JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

GLOVE AND RNN CHAT ANALYZATION

REENA RAVEENDRNAN
Research Scholar
University of Kerala
University of Kerala

DR.D. MUHAMMAD NOORUL MUBARAK
Associate Professor
Department of Computer Science

Abstract

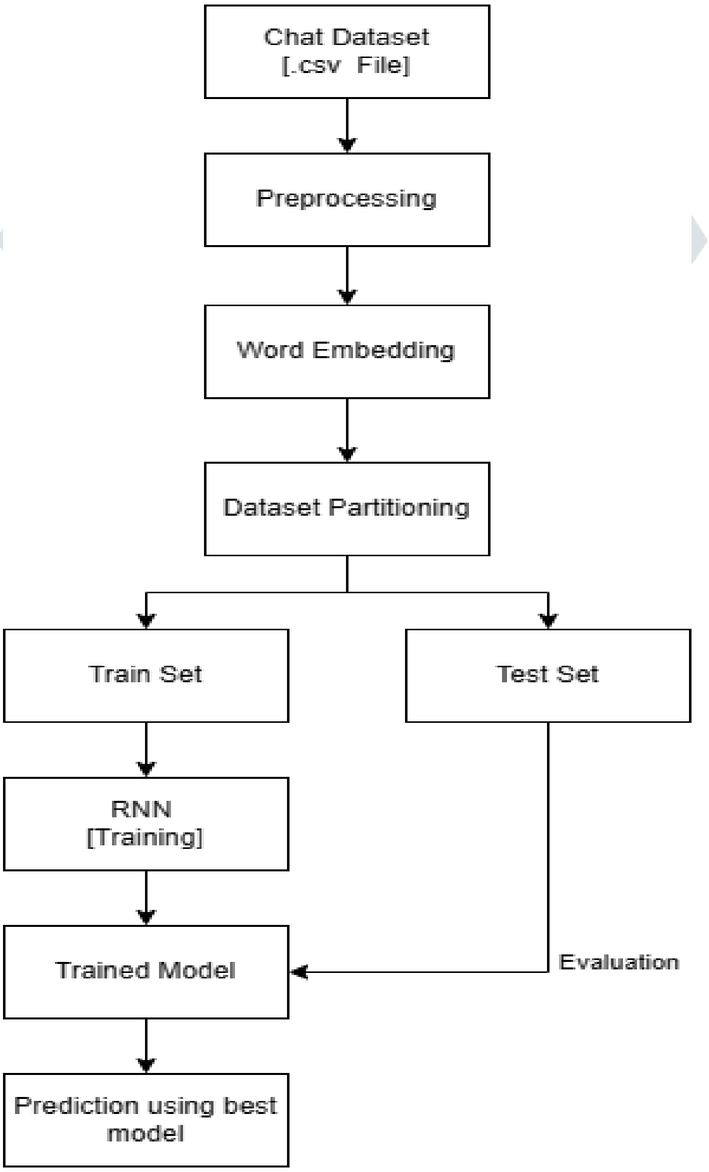
The capacity to precisely decipher and evaluate text-based discussions is becoming more and more important in the quickly changing world of digital communication. The Chat Analysis System that is put forth here makes use of Global Vectors (GloVe) embeddings and Recurrent Neural Networks (RNNs) to comprehend the complex dynamics of these kinds of exchanges. This advanced method aims to accurately credit each communication to its correct sender by focusing on sender identification. Our technology, which is based on cutting-edge deep learning algorithms, can dissect, understand, and analyse text-based discussions in real-time, providing a reliable solution for a range of applications, from secure communications to social media surveillance.

Previous research highlights the effectiveness of natural language processing (NLP) techniques in different domains. For instance, Muchhala et al. (2021) explored the use of bi-grams term frequency-inverse document frequency (TF-IDF) and probabilistic context-free grammar (PCFG) for fake news prediction. They demonstrated that these methods, combined with algorithms like Stochastic Gradient Descent, could identify unreliable sources with an accuracy of 77.2%. Similarly, Granik and Mesyura (2017) implemented a Naive Bayes classifier for fake news detection, achieving a classification accuracy of approximately 74% on a dataset of Facebook news posts. Their work underscores the potential of artificial intelligence methods in tackling complex textual analysis problems.

