Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-023-01570-7

ORIGINAL RESEARCH

Leveraging attention layer in improving deep learning models performance for sentiment analysis

Monir Yahya Salmony[1] · **Arman Rasool Faridi**[1] · **Faraz Masood**[1]

Received: 13 June 2023 / Accepted: 26 September 2023

Abstract Sentiment analysis (SA) is a rapidly expanding research field, making it difficult to keep up with all of its activities. It aims to examine people's feelings about events and individuals as expressed in text reviews on social media platforms. Recurrent neural networks (RNN) have been the most successful in the past few years at dealing with sequence data for many natural language processing (NLP) tasks. These RNNs suffer from the problem of vanishing gradients and are inefficient at memorizing long or distant sequences. The recent attention strategy successfully addressed these issues in many NLP tasks. This paper aims to leverage the attention mechanism in improving the performance of the models in sentiment analysis on the sentence level. Vanilla RNN, long short-term memory, and gated recurrent unit models are used as a baseline to compare to the subsequent results. Then, an attention layer was added to the architecture blocks, where the encoder state reads and summarizes the sequential data. This layer provides weights to the summarized portion so that the decoder state can translate it more accurately and the model can make more accurate predictions. Under the same parameter settings, the integrated attention approach is evaluated and compared to the baseline models. The experimental results show that combining attention with these models can increase overall performance by a good margin in the suggested evaluation metrics compared to other works; it will help to enhance the efficiency of the decision-making.

Keywords Sentiment analysis · Deep learning · RNN · LSTM · GRU · Attention

1 Introduction

Natural language processing (NLP) stands as a sub-field of artificial intelligence [1, 2], with the objective of improving computers' comprehension of human text and speech. Text classification is a crucial aspect of NLP. It aims to map labels to text [3, 4]. It has numerous applications, such as subject labeling, sentiment categorization, and spam detection [5, 6]. Sentiment Analysis (SA) is one of the fastest-growing NLP application research fields. It evaluates the user's behavior, represented by textual data available nearly everywhere on social media platforms, by determining whether it is positive (*Pos*) or negative (*Neg*) with a particular event [7, 8] and can be done at three levels, i.e., document, sentence and phrase [9, 10].

Traditional approaches to SA represent documents using sparse lexical features, such as bag of words (BOW), term frequency-inverse document frequency (TF-IDF) and one hot encoding [4], followed by linear model algorithms [8]. Applying these feature representations end up with sparse matrix, i.e., vectors of high dimensionality (containing many zeros), which could cause for decreasing the performance. Additionally, these representations do not consider the semantics of the words. So words like aircraft and airplane are handled as two different features. While they have a very similar meaning. These issues addressed using word embedding representation techniques, which transforms words in

✉ Monir Yahya Salmony
salmony22@gmail.com

Arman Rasool Faridi
arman.faridi@gmail.com

Faraz Masood
ffarazs@gmail.com

[1] Department of Computer Science, Faculty of Science, Aligarh Muslim University, Civil Lines, Aligarh 202001, India

Published online: 28 October 2023

a vocabulary to vectors of continuous real numbers with much lower dimensionality and also taking into account the semantic relationships between words that are reflected in the distance and direction of the vectors [5]. Therefore, the representation of words has become the basis for the development of various tasks in research [6].

Deep learning (DL) techniques, primarily recurrent neural networks (RNNs), have been the most effective in recent years for dealing with sequence information for many NLP tasks, such as question answering, topic classification [11] and anomaly detection [6]. Although RNN-based techniques have been effective for text categorization, they are susceptible to vanishing gradients and have difficulties memorizing long or distant sequences [11–13].

The attention mechanism addressed previous problems by permitting a model to directly examine and derive from an earlier sentence point state. Recently, attention has been used to a combination of DL models [14], including convolutional neural networks (CNN) [15] and RNN models [12, 16]. The attention-based DL models perform well in prediction and have interpretability. Applying attention to SA, the model can automatically identify important words by assigning higher weights to the sentiment parts of the input that are more important and low weights to the irrelevant information for the sentiment classification task [17]. Without using the attention mechanism, the RNN would have to process the entire sentence in one pass, which could be slow and may not allow the model to effectively learn long-term dependencies. On the other hand, the model with attention can selectively focus on certain parts of the input sentence as it processes it [12, 18].

Thus, this work presents an in-depth study on the performance of RNNs models, namely, Vanilla RNN (V-RNN), long short-term memory (LSTM), and gated recurrent unit (GRU) networks with/without attention approach and different features representation methods, i.e., Glove and FastText for SA on benchmark dataset. It also provides a comprehensive comparison between the various models and features representation methods based on the suggested evaluation measures to understand their effectiveness in SA task. The main contributions of this paper are as follows:

- Present a comprehensive evaluation of several RNN models, specifically V-RNN, LSTM, and GRU, and examine their performance in the context of SA.
- Compare the performance of these models using two popular word embedding techniques, namely, FastText and Glove. This provides valuable insights into the effectiveness of these methods in improving the performance of RNNs in SA.
- Examine the impact of leveraging attention method on the performance of these models by comparing the baseline RNNs (without attention) with the same RNNs plus the attention mechanism in all the feature embedding methods. This allows for a thorough evaluation of the efficiency of this approach in improving the performance of RNNs in SA.
- Perform the comparative analysis on a benchmark reviews dataset from three different domains, providing a solid foundation for the conclusions drawn in this study.

