun*" is interpreted as "*headaches are not fun*" by simply putting not with the verb in an utterance. The second approach employed the Moses toolkit [15] for Phrase-based translation. To tackle the sarcasm interpretation task, an 8-word phrase table was utilized, treating it as a monolingual Machine Translation task. In the third approach, three experiments were performed such as Gated neural network (GRU) based encoder-decoder model, encoder-decoder with attention model, and pointer generator network to handle the issue of the attention layer for long sentences. All the experiments were performed on two datasets of long and short tweets and the results revealed that the short tweets performed well and among all the models the performance of the Statistical Machine Translation-based approach outperforms other approaches for Bleu score. A Bleu score of 78.57% was reported. Attention network performed well for meteor score with 45.02% and the encoder-decoder model outperformed Rouge-L with 80.89% score. The summary of sarcasm intrepretation models is given in Table 1.

Table 1 Recent studies on sarcasm interpretation

Ref	Dataset	Approach	Limitation	Future scope
2017, [8]	Twitter	Sarcasm SIGN, Moses, Recurrent neural network (RNN)	Model left the sentences uninterpreted without word knowledge	To improve the model by making it contextually rich so that it can understand the meaning of words where the sentiment of the words is not expressed clearly
2019, [9]	Twitter	Rule-based, SMT, deep learning	Model trained and evaluated on small dataset	(1) To improve the performance of deep learning-based model for sarcasm interpretation (2) Rule-based approach: (2.1) Focus on analyzing dependency based parse trees of sarcastic tweets to gain deeper insights into the optimal placement of negation words (2.2) To include a broader range of negation words and consider adjectives when associating them with negations (3) Generate new large dataset for model evaluation
2022, [16]	Adults data	Sarcasm interpretation based on emoji's using log-linear models of statistical package for social sciences (SPSS)	(1) Focus on social connections and develop communication styles rooted in Confucian principles (2) small and less diverse population sample size	(1) A cross-cultural investigation into how these varied communication styles impact the interpretation of sarcasm (2) Increase the sample size and include a more diverse population to enhance the representativeness of the data

2.2 Sarcasm detection

The summary of sarcasm detection models is given in Table 2. The literature review identifies several gaps in the existing research on sarcasm interpretation using deep learning methods [8, 9]. Notably, no studies have applied a bidirectional encoder-decoder based approach with attention mechanisms, which could improve context generation of the sarcasm interpretation. Additionally, the current models do not incorporate external embeddings, which could potentially enhance their performance. The proposed model addresses these challenges and aims to develop a more effective framework for sarcasm interpretation.

3 Proposed work

The proposed BiLSTM-based encoder-decoder model with attention layer and word embedding mainly consists of three parts i.e. encoder, attention layer, and decoder. The encoder receives the sarcastic input, the attention layer helps to make the model more contextually rich, and decoder generates non-sarcastic output. The proposed model is demonstrated in detail below.

3.1 Data collection and pre-processing

This research has been using publicly available tweets dataset [8] containing 5000 sarcastic tweets and their corresponding non-sarcastic interpretations created by human annotators as shown in Table 3. In the original dataset [8] (dataset1) the tweet length is a maximum of 40 words. To investigate the impact of short length dataset on the proposed model, three more datasets have been created using the original dataset which are of 30-length (dataset2), 20-length (dataset3), and 10-length (dataset4). Various pre-processing techniques such as removal of special characters, letters, tokenization, and padding are applied to the dataset using the Natural Language Toolkit (NLTK). In addition, the out-of-vocabulary tokens (OOV tokens) have been handled on the dataset before feeding it to the model.

3.2 Word embedding

Word embedding techniques are required to convert the input data into vector form before giving it to the model for computations. To find the best word embedding technique for the proposed model, various experiments have been carried out with three external embedding methods namely, SpaCy-based embedding, Word2Vec, and Glove. Also, the performance of the model has also been accessed without

Table 2 Recent studies on sarcasm detection

Ref	Approach	Limitation	Future scope
2015, [17]	Maximum valued matrix-element ($MVME_{we}$)	–	(1) To further explore the characteristics and scale of training data beneficial for the LSSD task (2) To examine the challenge of noisy data on proposed approach
2015, [18]	Sarcasm classification using a behavioral modeling approach (SCUBA)	Model was evaluated only on Twitter dataset	(1) To implement model on other social media datasets and include their respective features (2) To identify other non-literal language forms, such as humor (3) To enhance SCUBA by incorporating user's social networks and their current and past interactions for improved sarcasm detection
2019, [19]	Sarcasm detection using multimodal	(1) Over-fitting in the model (2) In utilizing conversational context, incorporating various crucial conversation specific elements, such as the goals, intentions, and dependencies of the interlocutors were overlooked	(1) Address the over-fitting challenge through approaches such as pre-training, transfer learning, domain adaptation, or the use of low-parameter models (2) To include factors such as goals, intentions, and dependencies to enhance the performance of the model
2019, [20]	Hybrid neural network	(1) Model implemented on small dataset (2) Model lacks common sense knowledge	(1) To implement model on large dataset (2) To add common sense knowledge in the hybrid neural network model
2020, [21]	Affective and contextual embeddings (ACE1) and ACE2	–	To explore whether affective and contextual embeddings, such as WikiSarcA-BERT trained on ACE1, can enhance the performance of tasks like emotion detection