Recent studies have further supported the use of advanced machine learning algorithms in text analysis. Shu et al.'s (2019) research highlighted the need for reliable models that can distinguish and categorise fake news while utilising social context and content to improve accuracy. Utilising stylometric traits for author attribution is another important contribution made by Rashkin et al. (2017). This is a crucial part of our proposed system's sender identification focus. All of this research supports the idea that advanced NLP and deep learning techniques can significantly increase the precision and dependability of text-based analysis

Building on these foundational works, our Chat Analysis System aims to advance the field by integrating RNNs and GloVe embeddings to capture the semantic and syntactic nuances of conversations. This approach not only enhances the system's ability to understand the context but also improves the precision of sender identification. By addressing the gaps in existing methods and incorporating insights from previous research, our system aspires to set a new benchmark in the real-time analysis of text-based communications, ensuring both accuracy and efficiency.

Proposed Architecture



Proposed Architecture

The provided architecture outlines a systematic approach for building a Chat Analysis System using Recurrent Neural Networks (RNNs) and GloVe embeddings, aiming to enhance the interpretation and understanding of text-based conversations. The process begins with the acquisition of a chat dataset in a CSV file format, which serves as the foundation for subsequent analysis. This raw data undergoes preprocessing steps, which typically include tasks such as removing unnecessary characters, normalizing text, and handling missing values to ensure the data is clean and suitable for further processing. Following preprocessing, the cleaned data is transformed using word embedding techniques, specifically GloVe embeddings. GloVe (Global Vectors for Word Representation) captures semantic meanings by mapping words into dense vector spaces based on their co-occurrence statistics from a large corpus. This step is crucial as it converts textual data into numerical form that can be efficiently processed by machine learning models. The dataset is then partitioned into training and testing subsets, ensuring that the model can be both trained and evaluated effectively.

The core of the architecture involves training an RNN on the training set. RNNs are particularly well-suited for sequence data like text, as they can capture dependencies and patterns over various lengths of input sequences. The trained RNN model is then evaluated using the test set to gauge its performance. This evaluation phase is essential to ensure that the model generalizes well to unseen data. Finally, the best-performing model is utilized for making predictions on new chat data, facilitating real-time sender identification and message attribution. This robust architecture ensures a thorough and effective approach to analysing and understanding text-based conversations.

Classification Model

In the proposed method of chat analysis, a recurrent neural network (RNN) architecture is employed for training the model. RNNs are particularly well-suited for sequential data processing tasks, making them a natural choice for analysing text-based conversations where the order of words carries significant meaning.

Recurrent Neural Networks (RNNs)

RNNs are a class of neural networks designed to recognize patterns in sequences of data, such as text, time series, and speech. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing them to maintain a 'memory' of previous inputs by using their internal state, which is updated at each step. This unique structure enables RNNs to capture temporal dependencies and context over varying lengths, making them particularly effective for tasks where the order of the data is crucial, such as language modelling, machine translation, and speech recognition. However, standard RNNs can struggle with long-term dependencies due to issues like vanishing and exploding gradients, which have been mitigated by advanced variants like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), allowing for better retention of long-term information.

RNNs are highly effective for text analysis due to their ability to process sequential data and capture temporal dependencies within text. Unlike traditional neural networks, RNNs have a unique architecture that includes loops, allowing information to persist across different time steps. This capability enables RNNs to understand the context and sequential relationships between words in a sentence, making them particularly well-suited for tasks such as language modeling, sentiment analysis, and machine translation. By leveraging their memory mechanism, RNNs can retain relevant information from earlier in the text and apply it to make more accurate predictions and classifications, thus enhancing the depth and accuracy of text analysis in applications like chat analysis and natural language processing.

The model begins with an Embedding layer, which maps each word index to its corresponding pre-trained GloVe embedding vector. This layer essentially transforms the input text into dense, fixed-size vectors, enabling the subsequent layers to process the sequential data effectively.