This research study has been structured as follows: Sect. 2 presents the 'Related works' that explore various approaches used in SA, followed by Sect. 3, which presents the components used in the study. Section 4 covers the experiments details. The results and discussion are presented in Sects. 5 and 6, where the proposed approach performance is analyzed and compared to the baseline. Finally, the work concludes with a summary and outlines potential areas of future work.

2 Related work

Various approaches have been used to tackle the SA task, like traditional machine learning (ML), DL, and attention-based DL, focusing solely on recent and relevant articles in order in this section.

In terms of traditional ML approaches, various works have delved into the realm of text sentiment analysis (SA) with differing focuses and methodologies. One research study [19] conducted an investigation into the impact of text representations on the performance of multiple ML models for SA, utilizing IMDB and Twitter data. The study found that artificial neural networks exhibited the highest accuracy in both datasets compared to alternative models. Another work [20] undertook a comprehensive experimental investigation of SA using various well-known ML algorithms. They assessed three famous review datasets acquired from different domains, including Yelp, Amazon, and IMDb data for products, restaurants, and movies. The findings highlighted the superiority of maximum entropy and boosting over other algorithms on various evaluation measures. In a separate study [21], a comparison between Naive Bayes (NB) and support vector machine (SVM) classifiers was carried out, along with TF-IDF vectorizers for sentiment classification. By employing a grid search process to optimize parameters, the study determined that SVM slightly outperformed NB on Yelp and IMDB datasets, while NB showed better results on Amazon reviews. Similarly, [8] used same algorithms plus Logistic Regression (LR) with BOW and TF-IDF feature representations as baselines and employed negation scope identification methods to enhance the classifiers performance to classify the sentiment of twitter data. Also, [22] explore the same methods plus Decision Tree (DT) algorithm with N-gram (N = 1, 2, and 3) and both BOW and TF-IDF feature representations. The performance

of these models is compared with the fine-tuned BERT transformer model on fake and real news dataset and demonstrated that the traditional models are still good candidates and that the use of bigram combined with BOW and DT classifier performs comparable results to BERT with an accuracy of 99.74%. In addition to SVM and NB, [23] added Random Forest (RF) and K-nearest neighbor (KNN) with only TF-IDF representation method. These models evaluated on a combined dataset from different domains. The experiments revealed that RF performs the best among others. An innovative approach [24] proposed a hybrid feature selection technique for SA, involving a genetic algorithm and a combination of information gain, chi-squared, and GINI index methodologies. Additionally, an ensemble technique based on error rates across diverse domains was introduced. These methods were evaluated using four SVM versions on the same dataset as the previous study [21], demonstrating their efficacy across all dataset domains. Further enhancements were made by [25] using a similar dataset and incorporating the Reuters dataset. This study [26] introduced an Adaptive Ensemble classifier for SA, incorporating drift detection approach. This approach leverages the identified false-positive drift detections to mitigate their adverse effects. The evaluation is done on the same dataset of this study and showed that integrating drift can enhance the performance of traditional ML algorithms.

Shifting to the DL approaches, [27] combined RNN, LSTM, and GRU models with a CNN output, fed by the Glove embedding layer, for SA in movie reviews. The integration of these models, along with parameter optimization, led to enhanced accuracy. In [28] researchers developed an ensemble model comprising LSTM and CNN components, with one capturing temporal information and the other extracting local structure. This approach outperformed individual models on IMDB and SST datasets, achieving notably higher accuracy compared to earlier studies. Addressing accuracy improvement through DL, [5] focused on LSTM and word embedding, conducting extensive experiments across seven benchmark datasets with varying classes and training samples. The results showcased superior performance of LSTM under specific parameter settings compared to existing literature. For effective sentiment classification, [29] introduced a novel two-state GRU and encoder method to preprocess data and amplify the impact of word embedding. This model exhibited higher accuracy than GRU, LSTM, and Bi-LSTM recurrent models on IMDB and Amazon reviews, with Word2Vec outperforming other embedding methods. Another study [30] explored various RNN algorithms paired with word embedding strategies—word2vec, Glove, and FastText via Skip-grams—evaluated on the Amazon dataset. GLRNN techniques with FastText achieved exceptional accuracy of 93.75% for the unbalanced dataset, while the balanced dataset saw the LRNN algorithm attain a peak accuracy of 88.39%. Similarly, [31] presented experimental results obtained by conventional DL architectures and word embedding schemes. Also, proposed an architecture combines TF-IDF weighted Glove word embedding with CNN-LSTM architecture that outperforms the conventional DL models for detecting the sentiment on product reviews obtained from Twitter data.

Transitioning to DL with attention mechanisms, [32] argued that incorporating sentiment lexicon embedding enhances word representation accuracy. Additionally, an attention vector positioning method was developed for general SA without a target, bolstering LSTM's ability to capture global sentiment information and perform comparable results. Building upon this, [33] investigated the efficacy of the GRU network in addressing vanishing gradient and exploding problems inherent in traditional RNNs. A two state GRU with an attention mechanism was devised, selecting the most informative features for SA and improving accuracy while reducing information loss. Another approach [34] introduced a model utilizing multi-attention mechanisms—word attention, local attention, and cross attention—to accumulate textual dependencies. Mutual information was employed for data augmentation, mitigating overfitting concerns. The proposed solution exhibited robust performance across various sentiment datasets. In study [14] the limitations of current DL models for text SA were highlighted—they often focus narrowly on words and sentences, disregarding the impact of emotions on sentiment feature extraction. A remedy was presented in the form of an LSTM network that integrated emotional intelligence with attention mechanisms, resulting in enhanced performance.