Table 3 Dataset consists of sarcastic input and its non-sarcastic interpretations generated by human [8]

	Sarcastic	Non-Sarcastic
1	what a great day to start	what a great day to start
2	what a great day to start	could not get a worse start to a day
3	what a great day to start	what a horrible day to start
4	what a great day to start	what a bad way to start a day

any external embedding as shown in Table 7. The proposed approach converts text into vector form using an embedding matrix (W_g) given below.

$$W_g \in \mathbb{R}^{|V| \times e} \tag{1}$$

$$C \in \mathbb{R}^{m \times l} \tag{2}$$

Every word in the text is converted into vector form with 300 embedding dimensions (e) using Eq. 1 where, R is the real number, and the vocabulary size (V) is set to 5000. The Glove method generates better embedding with fewer dimensions [22] by applying a matrix factorization approach to the word co-occurrence matrix for acquiring the knowledge of the linear relationship between the words to generate a lower-dimensional matrix [23]. Proposed approach also uses SpaCy-based embedding because it is faster as compared to other word embeddings [24]. While Glove and Word2Vec represents text data into a static vector regardless of the context. SpaCy adds contextualized embedding as it's based on the recent transformer-based architecture which provides a nuanced representation and is known to perform well in scenarios where context is important.

3.3 Encoder

Encoder part consists of two layers of BiLSTM which performs computations bidirectionally i.e. it considers both future and past information of a word for the computation of context.

The working of the proposed approach is based on conditional probability $p(y|x)$ for translating a sarcastic input sentence $x_1, \ldots, x_j$ into a non-sarcastic output sentence $y_1, \ldots, y_j$. For this interpretation, the embedding matrix obtained in Eq. 2 is given to BiLSTM in the encoder as shown in Fig. 1 and this input sequence is processed bidirectionally to make it context rich. Then,

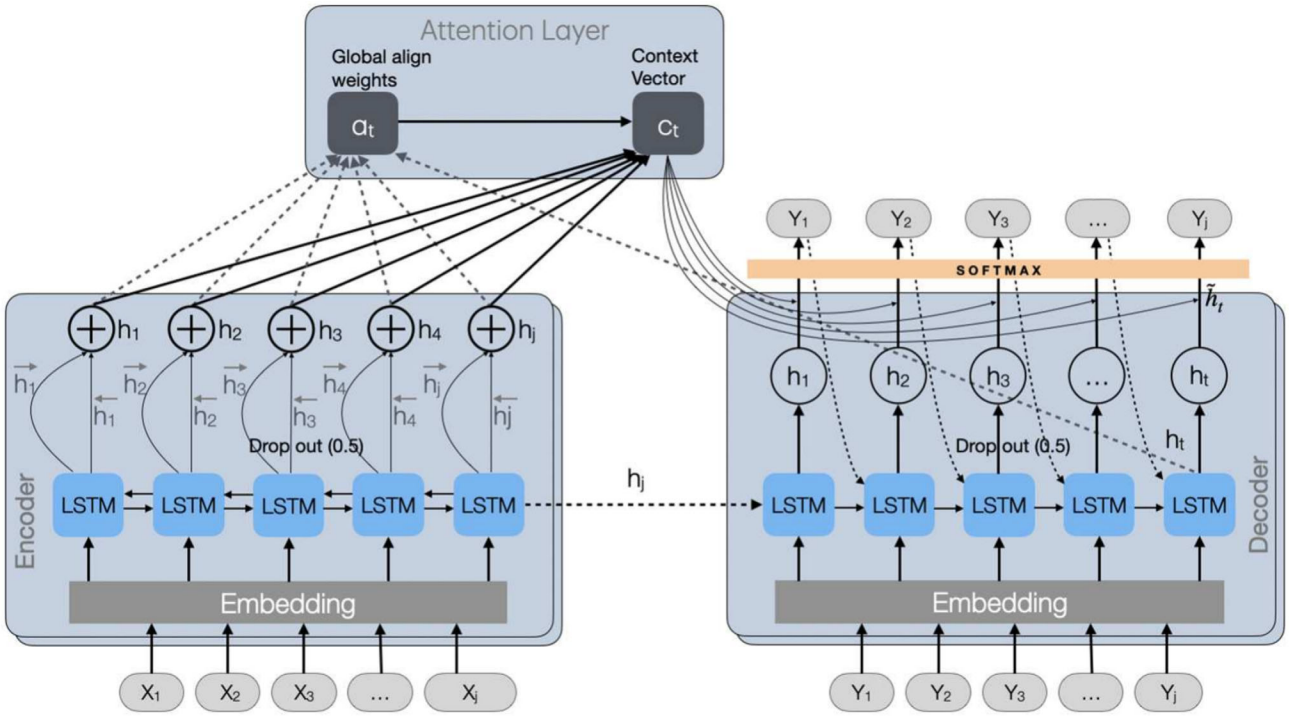

Fig. 1 Displays the 2-layer based BiLSTM encoder and 2-layer based LSTM decoder using global attention for sarcasm interpretation