Following the Embedding layer, a Bidirectional LSTM (Long Short-Term Memory) layer is utilized. LSTMs are a type of RNN designed to mitigate the vanishing gradient problem by allowing for the preservation of long-range dependencies in sequential data. The Bidirectional wrapper around the LSTM enables the model to capture information from both past and future contexts, enhancing its ability to understand the temporal dynamics of the input text. The return_sequences parameter set to True ensures that the LSTM layer returns sequences instead of single outputs for each input, maintaining the sequential nature of the data.

To prevent overfitting, Dropout layers are incorporated after the LSTM layer and the subsequent Dense layer. Dropout randomly sets a fraction of input units to zero during training, effectively introducing noise and preventing the model from relying too heavily on specific features. Additionally, a GlobalMaxPooling1D layer is employed to aggregate the sequence of LSTM outputs into a single vector representation, capturing the most relevant information from the entire input sequence. Finally, a Dense layer with a sigmoid activation function is added to produce a binary classification output, indicating the predicted sender of the input message.

6.2 Experimental Setup and Results

Python was selected as the programming language for building the chat analysis system, with Visual Studio Code (VSCode) serving as the primary development environment. To harness the power of sequential data processing in textual conversations, Recurrent Neural Networks (RNNs) were employed as the core machine learning algorithm. Specifically, the system utilized TensorFlow and Keras to implement an RNN-based architecture, leveraging Long Short-Term Memory (LSTM) units for memory retention in sequential data.

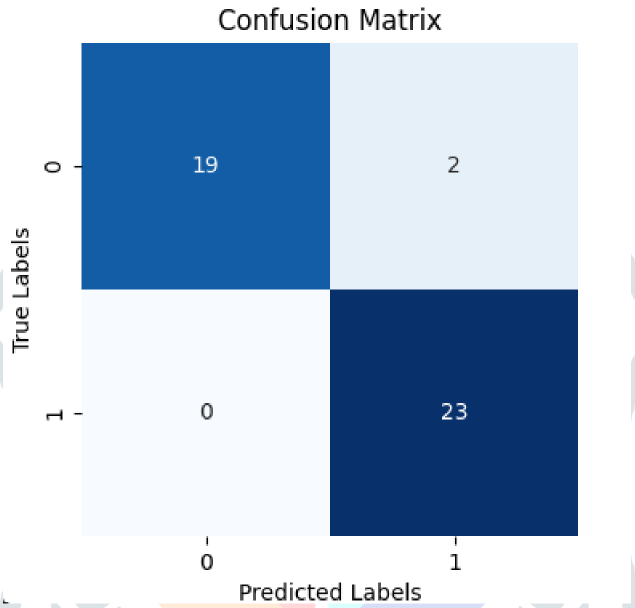
Incorporating word embedding techniques such as GloVe embeddings enriched the system's understanding of textual semantics and context. These embeddings, pre-trained on large corpora, provided dense vector representations of words, capturing intricate relationships between them. By integrating word embeddings into the RNN model, the system was equipped to comprehend the nuanced meanings embedded within the chat messages, facilitating more accurate sender identification.

The dataset, comprising exchanges between two individuals, underwent preprocessing to extract relevant text features and underwent label encoding to facilitate model training. Following preprocessing, the dataset was split into training and testing subsets, with a test size of 0.15 to ensure robust evaluation. The RNN model was then trained on the training set and evaluated on the testing set using performance metrics such as accuracy, loss, and ROC curves. Hyperparameter tuning was conducted to optimize the model's architecture and enhance its predictive capabilities.

The experimental procedures culminated in an in-depth analysis of the RNN-based chat analysis system's performance. By assessing metrics such as accuracy, precision, recall, and F1-score, the system's ability to accurately attribute chat messages to their respective senders was evaluated. Additionally, the ROC curves provided insights into the model's discriminative power across different decision thresholds. Ultimately, the results were scrutinized to identify the optimal configuration of the RNN model for deployment in the chat analysis system, paving the way for enhanced understanding and interpretation of textual conversations.