In conclusion, previous research has shown that ML approaches have been trained and tested independently on datasets from different domains, leading to inconsistent results. Furthermore, DL models have been found to exhibit different behaviors when combined with different feature representation techniques. The recent success of attention mechanisms in other NLP tasks highlights the need for further evaluation of the effectiveness of attention in weighing important words in short review text, specifically in a broader range of datasets from diverse domains. This study aims to address this gap in research and provide a comprehensive evaluation of the effectiveness of attention mechanisms in SA.

3 Methodology work flow

3.1 Proposed work components

The proposed approach for enhancing sentiment classification performance involves several key components: data collection, data preparation, embedding features representation

(using Fasttext and Glove), modeling (using V-RNN, LSTM and GRU), attention layer, and evaluation. These components are discussed in detail in the following sections.

3.1.1 Dataset description and preparation

This study focused on determining whether a given text expresses *Pos* or *Neg* sentiment. For that, a publicly available dataset of information extracted from user reviews of three popular review websites: imdb.com, amazon.com, and yelp.com, is used. There is ~ 3000 samples total, with 1500 *Pos* and 1500 *Neg* sentiment sentences. Each entry has two components: a review and label (0 or 1) indicating the entry's tone, (*Neg* or *Pos*) respectively [35]. Then, the text reviews are prepared by tokenizing and truncating or padding them to a fixed length.

3.1.2 Embedding layer

Generally, neural networks are composed of linear algebra operators and nonlinear activation functions. For these computations, the input text must be encoded as a vector of numbers (convert text into numerical representations in a low dimensional space) [18, 36]. This work utilizes Glove and FastText.

1. **Glove** is a type of unsupervised learning that acquires representations of word vectors [30]. It combines the advantages of latent semantic analysis and Word2vec, while also enhancing the speed of parameter training and the ability to scale effectively. This makes it a suitable choice for processing large corpus datasets [37].
2. **FastText** is another approach that utilizes morphological features to identify difficult words, making it an appropriate choice for representing vectors. This ability increases its generalizability as well. It generates vectors based on n-grams, which facilitates handling unknown words [38, 39].

3.1.3 Recurrent neural network (RNN) models

These neural network types are particularly suited for sequential data, such as time series or natural language. In these networks, the earlier step's output is fed as data to the current timestamp. In simple neural networks, the inputs and outputs are not interdependent and do not have any direct influence on each other. In contrast, the key feature of RNNs is that they maintain an internal state that can be updated at each time step, allowing them to keep information about past inputs, which helps in situations such as predicting the upcoming word of a sentence [30]. RNNs can be categorized into: one-to-one (also known as feedforward) and many-to-many (also known as sequence-to-sequence).

3.1.3.1 Vanilla recurrent neural network (V-RNN)

V-RNN is the simplest type of RNN. It consists of a single recurrent layer that receives the present input and the preceding hidden state as inputs and produces an output and a new hidden state. It is called "vanilla" because it is the simplest form of RNNs, without additional mechanisms such as gates [30, 31].

3.1.3.2 Long short-time memory (LSTM)

LSTM network is a more advanced version of the V-RNN network, which incorporates memory cells to prevent loss of information when handling data in a sequence [5]. It can manage long-term dependencies. It processes the data sequence by employing gate vectors at each position to regulate the data flow along the sequence. At each time step, there is a vector set containing various gates, such as input, forget, output, and memory gates. These are combined to determine the output of the hidden layer [31, 40].

3.1.3.3 Gated recurrent unit (GRU)

GRU is a contemporary version of RNN that performs similarly to LSTM but has a less intricate design and is more straightforward to implement. Unlike LSTM, it doesn't have a cell state, instead, it employs a hidden layer to pass information. The hidden layer produces the present time step value by merging its input with the hidden layer state from the prior one [13, 30, 31].

3.1.4 Attention layer

Although RNNs have been a tremendous success, their issue involves gradients disappearing over time, which makes them ineffective at retaining information from lengthy or faraway sequences. The attention mechanism addressed previous issues by allowing a model to directly examine and derive from the condition of an earlier sentence point. It has the ability to retrieve all past states and evaluate their importance using a learned metric with respect to the current token can offer more detailed insights about far-off relevant tokens [12, 18]. The dot product attention approach [41, 42] is used in this work. It is used in this work to weigh different parts of the input differently and help the model to focus on essential elements while making predictions. It works by computing the dot product between the input and output of a model at each time step, taking into account the context of the previous time steps, which generates a weight that is then used to weigh the input and output of the model. That permits the model to focus on the most relevant information. It can be represented using the following steps:

- Compute score $(h_{-t}, x_{-t}) = h_{-t} * W * x_{-t}$, where h_{-t} is the output, x_{-t} is the input at timestamp t, and W is a trainable weight matrix.

- Obtain attention weight by applying softmax activation on the score tensor (alpha_t = softmax(score))
- Calculate context vector by element-wise multiplication of input and attention weight ($c_t = alpha_t * x_t$)
- The attention output is obtained by concatenating the context vector and output layer (h_t)
- The attention vector is obtained by passing attention output through a dense layer with *tanh* activation (attention_vector = *tanh* (W * attention output))

This method is different from other attention mechanisms like additive attention, multiplicative attention, scaled dot-product attention, etc., in how it calculates the attention weight. This mechanism uses the dot product of the input and output to calculate the attention weight, while other attention mechanisms use different mathematical operations. Also, the dot product attention mechanism produces better results and computes faster than other attention mechanisms [41].