concatenation of the forward $\underset{h_j}{\rightarrow}$ and the backward hidden state $\underset{h_j}{\leftarrow}$ for each input token x_j results into a bidirectional state $x_j = \left[\underset{h_j}{\rightarrow}; \underset{h_j}{\leftarrow}\right]$. This way h_j detailed in Eq. 3 consists of all the information about the preceding and succeeding words of x_j which is used for the computation of context vector c_t detailed in Eq. 6, that is used by the decoder. The context vectors c_t are generated using different attention mechanisms to find the best attention for the proposed model various attention mechanisms are applied and are detailed below.

3.4 Attention layer

Sarcasm mainly consists of positive words and negative situations or vice versa for instance"*Being left out is such an amazing feeling*". In this sentence"*left out*" is negative and"*amazing*" is a positive word. To capture this incongruity and better contextual information, the proposed work incorporates various attention mechanisms with an encoder-decoder model such as Global [25], Bahdanau [25, 26], and Local [25]. All these attention mechanisms are discussed in detail as follows.

3.4.1 Global attention

This section presents the Global attention mechanism for sarcasm interpretation which is simple and require minimum domain knowledge for computation. It captures the context information by considering all the words in a sarcastic utterance i.e. all hidden states of sarcastic input words from the encoder part are connected to the attention layer as shown in Fig. 1. To calculate the probability of decoding every word y_j in target utterance, hidden state h_j needs to be computed as shown in Eq. 3. Here, current hidden state h_j is computed by f (anyone among RNN, GRU, LSTM) using the preceding hidden state and x_j input word.

$$h_j = f\left(h_{j-1}, x_j\right) \tag{3}$$

Alignment vector is calculated by the dot product of present target hidden state h_t and all source hidden state h_j shown in Eq. 4

$$a_{t,j} = \mathrm{h}_t.h_j \tag{4}$$

$$\alpha_{t,j} = \frac{\exp\left(a_{t,j}\right)}{\sum_{j'=1}^{n}\exp\left(a_{t,j'}\right)} \tag{5}$$

Using the alignment score from Eq. 4, attention weights are computed using Eq. 5. Then the context vector c_t shown in Eq. 6 is calculated as a weighted sum of the

encoder's hidden states, with each hidden state h_j being multiplied by its respective attention weight $\alpha_{t,j}$.

$$c_t = \sum_{j=1}^{n} \alpha_{t,j} h_j \tag{6}$$

$$score(h_t, h_j) = \begin{cases} h_t^T h_j & dot \\ h_t^T W_a h_j & general \\ v_a^T \tanh\left(W_a [h_t; h_j]\right) & concat \end{cases} \tag{7}$$

Also, Fig. 2 shows the heatmap generated by the proposed model using global attention. Here, x-axis shows the input sarcastic sentence and y-axis shows the predicted non sarcastic sentences by the proposed model. The weights highlighted in different colors shows the importance of weights for making predictions. Scores highlighted in light color are important for prediction as compared to scores in bright color as depicted by the color scale.

3.4.2 Bahdanau attention

Bahdanau attention works globally i.e. it consider all the encoder hidden states for alignment score calculation. Bahdanau attention uses only concat (additive) function detailed in Eq. 7 for alignment score calculation but other scores can also be effective [26]. To find the impact of dot product score function given in Eq. 7, on alignment score, attention weights and context vector, is used with global attention in Sect. 3.4.1.

3.4.3 Local attention

This section presents the encoder-decoder model with Local attention mechanism. The significance of Local attention mechanism is it's capability to work with a small window of words at a time, which is an advantage compared to Global attention as shown in Fig. 3 and Fig. 1. Global attention has a drawback of mandating attention to all source words in a sentence [26] which makes it impractical and computationally expensive for the translation of long sequences such as paragraphs and documents.

In this model the context vector c_t is computed by the weighted average of source hidden states inside the window $[p_t - D_{loc}, p_t + D_{loc}]$. Here, D_{loc} represents the size of local window around the position p_t in a sequence. For every target word an aligned position p_t is generated by the model which is taken equal to t due to this input and output sequences are aligned to each other monotonically. Subsequently, the alignment vector, context vector c_t and attention weights are computed using Eq. 4, Eq. 5, and Eq. 6 based on the window used for the position p_t, the proposed model uses a window size of 5. The decoder utilizes the computed context vector to predict the subsequent.

non-saracstic output words.