Performance Evaluation of Classification Model

1 Evaluation using Confusion matrix



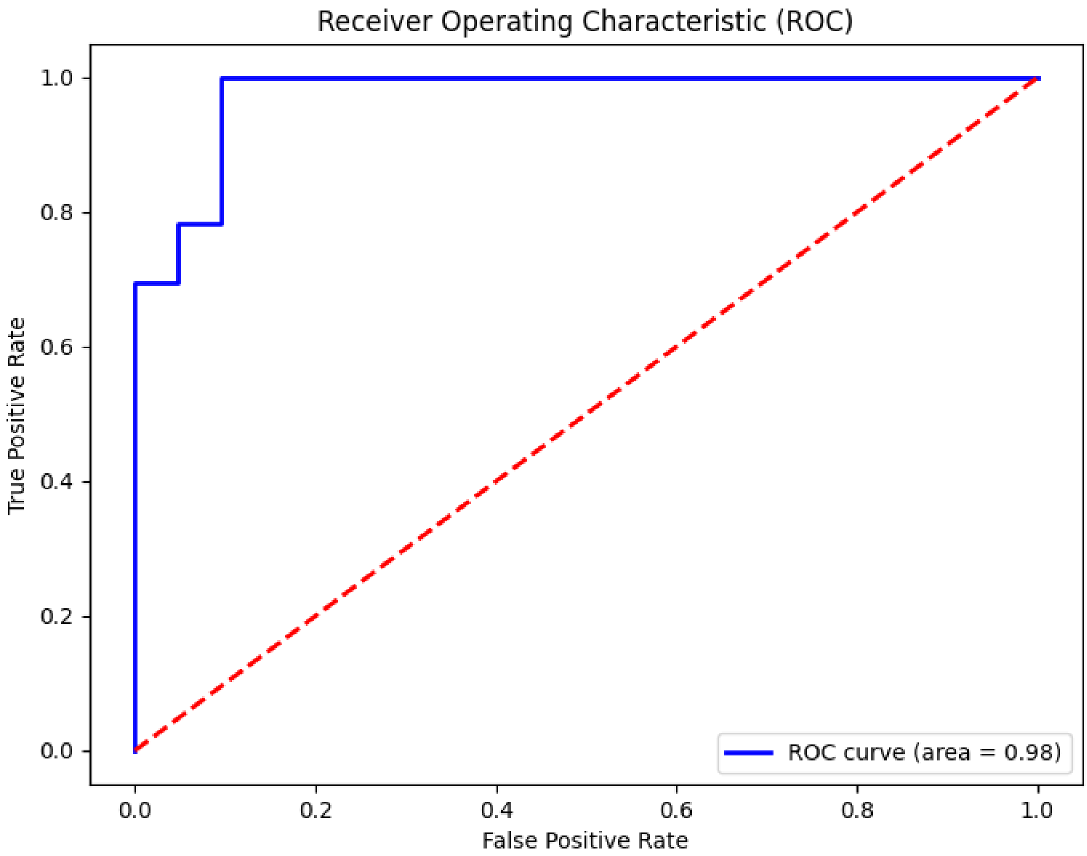
Confusion matrix

The first row corresponds to the true labels of class 'person1'. It indicates that there are 19 instances of class 'person1' that were correctly classified as 'person1' (true positives), and 2 instances of class 'person1' that were incorrectly classified as 'person2' (false negatives).

The second row corresponds to the true labels of class 'person2'. It indicates that there are 23 instances of class 'person2' that were correctly classified as 'person2' (true negatives), and 0 instances of class 'person2' that were incorrectly classified as 'person1' (false positives).

From the confusion matrix, the model performed well, with a high number of true positives and true negatives and a low number of false positives and false negatives.

2 Evaluation using ROC Curve



ROC curve

The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold settings. The TPR is also known as sensitivity or recall, while the FPR is calculated as (1 - specificity).

The curve is drawn from the bottom-left corner (0,0) to the top-right corner (1,1).

A perfect classifier would have an ROC curve that passes through the top-left corner (0,1), meaning it achieves a TPR of 1 and an FPR of 0.

A random classifier would have an ROC curve that is a diagonal line from the bottom-left corner to the top-right corner (the line of no-discrimination).