3.1.5 Performance evaluation

This is used to determine the performance in each scenario relative to the original label of the dataset and to compare them to one another [7, 8]. The performance is calculated using accuracy (AC), precision (PR), recall (RC), and f1-score (F1) methods. These methods are calculated using the functions available in the Python Scikit-learn Metrics module [30, 43].

3.2 Algorithm work flow

	***Input** : Text review, Labels*
	***Output**: Model Evaluation (Accuracy, Precision, Recall, F1-score)*
	Begin
1.	***STEP 1: Preparing***
2.	*Tokenize and truncate/pad reviews;*
3.	*Choose hyperparameter for fixed length;*
4.	***STEP 2: Word Embeddings***
5.	*Load desired word embeddings (Glove/FastText);*
6.	*Create embedding matrix for initializing weights;*
7.	***STEP 3: Define Model Architecture***
8.	***STEP 3.1: Define input layer***
9.	*Input tensor shape (max_len);*
10.	***STEP 3.2: Define Embedding Layer***
11.	*Embedding layer maps word index to dense vector of size embedding dim;*
12.	*Embedding layer initialized with chosen word embeddings and set as not trainable;*
13.	***STEP 3.3: Define Deep Learning Layer***
14.	*Pass previous output through model layer (V_RNN/GRU/LSTM) with specified units;*
15.	*Obtain 3D tensor of shape (batch_size, time_steps, 100);*
16.	***STEP 3.4: Define Attention/Dense Layer***
17.	*Pass 3D tensor through attention layer or dense layer;*
18.	***If** attention:Compute weights between input and output (dot product attention)* ***Else:** Use dense layer with specified units and activation function;* ***End If***
19.	***STEP 3.4: Define Dropout Layer***
20.	*Pass output of attention layer or dense layer through dropout layer with specified rate;*
21.	***STEP 3.5: Define Prediction Layer***
22.	*Pass output of dropout layer through dense layer with single unit and sigmoid activation;*
23.	***STEP 4: Model Compilation***
24.	*Use binary_crossentropy loss function;*
25.	*Use Adam optimizer;*
26.	*Compile model;*
27.	***STEP 5: Model Training***
28.	*Set batch size*
29.	*Set epochs;*
30.	*Fit (train) model using training data with the specified batch size and epochs;*
31.	***STEP 6: Evaluation***
32.	*Evaluate models on testing data;*
33.	*Plot Accuracy, Precision, Recall, F1-score;*
	End

4 Implementation

This study focuses on classifying the reviews as *Pos* or *Neg* based on the sentiment expressed. The proposed architecture is based on RNNs with embedding layers and the dot attention mechanism. The dataset that contains reviews from three popular websites, i.e., imdb, yelp, and amazon, is used to evaluate the models' performance and generalization. The architecture consists of five elements: the initial layer for receiving input, a layer for embedding, a layer for RNN processing, a layer for attention, and a final output layer. Before the output layer, a dropout strategy was applied to prevent overfitting. The output layer calculates the loss using the binary cross-entropy function, and classification is done using a sigmoid function. The steps followed in the experiments were:

1. Prepare the reviews by tokenizing and truncating or padding them to a fixed length.
2. Load pre-trained word embeddings (FastText or Glove) with 300 dimensions and create an embedding matrix to initialize the weights of the embedding layer. The reason for using pre-trained embeddings is that they provide a rich representation of the words learned from a large corpus, which improves the model's performance.
3. Define the model architecture: input tensor is passed through an embedding layer and then an RNN layer (V-RNN, LSTM, or GRU) with 100 units. The embedding layer is initialized with pre-trained embeddings and set not to be trainable; this helps to prevent overfitting and improves the model's generalization ability.
4. Apply attention mechanism (dot product) when attention is used or a dense (feedforward) layer with 350 units and *Relu* activation when attention is used. The attention mechanism is applied to weigh different input parts differently and help the model focus on essential parts while making predictions.
5. Add a *dropout* layer (0.5) to prevent overfitting.
6. Pass the output through a dense layer with a single unit and *sigmoid* activation function to produce a prediction for each review.
7. Compile the model using binary cross-entropy function and Adam as the optimizer.
8. Fit the model using the reviews and labels as training data with a batch size of 64 and 10 epochs.
9. Evaluate the model's performance on the test set by utilizing the four evaluation measures that were previously discussed.

The use of word embeddings, attention mechanism, and dropout layer are crucial to improve the model's performance and generalization ability.

5 Results

The results of our experiments have been carefully analyzed and presented clearly and concisely to allow for easy comparison and interpretation. We have employed Excel software to create comparison charts, demonstrating the impact of the attention method on the performance of the RNN models. Table 1 demonstrates the quantitative performance of each model in terms of the suggested performance measures with and without the attention mechanism. Furthermore, Figs. 1 and 2 provide a visual representation of the overall performance of the classifiers, allowing for a quick and easy comparison of the models with and without attention. These results provide valuable insights into the efficiency of the attention method in improving the RNN models' performance in SA.