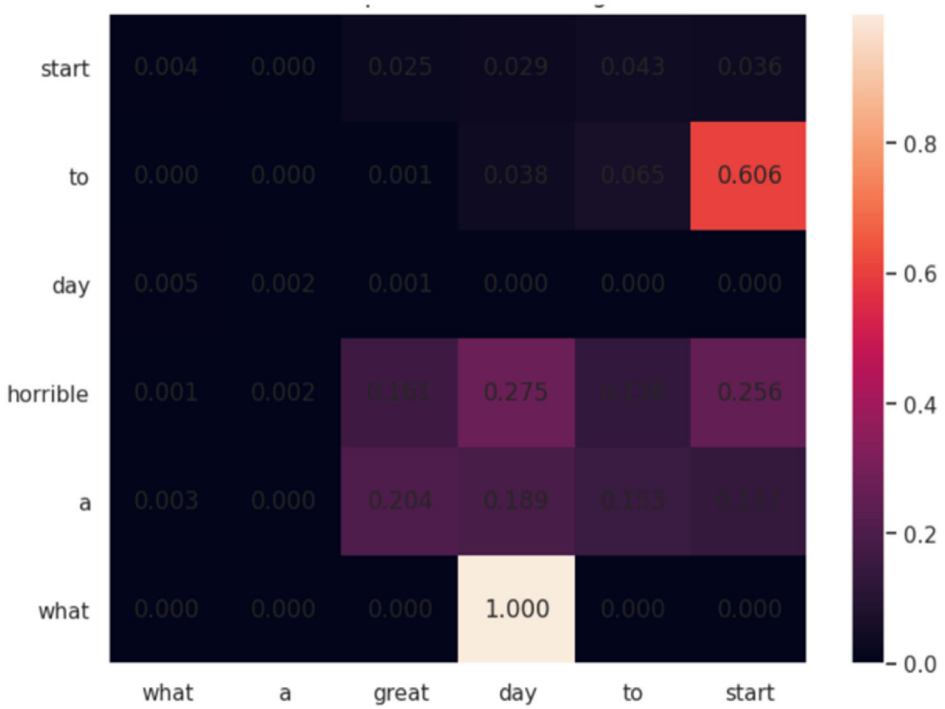

Fig. 2 Visualization of attention weights generated by attention mechanism for a sentence

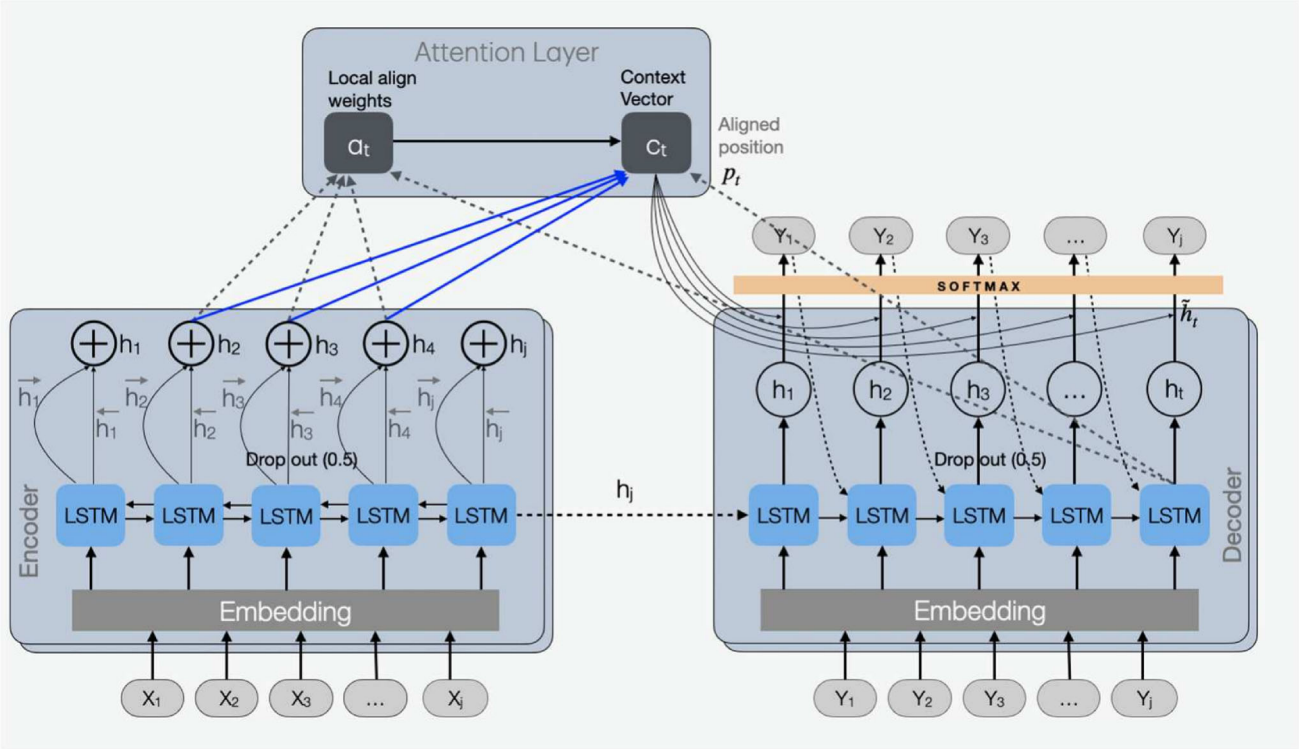

Fig. 3 Displays the 2-layer based BiLSTM encoder and 2-layer based LSTM decoder using Local attention for sarcasm interpretation