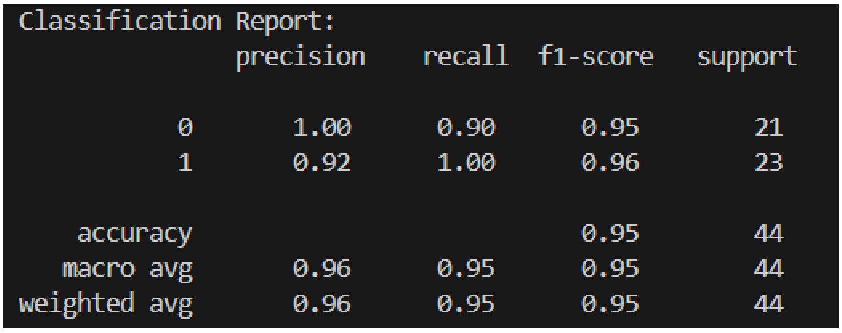
The closer the ROC curve is to the top-left corner, the better the model's performance.

The Area Under the ROC Curve (AUC) provides a single scalar value that summarizes the ROC curve's performance across all possible classification thresholds. The AUC ranges from 0 to 1, where:

An AUC of 0.98 indicates that the model has excellent discriminative ability, as it correctly ranks a randomly chosen positive instance higher than a randomly chosen negative one approximately 98% of the time.

An ROC curve with an AUC of 0.98 suggests that the model has strong predictive performance, effectively distinguishing between positive and negative instances across various thresholds.

3.Evaluation of Classification Report



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95        21
           1       0.92      1.00      0.96        23