Table 1 Attention impact on the performance of V-RNN, LSTM, and GRU classifiers

Model		Attention	Accuracy	Precision	Recall	F1-score
V-RNN	FastText	✗	76.13	80.58	72.18	75.3
		✓	**82.97**	85.2	80.28	82.19
LSTM		✗	84.13	85.9	81.45	83.24
		✓	**84.28**	83.83	86.31	82.3
GRU		✗	84.57	86.38	82.36	83.91
		✓	**85.74**	87.81	82.78	84.86
V-RNN	Glove	✗	81.95	81.00	85.07	82.48
		✓	**85.74**	86.75	85.31	85.62
LSTM		✗	84.28	82.63	89.18	85.27
		✓	**86.61**	87.4	87.46	86.93
GRU		✗	86.17	86.35	88.28	86.58
		✓	**86.46**	89.28	84.2	86.16

Bold values indicate better results than others

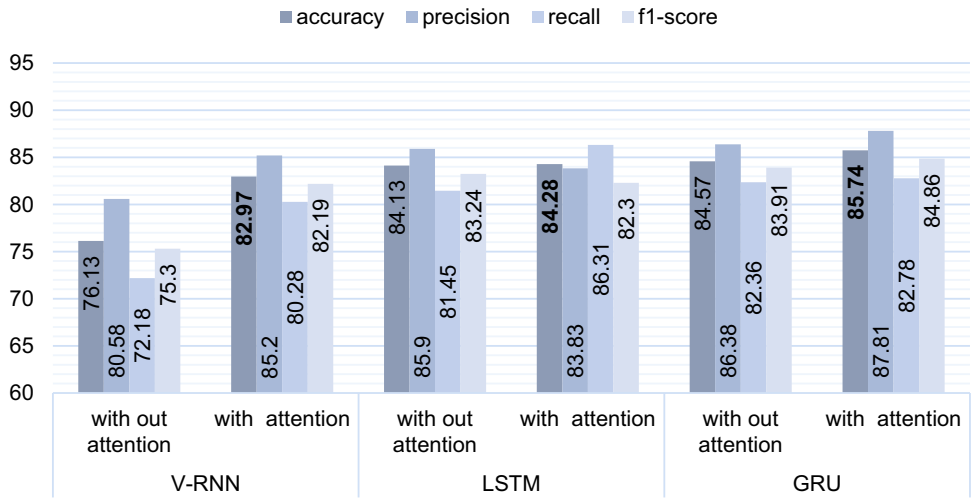

Fig. 1 Models' performance with and without attention using FastText embedding on the validation dataset

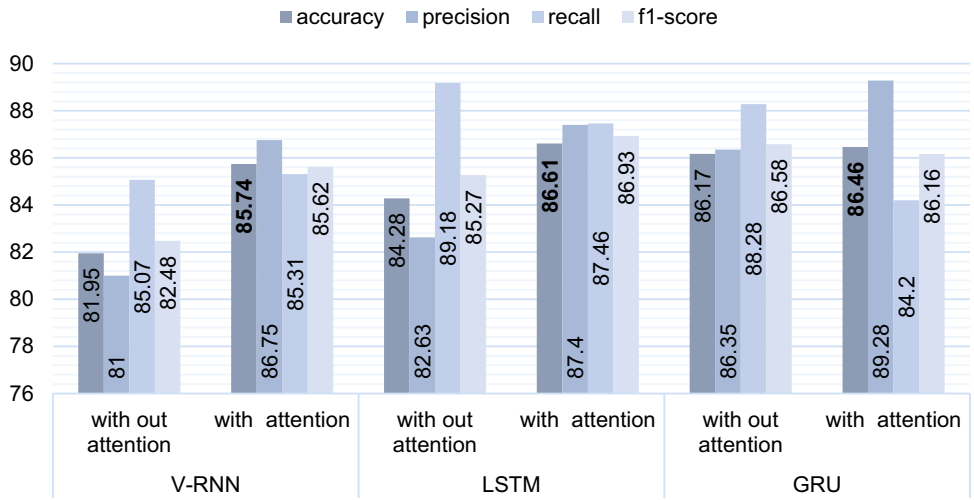

Fig. 2 Models' performance with and without attention using Glove embedding on the validation dataset

Table 1 (upper part) and Fig. 3 show the results of the V-RNN, LSTM, and GRU (with and without attention) classifiers with FastText embedding on the validation dataset. It is demonstrated that employing the attention approach enhances the performance of the models on all the evaluation metrics by high (~6%) and good (~1%) accuracy margins for V-RNN and GRU, respectively. In contrast, utilizing attention with LSTM enhances the model's accuracy slightly by (~0.15%) but improves the recall by (~5).

Table 1 (lower part) and Fig. 1 show the results of the V-RNN, LSTM, and GRU (with and without attention) classifiers with Glove on the validation dataset. It is demonstrated that employing the attention approach enhances the performance of the models on all the evaluation metrics by high (~4%) and good (~2%) accuracy margins for V-RNN and LSTM, respectively. In contrast, utilizing attention with GRU enhances the models' accuracy slightly but improves the precision improved by (~3%).

6 Discussion and comparison

The experimental results highlighted the effectiveness of combining RNNs with the attention approach and Glove and FastText embeddings in sentiment analysis. Firstly, when attention and FastText embedding were employed on RNNs, the results showed that the baseline methods, such as V-RNN, LSTM, and GRU, without attention mechanisms, reported the lowest performance on nearly all metrics. However, the use of the dot attention approach significantly enhance the performance of the baseline models, with the best accuracy achieved by the GRU model at 85.74%.