3.5 Decoder

Decoder part is mainly responsible for prediction task, and it perform this prediction using $\widetilde{h}_t$ detailed in Eq. 8 which is produced by the concatenation of context vector c_t and target hidden state h_t. Here, W_c is the weight matrix that transform the concatenation of h_t, c_t and *tanh* introduces non-linearity in the model. In Eq. 9 $\widetilde{h}_t$ is then fed to a *softmax* layer with weighted matrix (W_s) to assign probability to the vocabulary of words for the prediction of target non-sarcastic word.

$$\widetilde{h}_t = \tanh(W_c[c_t; h_t]) \tag{8}$$

$$p\left(y_j|\left(y_1, y_2, ..., y_{j-1}\right), x\right) = \text{softmax}\left(W_s\widetilde{h}_t\right) \tag{9}$$

$$L_{total} = -\frac{1}{l}\sum_{j=1}^{l} y_j \log\left(\widehat{y}_j\right) + \lambda||\theta||\frac{2}{F} \tag{10}$$

4 Experimental setup and results

The performance of the model has been evaluated on different aspects such as different number of BiLSTM and LSTM layers, datasets of varied lengths, word embeddings, various attention mechanisms, and dropouts are demonstrated below.

4.1 Model training and configuration

The proposed model consists of three parts i.e. encoder, decoder, and attention mechanism. Pre-processed data obtained from Sect. 3.1 have been used for creating word embedding of 300 dimensions. Post that the encoder, and decoder are applied to the input, and output embeddings to obtain the hidden states, then compute the alignment score using these hidden states. Then context vector c_t is generated by the attention mechanism is used by the decoder. The decoder generates one non-sarcastic output token at a time detailed in Algorithm 1. Then all the model parameters are updated for next word prediction and loss is computed according to Eq. 10. Here, sparse categorical cross-entropy loss function is used which calculate the average negative log-likelihood of the correct word at every position in the output sequence. Adam optimizer is used to tune the loss function during the process of stochastic gradient descent which can bring down the risk of gradient disappearance [27]. In Eq. 10, L represents the length of output sequence, y_j represents true word at position j in the output sequence $\widehat{y}_j$ is the distribution of the probability of predicting the vocabulary of the word at position j and $\lambda||\theta||\frac{2}{F}$ is the regularization used to prevent over-fitting in the model. To avoid over-fitting and to find the effectiveness of the model, various dropouts (0.5, 0.3,

0.1) have been employed. We have used learning rate = 0.001, epochs = 100, hidden units = 1024, batch size = 64. For final results we have considered the highest score of evaluation metrics among all epochs.

Algorithm 1 Pseudo-code of the proposed approach

Require	:	Input matrix of sarcastic sentences $C \in \mathbb{R}^{m \times l}$ and pre-trained embedding matrix $W_g \in \mathbb{R}^{\|V\| \times e}$, where l represents padding length of sequence, e represents embedding dimensions, V represents vocabulary and m represent s number of inputs.
Ensure	:	Generates non-sarcastic sentence corresponding to input sarcastic sentence.
1	:	**begin**
2	:	Use W_g given in eq. 1 to create word embedding matrix C given in eq. 2.
3	:	**for** i in $[1, n]$ **do**
4	:	Apply BiLSTM-based encoder and LSTM-based decoder on embedding vectors to obtain hidden states h_j, h_t of input and target sequence using eq. 3
5	:	Compute the alignment score using encoder and decoder hidden states according to eq. 4
6	:	Compute the attention weights using eq. 5.
7	:	Compute the context vector c_t by the product of attention weights and encoder hidden states using eq. 6
8	:	In the decoding phase, $\tilde{h}_t$ vector is calculated according to eq. 8 to predict output word y_t.
9	:	$\tilde{h}_t$ vector is then passed through the softmax layer to assign the probability to all the words in the vocabulary using eq. 9 and the word with highest probability is selected for prediction.
10	:	Decoder hidden states, attention weights, and corresponding context vectors are updated for next word prediction until the entire interpretation is completed.
11	:	Repeat steps 4-10 to obtain the output tokens of n input sequence.
12	:	**end for**
13	:	Evaluate the model performance using Bleu and Rouge score.
14	:	**end**

4.2 Results

The results of the proposed model on different lengths of datasets, different word embeddings, and different layers of BiLSTM, LSTM's layers with different attention mechanisms are discussed as follows.

4.2.1 Evaluation of the proposed model on the different number of BiLSTM, LSTM layers in Encoder-Decoder with various attentions

In Table 4, various experiments have been carried out using 1, 2, 3 layers of BiLSTM-based encoder and LSTM-based decoder with Glove embedding. From Table 4 it is reported that the 2-layer based model is giving a better performance as compared to 1 and 3 layers. So, the next experiments have been carried out using 2 layers based model.