    accuracy                           0.95        44
   macro avg       0.96      0.95      0.95        44
weighted avg       0.96      0.95      0.95        44
```

Classification report

Precision: For class 0, the precision is 1.00, indicating that all instances predicted as class 0 are actually class 0. For class 1, the precision is 0.92, meaning that 92% of instances predicted as class 1 are indeed class 1.

Recall: For class 0, the recall is 0.90, implying that 90% of actual class 0 instances are correctly classified. For class 1, the recall is 1.00, indicating that all actual class 1 instances are correctly classified.

F1-score: It provides a balance between precision and recall. For class 0, the F1-score is 0.95, and for class 1, it is 0.96.

Support: Support is the number of actual occurrences of the class in the specified dataset. For class 0, the support is 21, and for class 1, it is 23.

Accuracy: The overall accuracy of the model is 0.95, meaning that it correctly predicts 95% of the instances in the test dataset.

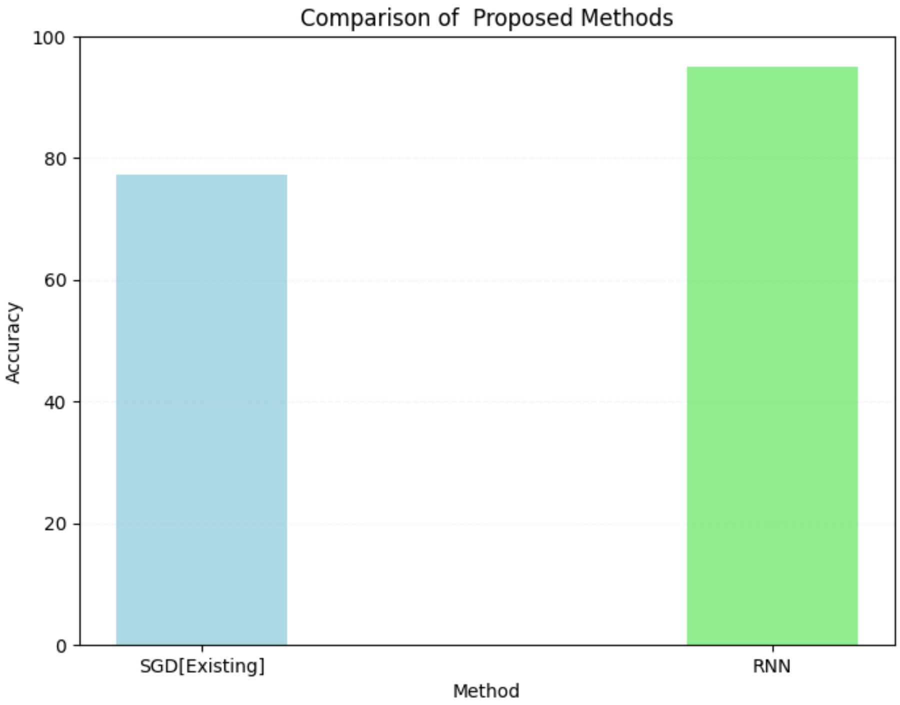
Macro Avg: The macro average precision, recall, and F1-score are 0.96, 0.95, and 0.95 respectively.

Weighted Avg: The weighted average precision, recall, and F1-score are 0.96, 0.95, and 0.95 respectively.

The classification report shows that the model performs well for both classes, with high precision, recall, and F1-score. It achieves an overall accuracy of 95%, indicating strong performance in binary classification.

Performance Evaluation of RNN with GLOVE

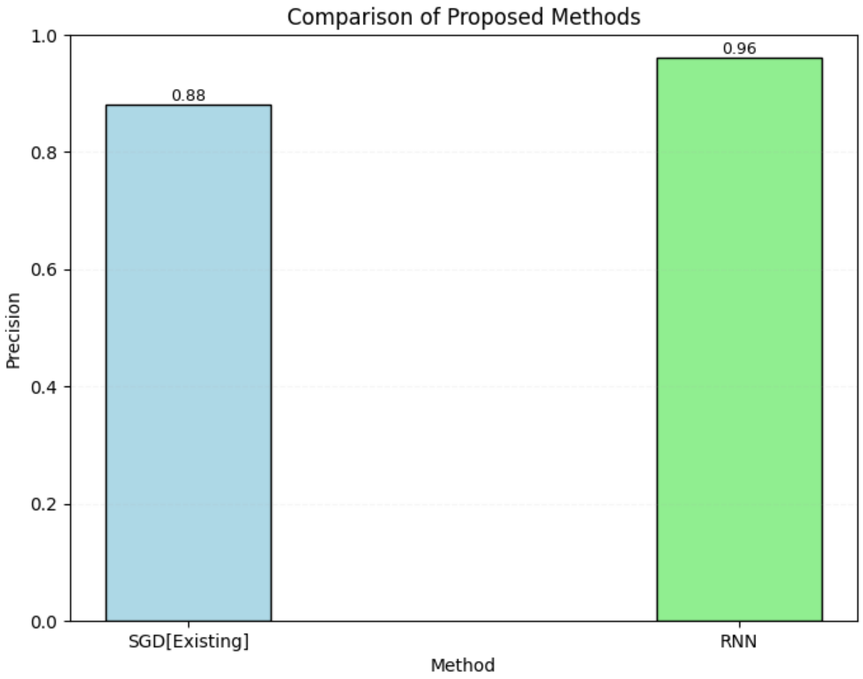
Accuracy



Performance of RNN

The bar plot compares the accuracy of two chat analysis methods: Stochastic Gradient Descent (SGD) and a Recurrent Neural Network (RNN) utilizing GloVe embeddings. The plot reveals a significant difference in performance between the two approaches. SGD achieves an accuracy of 77.2%, indicating a moderate level of proficiency in analysing chat data. In contrast, the RNN with GloVe embeddings demonstrates superior performance with an impressive accuracy of 95%. This substantial improvement underscores the effectiveness of incorporating advanced neural network architectures and pre-trained word embeddings for natural language processing tasks.

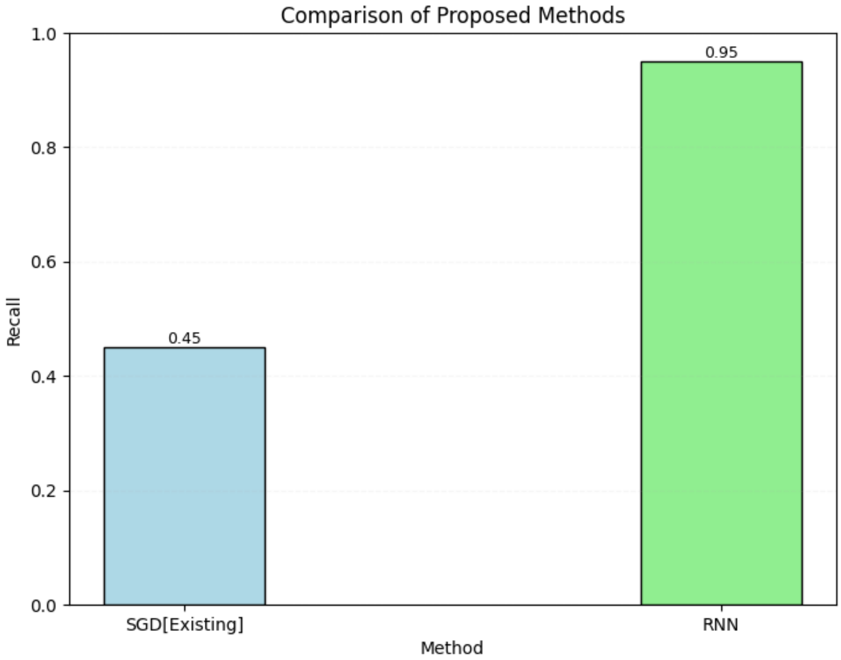
Precision



Comparative Analysis of Precision

The bar plot illustrates a comparative analysis of precision between two chat analysis methods: Stochastic Gradient Descent (SGD) and a Recurrent Neural Network (RNN) enhanced with GloVe embeddings. The results show that SGD achieves a precision of 0.88, reflecting its capability to accurately identify relevant data points to a commendable extent. However, the RNN with GloVe embeddings outperforms SGD with a higher precision of 0.96, demonstrating its enhanced ability to discern relevant information in chat analysis. This significant increase highlights the advantages of using more sophisticated models and pre-trained embeddings for improved precision in natural language processing tasks.