Secondly, similar results were observed when attention and Glove embedding were employed. The baseline methods without attention mechanisms reported the lowest performance, whereas utilizing the attention approach enhance the performance of the same models by a good margin. Additionally, Glove embedding resulted in even

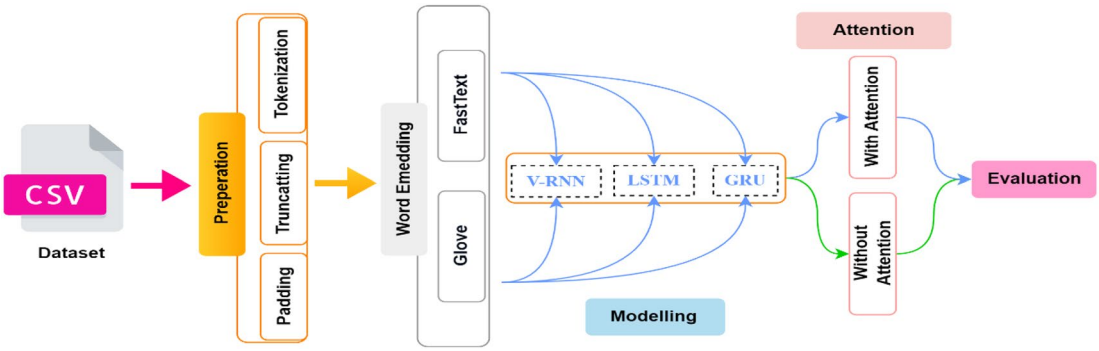

Fig. 3 Proposed work components

Table 2 Models performance comparison

Paper	Model	Features	Accuracy	Precision	Recall	F1-score	Notes
[26]	NB	–	74	75	74	75	Ensemble based on the Drift detection algorithm
[8, 21]	SVM	Uni-gram + TF-IDF	82	82	82	82	Re-implemented
	NB		64	64	64	64	
[8]	LR		82	82	82	82	
[23]	KNN		76	76	76	76	
	RF		77	77	77	77	
[22]	DT		70	70	70	70	
[8, 21]	NB	Uni-gram + BOW	63	63	63	63	
	SVM		81	81	81	81	
[8]	LR		81	81	81	81	
[22]	DT		74	74	74	74	
[30, 31]	Baseline Lstm	Glove	84.28	82.63	89.18	85.27	
	Baseline GRU		86.17	86.35	88.28	86.58	
	Baseline Lstm	FastText	84.13	85.9	81.45	83.24	
	Baseline GRU		84.57	86.38	82.36	83.91	
Ours	LSTM + attention	Glove	**86.61**	87.4	87.46	86.93	
	GRU + attention		**86.46**	89.28	84.2	86.16	

Bold values indicate better results than others

better results than FastText embedding, specifically for the V-RNN and GRU models without attention. The attention approach further enhanced the results, particularly for the LSTM model with Glove embedding, which achieved an accuracy of 86.61%.

The experiments revealed that leveraging the attention method can significantly boost the models' accuracy, especially when combined with Glove embeddings. Additionally, the results also indicated that the effectiveness of the attention approach is affected by the embedding type, with Glove embeddings resulting in better performance gains than FastText embeddings. It may be due to the fact that the length of sentences in the dataset is small and cannot find many n-gram character patterns. Furthermore, the results showed that GRU with Glove embeddings achieved nearly similar accuracy and better precision than LSTM with Glove embeddings, making it a better choice due to the advantages of GRU over LSTM.

Additionally, for comparison purposes, prior research has outlined diverse strategies employed to address SA, Table 2. To contextualize our work within this framework, some of which have been re-implemented in this paper for a fair comparison, we proceed with the following analysis.

Firstly, upon implementing the baseline methodologies from [30, 31], it becomes evident that the utilization of Glove feature extraction in conjunction with the GRU-based

RNN model led to the highest accuracy of 86.17%. Subsequently, accuracy values of 84.57%, 84.28%, and 84.13% were attained for GRU with FastText, LSTM with Glove, and LSTM with FastText, respectively. In contrast, FastText feature extraction exhibited comparatively inferior results across all RNN algorithms, with the least outcome attributed to [31]'s and Table 1 VRNN + FastText combination, resulting in an accuracy of 76.13%. Moving forward, a comparison against both [8, 21–23, 26] methodologies is presented. The Naïve Bayes approach with BOW and TF-IDF + unigrams feature extraction [8, 21] yielded the lowest accuracies, registering 63.00% and 64.00% accuracy, respectively. On a contrasting note, the SVM algorithm and LR [8] showcased same accuracy levels of 82.00% when utilizing the TF-IDF + unigrams feature extraction method.

In summary, this research highlights the superior performance of the proposed LSTM and GRU architectures, combined with Glove feature extraction and attention mechanisms. These models consistently outperformed other methods assessed in the study. Overall, the experiments demonstrate that incorporating attention in RNNs enhances sentiment analysis performance, with Glove embeddings proving more effective than FastText embeddings for this task.