4.2.2 Impact of different lengths (40, 30, 20, 10) of datasets on the proposed model

Table 5 and 6 display the performance of the encoder-decoder model using various attentions on four datasets of different lengths. From both tables, it has been observed that the performance of the encoder-decoder model using different attention layers is better for long text (up to 40 length) as compared to short-length text (up to 10, 20, 30 length or less). It can happen because model is getting

Table 4 Display the results of the encoder decoder model with different numbers of BiLSTM, and LSTM layers with attention

Attention	No. of layers	Bleu	Rouge-1	Rouge-2	Rouge-L
Global	1	43.57	65.18	47.75	63.88
Local	1	44.71	66.35	49.11	65.06
Bahdanau	1	44.70	66.35	49.11	65.06
Global	2	**47.50**	**67.45**	**51.72**	**66.76**
Local	2	**46.81**	**67.01**	**50.45**	**66.77**
Bahdanau	2	**47.26**	**67.36**	**51.36**	**66.60**
Global	3	45.95	65.58	49.58	64.89
Local	3	46.00	65.61	49.16	64.85
Bahdanau	3	46.2	66.02	49.82	65.31

Values in bold means highest score and % is omitted for Bleu and Rouge score. The above comparison is performed on dataset1

Table 5 Performance comparison of proposed model using different lengths (40, 30 length) dataset

Attention	Dataset1[1]				Dataset2[2]			
	Bleu	Rouge-1	Rouge-2	Rouge-L	Bleu	Rouge-1	Rouge-2	Rouge-L
Global	**46.18**	**65.45**	**49.14**	**64.50**	44.01	63.72	47.20	62.92
Local	**45.54**	**65.19**	**49.17**	**64.38**	42.49	61.79	45.17	61.30
Bahdanau	**46.50**	**66.29**	**49.97**	**65.40**	44.14	63.65	46.83	62.78

[1]dataset1 denotes original 40 length dataset

[2]dataset2 denotes 30 length dataset

Bold indicates higher value in the column

Table 6 Performance comparison of proposed model using different lengths (20, 10 length) dataset

Attention	Dataset3[3]				Dataset4[4]			
	Bleu	Rouge-1	Rouge-2	Rouge-L	Bleu	Rouge-1	Rouge-2	Rouge-L
Global	38.81	60.03	42.02	59.15	33.52	61.58	41.86	60.80
Local	39.07	59.93	42.31	59.22	31.09	57.27	37.45	56.68
Bahdanau	38.87	59.51	41.90	58.83	32.80	59.68	39.99	59.13

[3]dataset3 denotes 20 length dataset

[4]dataset4 denotes 10 length dataset

Table 7 Performance analysis of the proposed model using different word embeddings on dataset1

Embedding	Attention	Bleu	Rouge-1	Rouge-2	Rouge-L
Without external embedding	Global	44.29	62.86	47.03	62.18
	Local	43.12	61.90	45.36	60.96
	Bahdanau	44.15	63.44	46.68	62.60
Word2Vec	Global	43.15	61.84	45.84	61.06
	Local	42.41	60.60	45.31	59.82
	Bahdanau	42.52	61.08	45.47	60.53
Glove	Global	46.18	65.45	49.14	64.50
	Local	45.54	65.19	49.17	64.38
	Bahdanau	46.50	66.29	49.97	65.40
SpaCy based	Global	**47.50**	**67.45**	**51.72**	**66.76**
	Local	**46.69**	**66.62**	**50.45**	**66.77**
	Bahdanau	**47.26**	**67.36**	**51.36**	**66.60**

Bold values indicate the highest score in a column

better context from more word for long sentences and less context from short sentences to make predictions. Due to the better performance of the model with dataset1, further. experiments have been carried out using dataset1.

4.2.3 Performance of the proposed model according to different word embeddings

In Table 7, the effectiveness of the model is evaluated using three-word embedding techniques i.e. Word2Vec, Glove, and SpaCy-based embedding. Moreover, the performance of the model has also been accessed without any external embedding i.e. with default encoder decoder embedding. It has been observed that SpaCy-based embedding shows better results for Global Attention in terms of Bleu, Rouge-1, and Rouge-2. However, the Local Attention-based model shows better performance for

Table 8 Performance analysis of proposed method with global attention on different dropouts using SpaCy-based embedding

Embedding	Bleu	Rouge-1	Rouge-2	Rouge-L	Dropout
Global	**47.50**	**67.45**	**51.72**	**66.76**	**0.5**
	47.08	67.24	51.16	66.72	0.3
	47.33	67.33	51.21	66.67	0.1
	47.20	67.40	51.19	66.66	0.0

Bold indicates higher value in the column

Rouge-L as compared to the Global. In this comparison, the Global Attention-based model gives higher values for Rouge-1, Rouge-2, and Bleu scores but Local attention is better only for Rouge-L. Thus, further experiments have been performed using Global Attention and SpaCy-based embedding.

4.2.4 *Performance analysis of proposed method with global attention on different dropouts using SpaCy-based embedding*

In Table 8, the performance of the global attention-based model has been analyzed using various dropouts to reduce the over-fitting. Figure 4 shows that the performance of the model starts to increase as we increase the dropout from 0.0–0.5. Which means there is decrease in over-fitting. Figure 2 shows the attention weights generated by the model, these weights are further used for context generation and to make non-sarcastic predictions. The color scale of this figure highlights the importance of weights for making predictions. The range of this scale lies between 0.0–0.8 for this sentence, and the importance of weights increases as the range moves from 0–1. Table 9 demonstrates the non-sarcastic interpretations generated by the proposed model for sarcastic inputs. From this table we can see how accurate predictions are generated by the proposed model. Figure 4 depicts the loss curve throughout the training and validation stages of the proposed model. The loss starts decreasing as the training of the model starts and it is almost stable after 20 epochs with minor variations.