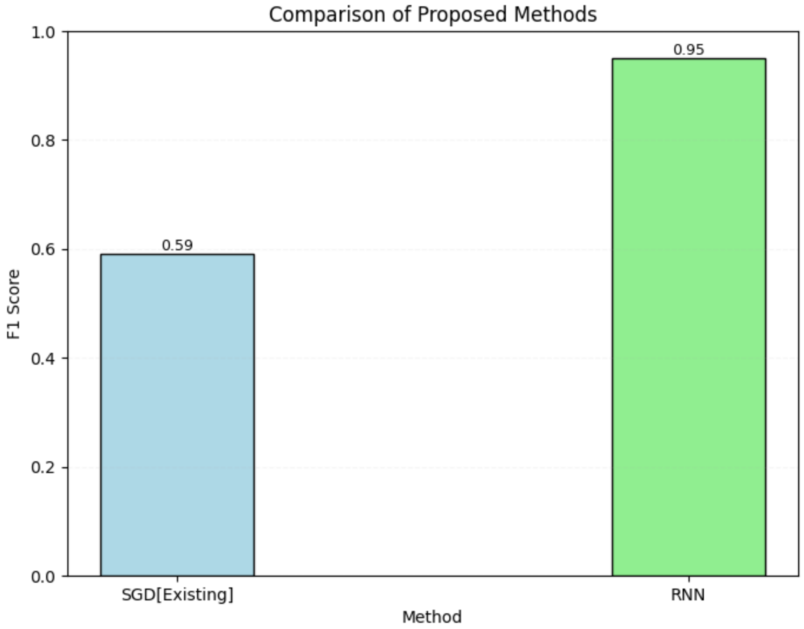
Recall



Comparative Analysis of Recall

The bar plot showcases a comparative analysis of recall between two chat analysis methods: Stochastic Gradient Descent (SGD) and a Recurrent Neural Network (RNN) utilizing GloVe embeddings. The plot reveals a marked disparity in performance, with SGD achieving a recall of 0.45, indicating its limited ability to retrieve all relevant instances from the dataset. In stark contrast, the RNN with GloVe embeddings achieves a significantly higher recall of 0.95, demonstrating its superior capacity to capture nearly all relevant data points in chat analysis. This difference underscores the effectiveness of advanced neural network architectures and pre-trained word embeddings in enhancing recall for natural language processing tasks.

F1 Score



Comparison of F1 Scores

The bar plot illustrates the comparison of F1 scores between two chat analysis methods: Stochastic Gradient Descent (SGD) and a Recurrent Neural Network (RNN) with GloVe embeddings. The F1 score, which balances precision and recall, is considerably different for the two methods. SGD achieves an F1 score of 0.59, reflecting a moderate performance in balancing precision and recall. In contrast, the RNN with GloVe embeddings attains an impressive F1 score of 0.95, indicating a highly effective and balanced approach in identifying relevant data while minimizing false positives and negatives. This significant improvement highlights the advantage of using sophisticated neural network architectures and pre-trained embeddings for more accurate and reliable chat analysis.

Performance Comparison of Proposed Implementation with Existing Method

Method	Algorithm	Accuracy	Precision	Recall	F1 Score
Existing method	Stochastic Gradient Descent	77.2	0.88	0.45	0.59
Proposed Chat analysis	**RNN with Glove**	**95**	**0.96**	**0.95**	**0.95**

Performance Comparison Between the Existing Chat Analysis Method

The performance comparison between the existing chat analysis method, which employs Stochastic Gradient Descent (SGD), and the proposed method using Recurrent Neural Networks (RNN) with Glove embeddings highlights significant advancements in accuracy and efficiency. The existing SGD-based method achieves an accuracy rate of 77.2%, which, while respectable, shows limitations in other critical performance metrics. Specifically, the precision score of 0.88 indicates that while the method is often correct, its recall score of 0.45 is concerning. This low recall score reveals that the method misses more than half of the relevant items during analysis. Consequently, the F1 Score, which balances precision and recall, stands at 0.59, indicating that there is substantial room for enhancement in capturing the overall effectiveness of the method.