7 Conclusion

RNNs have been widely used and have shown great success in dealing with sequence data for various NLP tasks in recent years. However, these models are known to suffer from the problem of vanishing gradients and are not always capable at memorizing distant sequences. To address the previous issues, the attention mechanism has been introduced and proven successful in many NLP tasks. This paper aims to leverage the dot product attention mechanism to enhance the performance of V-RNN, LSTM, and GRU models in SA on the sentence level. Then a comparison of the proposed models to the baseline models (without attention) is presented to see the performance improvement. The experimental results have shown that combining attention with these models can significantly increase the performance in the suggested evaluation metrics. Consequently, it highlighted the importance of dot product attention in addressing the limitations of RNNs and improving the performance in SA. Additionally, it was found that Glove embedding produced better results than the FastText method. Current DL models for text SA are limited in their focus, primarily concentrating on individual words and sentences, ignoring the modulating effect of emotions on sentiment feature extraction. This aspect needs to be thoroughly investigated and incorporated into future research endeavors. This can be achieved through the exploration of various attention mechanisms, feature representation techniques like BERT and ELMo, and diverse deep learning approaches. Evaluating the efficacy of these techniques holds the potential to significantly enhance performance.

Data availability Not applicable.

References

1. Nadkarni PM, Ohno-Machado L, Chapman WW (2011) Natural language processing: an introduction. J Am Med Inform Assoc 18(5):544–551. https://doi.org/10.1136/amiajnl-2011-000464
2. Lauriola I, Lavelli A, Aiolli F (2022) An introduction to deep learning in natural language processing: models, techniques, and tools. Neurocomputing 470(xxxx):443–456. https://doi.org/10.1016/j.neucom.2021.05.103
3. Li G, Zhao X, Wang X (2022) Quantum self-attention neural networks for text classification, pp 1–18. [Online]. http://arxiv.org/abs/2205.05625
4. Barman D, Chowdhury N (2020) A novel semi supervised approach for text classification. Int J Inf Technol 12(4):1147–1157. https://doi.org/10.1007/s41870-018-0137-9
5. Adamuthe AC (2020) Improved text classification using long short-term memory and word embedding technique. Int J Hybrid Inf Technol 13(1):19–32. https://doi.org/10.21742/ijhit.2020.13.1.03
6. Khan W, Haroon M (2022) An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks. Int J Cogn Comput Eng 3(August):153–160. https://doi.org/10.1016/j.ijcce.2022.08.002
7. Ali Salmony MY, Rasool Faridi A (2021) Supervised sentiment analysis on amazon product reviews: a survey. In: Proceedings of the 2021 2nd international conference of intelligent engineering and management ICIEM 2021, no. June, pp 132–138. https://doi.org/10.1109/ICIEM51511.2021.9445303
8. Ali Salmony MY, Faridi AR (2021) An enhanced twitter sentiment analysis model using negation scope identification methods. In: Proceedings of the 2021 8th international conference on computing for sustainable global development INDIACom 2021, pp 864–869. https://doi.org/10.1109/INDIACom51348.2021.00155
9. Vashisht G, Sinha YN (2021) Sentimental study of CAA by location-based tweets. Int J Inf Technol 13(4):1555–1567. https://doi.org/10.1007/s41870-020-00604-8
10. Gupta I, Chatterjee I, Gupta N (2023) A two-staged NLP-based framework for assessing the sentiments on Indian supreme court judgments. Int J Inf Technol 15(4):2273–2282. https://doi.org/10.1007/s41870-023-01273-z
11. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2020) Deep learning based text classification: a comprehensive review, vol. 54, no. 3. [Online]. http://arxiv.org/abs/2004.03705
12. Kardakis S, Perikos I, Grivokostopoulou F, Hatzilygeroudis I (2021) Examining attention mechanisms in deep learning models for sentiment analysis. Appl Sci. https://doi.org/10.3390/app11093883
13. Sachin S, Tripathi A, Mahajan N, Aggarwal S, Nagrath P (2020) Sentiment analysis using gated recurrent neural networks. SN Comput Sci. https://doi.org/10.1007/s42979-020-0076-y
14. Huang F, Li X, Yuan C, Zhang S, Zhang J, Qiao S (2022) Attention-emotion-enhanced convolutional LSTM for sentiment analysis. IEEE Trans Neural Netw Learn Syst 33(9):4332–4345. https://doi.org/10.1109/TNNLS.2021.3056664