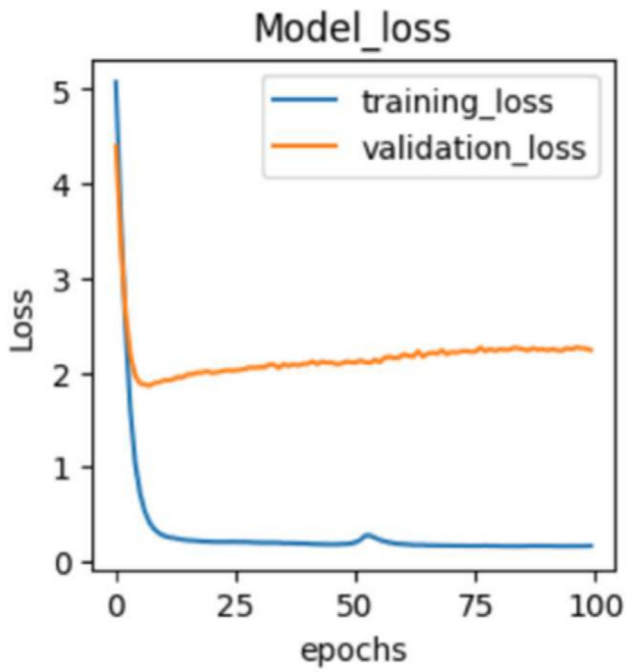

Fig. 4 Display the model loss during training and validation

This figure also depicts that there are minor variations between training and validation loss which means the model is performing well.

5 Comparison of results

In Table 10 it is reported that in terms of Bleu score, the performance of the proposed approach is improved by 6.45% as compared to RNN based encoder-decoder model [8] and 21.73% as compared to GRU based encoder-decoder with attention [9]. Also, GRU based encoder-decoder is giving better performance as compared to proposed approach for Bleu score. Furthermore, our BiLSTM based encoder-decoder model with global attention improves the performance by 25.25% for Rouge-1 and 21.75% for Rouge-2 as compared to [8], for Rouge-L Local attention based.

proposed model is giving 15.90%, 12.17% and 37.52% increase as compared to RNN based encoder-decoder [8], GRU based encoder-decoder, and GRU + attention based encoder decoder model [9]. Table 10 shows that the performance of RNN based encoder-decoder model is not good among all the approaches and it can be due to the vanishing gradient issue which can cause RNNs to struggle with learning dependencies that extend beyond a short sequence range. This poses a significant challenge for applications that need to interpret complex sequences, where understanding extensive contextual information is crucial. Moreover, another variant of RNN i.e. GRU based encoder-decoder model perform well as compared to RNN as they are designed to address the vanishing gradient issue. GRUs regulate information flow through gating mechanisms.

Unidirectional GRUs are generally simpler and more computationally efficient. To capture complex context in sarcastic text, BiLSTM-based encoder-decoder model is used. It works well for Rouge score as compared to other approaches shown in Table 10 but, not performing well in terms of Bleu score as compared to GRU based encoder-decoder model it can be due to its complex architecture.

6 Conclusion and future work

Sarcastic text is complex to understand and interpret but the proposed model proves to be adept for sarcasm interpretation tasks. For efficient interpretation of sarcastic text, context is the most crucial part. Consequently, to achieve a more comprehensive understanding of the context bidirectionally, BiLSTM based encoder-decoder model with various attention mechanisms has been proposed. Among all these attentions, Global attention gave the best results

Table 9 Predicted non sarcastic outputs generated by the proposed model for an input sarcastic sentence

	Input (sarcastic)	Output (non-sarcastic)
1	what a great day to start	what a horrible day to start
2	so happy i always make the most of my weekends	so sad i always make the most of my weekends
3	beautiful day to be stuck inside working	hideous day to be stuck inside working
4	best day of my life	worst day of my life

Table 10 Comparison of proposed approach with existing Deep learning based approaches

Approaches	Bleu	Rouge-1	Rouge-2	Rouge-L
Encoder-Decoder (RNN) [8]	41.05	42.20	29.97	40.87
Encoder-Decoder (GRU) [9]	**53.60**	–	–	54.60
Encoder-Decoder (GRU) + attention [9]	25.77	–	–	29.22
Encoder-Decoder (BiLSTM) + Global + SpaCy (Proposed)	47.50	**67.45**	**51.72**	**66.76**
Encoder-Decoder (BiLSTM) + Local + SpaCy (Proposed)	46.69	66.62	50.45	**66.77**
Encoder-Decoder (BiLSTM) + Bahdanau + SpaCy (Proposed)	47.26	67.36	51.36	**66.60**

Bold indicates higher value in the column

for Rouge-1, Rouge-2 and Local attention for Rouge-L. Also, the proposed model performed well with transformer-based dynamic SpaCy embedding and improves Rouge-1, Rouge-2 and Rouge-L scores as compared to other approaches. In terms of Bleu score, the performance of the proposed model is enhanced when compared to RNN based encoder-decoder model and GRU based encoder-decoder model with attention. Also, the performance of the proposed model is better for long sentences as compared to short sentences.