15. Chen K, Wang J, Chen L-C, Gao H, Xu W, Nevatia R (2015) ABC-CNN: an attention based convolutional neural network for visual question answering. [Online]. http://arxiv.org/abs/1511.05960
16. Zhou P et al (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: 54th annual meeting of the Association for Computational Linguistics ACL 2016—short paper, pp 207–212. https://doi.org/10.18653/v1/p16-2034
17. Letarte G, Paradis F, Giguère P, Laviolette F (2018) Importance of self-attention for sentiment analysis. In: EMNLP 2018—2018 EMNLP work. BlackboxNLP analyzing and interpreting neural networks NLP, Proceedings of the 1st workshop, pp 267–275. https://doi.org/10.18653/v1/w18-5429
18. Yu Y, Liu G, Yan H, Li H, Guan H (2018) Attention-based Bi-LSTM model for anomalous HTTP traffic detection. In: 2018 15th international conference on service systems and service management (ICSSSM), July 2018, pp 1–6. https://doi.org/10.1109/ICSSSM.2018.8465034
19. Basarslan MS, Kayaalp F (2020) Sentiment analysis with machine learning methods on social media. ADCAIJ Adv Distrib Comput Artif Intell J 9(3):5–15. https://doi.org/10.14201/adcaij202093515
20. Kabir M, Kabir MMJ, Xu S, Badhon B (2021) An empirical research on sentiment analysis using machine learning approaches. Int J Comput Appl 43(10):1011–1019. https://doi.org/10.1080/1206212X.2019.1643584
21. Arora A, Patel P, Shaikh S, Hatekar A (2020) Support vector machine versus Naive Bayes classifier: a juxtaposition of two machine learning algorithms for sentiment analysis. Int Res J Eng Technol, July, pp 3553–3563. [Online]. www.irjet.net
22. Qasem M, Esmail A, Sajid (2022) Exploring the effect of N-grams with BOW and TF-IDF representations on detecting fake news. In: International conference on data analysis business industry 2022, pp 741–746. https://doi.org/10.1109/ICDABI56818.2022.10041537
23. Gupta K, Jiwani N, Afreen N (2023) A combined approach of sentimental analysis using machine learning techniques. Rev d'Intelligence Artif 37(1):1–6. https://doi.org/10.18280/ria.370101
24. Jain A, Jain V (2022) Sentiment classification using hybrid feature selection and ensemble classifier. J Intell Fuzzy Syst 42(2):659–668. https://doi.org/10.3233/JIFS-189738
25. Ojo OE, Gelbukh A, Calvo H, Adebanji OO (2021) Performance study of N-grams in the analysis of sentiments. J Niger Soc Phys Sci 3(4):477–483. https://doi.org/10.46481/jnsps.2021.201
26. Kumar S, Singh R, Khan MZ, Noorwali A (2021) Design of adaptive ensemble classifier for online sentiment analysis and opinion mining. PeerJ Comput Sci 7:1–24. https://doi.org/10.7717/peerj-cs.660
27. Wang Q, Sun L, Chen Z (2019) Sentiment analysis of reviews based on deep learning model. In: Proceedings of the 18th IEEE/ACIS international conference on computer technology and information science ICIS 2019, pp 258–261. https://doi.org/10.1109/ICIS46139.2019.8940267
28. Minaee S, Azimi E, Abdolrashidi A (2019) Deep-sentiment: sentiment analysis using ensemble of CNN and Bi-LSTM models. [Online]. http://arxiv.org/abs/1904.04206
29. Zulqarnain M, Ishak SA, Ghazali R, Nawi NM, Aamir M, Hassim YMM (2020) An improved deep learning approach based on variant two-state gated recurrent unit and word embeddings for sentiment classification. Int J Adv Comput Sci Appl 11(1):594–603. https://doi.org/10.14569/ijacsa.2020.0110174
30. Alharbi NM, Alghamdi NS, Alkhammash EH, Al Amri JF (2021) Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews. Math Probl Eng. https://doi.org/10.1155/2021/5536560
31. Onan A (2021) Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. Concurr Comput Pract Exp 33(23):1–12. https://doi.org/10.1002/cpe.5909
32. Fu X, Yang J, Li J, Fang M, Wang H (2018) Lexicon-enhanced LSTM with attention for general sentiment analysis. IEEE Access 6(c):71884–71891. https://doi.org/10.1109/ACCESS.2018.2878425
33. Zulqarnain M, Ghazali R, Aamir M, Hassim YMM (2022) An efficient two-state GRU based on feature attention mechanism for sentiment analysis. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-13339-4
34. Liu Y, Xu Q (2020) Short text classification model based on multi-attention. In: Proceedings of the 2020 13th international symposium on computational intelligence design issues. 2020, pp 225–229. https://doi.org/10.1109/ISCID51228.2020.00057
35. Kotzias D, Denil M, De Freitas N, Smyth P (2015) From group to individual labels using deep features. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, Aug 2015, vol 2015-August, pp 597–606. https://doi.org/10.1145/2783258.2783380
36. Adjuik TA, Ananey-Obiri D (2022) Word2vec neural model-based technique to generate protein vectors for combating COVID-19: a machine learning approach. Int J Inf Technol 14(7):3291–3299. https://doi.org/10.1007/s41870-022-00949-2
37. Ni R, Cao H (2020) Sentiment analysis based on GloVe and LSTM-GRU. In: Chinese control conference CCC, vol. 2020-July, pp 7492–7497. https://doi.org/10.23919/CCC50068.2020.9188578
38. Umer M et al (2022) Impact of convolutional neural network and FastText embedding on text classification. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-13459-x
39. Abid F, Li C, Alam M (2020) Multi-source social media data sentiment analysis using bidirectional recurrent convolutional neural networks. Comput Commun 157(April):102–115. https://doi.org/10.1016/j.comcom.2020.04.002
40. Miedema F (2018) Sentiment analysis with long short-term memory networks. Res Pap Bus Anal, pp 1–17. [Online]. https://cs.vu.nl/~sbhulai/papers/paper-miedema.pdf. Accessed 22 Oct 2023
41. Luong MT, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Conference Proceedings—EMNLP 2015 conference on empirical methods in natural language processing, pp 1412–1421. https://doi.org/10.18653/v1/d15-1166
42. Bahdanau D, Cho KH, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd international conference on learning representations ICLR 2015 conference track Proceedings, pp 1–15
43. Kotiyal B, Pathak H, Singh N (2023) Debunking multi-lingual social media posts using deep learning. Int J Inf Technol 15(5):2569–2581. https://doi.org/10.1007/s41870-023-01288-6