In future work, the performance of model can further be enhanced by making it even more contextually rich. The impact of sentence length can also be explored in more detail because this model is more efficient with longer sentences. Moreover, to advance research on this topic there is limited availability of datasets. Future work can also be extended on the creation of new datasets which would help to make such models more robust and efficient. Further, this task can also be extended by performing experiments with transformer models.

Funding All the co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report.

Data availability All data generated or analyzed during this study are included in this article.

Declarations

Conflict of interest The authors have no conflicts of interest to declare.

Ethical approval Not applicable.

References

1. Akula R, Garibay I (2021) Interpretable multi-head self-attention architecture for sarcasm detection in social media. Entropy. 23(4):394
2. Kapoor KK, Tamilmani K, Rana NP, Patil P, Dwivedi YK, Nerur S (2018) Advances in social media research: past, present and future. Inf Syst Front. 20:531–558
3. Ortega-Bueno R, Rosso P, Pagola JEM (2022) Multi-view informed attention-based model for Irony and Satire detection in Spanish variants. Knowl-Based Syst. 235:107597
4. Kumar A, Narapareddy VT, Srikanth VA, Malapati A, Neti LBM (2020) Sarcasm detection using multi-head attention based bidirectional LSTM. IEEE Access. 8:6388–6397
5. Poria S, Hazarika D, Majumder N, Mihalcea R (2020) Beneath the tip of the iceberg: current challenges and new directions in sentiment analysis research. IEEE Trans Affect Comput. 14(1):108–132
6. Birjali M, Kasri M, Beni-Hssane A (2021) A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowl-Based Syst. 226:107134
7. Touahri I, Mazroui A (2021) Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied sarcastic features. Knowl-Based Syst. 227:107232
8. Peled L, Reichart R. Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation. arXiv preprint arXiv:170406836. 2017
9. Dubey A, Joshi A, Bhattacharyya P. Deep models for converting sarcastic utterances into their non sarcastic interpretation; 2019. p. 289–292.
10. Gupta R, Kumar J, Agrawal H. A statistical approach for sarcasm detection using Twitter data. IEEE; 2020. p. 633–638.
11. Srinu N, Sivaraman K, Sriram M (2024) Enhancing sarcasm detection through grasshopper optimization with deep learning based sentiment analysis on social media. Int J Inf Technol. https://doi.org/10.1007/s41870-024-02057-9
12. Pandey R, Kumar A, Singh JP, Tripathi S (2021) Hybrid attention-based long short-term memory network for sarcasm identification. Appl Soft Comput. 106:107348
13. Fkih F, Rhouma D, Alghofaily H (2024) A semantic approach for sarcasm identification for preventing fake news spreading on

social networks. Int J Inf Technol. https://doi.org/10.1007/s41870-024-02156-7
14. Potamias RA, Siolas G (2020) Stafylopatis AG A transformer-based approach to irony and sarcasm detection. Neural Comput Appl. 32(23):17309–17320
15. Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al. Moses: Open source toolkit for statistical machine translation. Association for Computational Linguistics; 2007. p. 177–180.
16. Cui J (2022) Respecting the old and loving the young: emoji-based sarcasm interpretation between younger and older adults. Front Psychol 13:897153
17. Ghosh D, Guo W, Muresan S. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In: proceedings of the 2015 conference on empirical methods in natural language processing; 2015. p. 1003–1012.
18. Rajadesingan A, Zafarani R, Liu H. Sarcasm detection on twitter: A behavioral modeling approach. In: Proceedings of the eighth ACM international conference on web search and data mining; 2015. p. 97–106.
19. Castro S, Hazarika D, P´erez-Rosas V, Zimmermann R, Mihalcea R, Poria S. Towards multimodal sarcasm detection (an obviously perfect paper). arXiv preprint arXiv:190601815. 2019;.
20. Misra R, Arora P. Sarcasm detection using hybrid neural network. arXiv preprint arXiv:190807414. 2019;.
21. Babanejad N, Davoudi H, An A, Papagelis M. Affective and contextual embedding for sarcasm detection. In: Proceedings of the 28th international conference on computational linguistics; 2020. p. 225–243.
22. Johnson AA, Karthik R. Performance Evaluation of Word Embeddings for Sarcasm Detection-A Deep Learning Approach. In: Proceedings of the First International Conference on Advanced Scientific Innovation in Science, Engineering and Technology, ICASISET 2020, 16–17 May 2020, Chennai, India;2021. .
23. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.
24. Dubey K, Nair R, Khan MU, Shaikh S. Toxic comment detection using lstm. IEEE; 2020. p. 1–8.
25. Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:150804025. 2015
26. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014
27. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014