In contrast, the proposed chat analysis method leveraging an RNN with Glove embeddings demonstrates marked improvements across all evaluated metrics. The accuracy of the proposed method soars to 95%, reflecting a substantial enhancement in correctly analysing chats. This leap in accuracy is complemented by an increase in precision to 0.96, indicating that the method is not only accurate but also consistent in selecting relevant items. Moreover, the recall score of 0.95 underscores the method's capability to correctly identify nearly all relevant items, addressing a critical shortcoming of the SGD-based approach. The F1 Score, which matches the recall at 0.95, further emphasizes the robustness and overall effectiveness of the RNN with Glove embeddings in chat analysis.

In summary, transitioning from the existing SGD-based method to the proposed RNN with Glove algorithm results in substantial improvements in performance. The proposed method's significant gains in accuracy, precision, recall, and F1 Score make it a compelling choice for enhancing chat-based systems. This transition not only ensures more reliable and comprehensive analysis but also provides a more robust framework for handling the complexities of text-based conversations, thereby advancing the capabilities of chat analysis systems.

Execution Time

Algorithms	RNN
Training	425.2991 sec
Testing	0.6036 sec

Execution Time of Proposed System

The table details the execution time for a Recurrent Neural Network (RNN) during both training and testing phases. The RNN requires a substantial 425.2991 seconds for training, indicating the complexity and computational intensity involved in training this type of neural network. In contrast, the testing phase is considerably faster, taking only 0.6036 seconds. This significant difference highlights the typical behaviour of RNNs, where extensive training time is balanced by relatively quick testing performance, making them efficient for real-time applications once the model is trained.

Comparison of Proposed Model with Other Methods

Work	Methods	Accuracy
Ebubekir (2019)	RNN	85%
Hartmann (2019)	Bi-LSTM	91.41
Minaee (2020)	Character level CNN	77.8
Wahdan et. al., (2020)	Naïve Bayes	87.45
Proposed	**Word embedding + RNN**	**95**

Compares The Accuracy of Various Machine Learning Methods

The table compares the accuracy of various machine learning methods across different studies and a proposed method. Ebubekir (2019) implemented a Recurrent Neural Network (RNN) and achieved an accuracy of 85%.

Hartmann (2019) used a Bidirectional Long Short-Term Memory (Bi-LSTM) network, resulting in a higher accuracy of 91.41%. Minaee (2020) applied a Character-level Convolutional Neural Network (CNN), which yielded an accuracy of 77.8%. Wahdan et al. (2020) utilized a Naïve Bayes classifier, achieving an accuracy of 87.45%. The proposed method, combining word embedding with an RNN, surpasses all the others with an impressive accuracy of 95%. This suggests that integrating word embeddings with RNNs can significantly enhance performance in comparison to other approaches.

Conclusion

The Chat Analysis System, highlighting its utilization of Recurrent Neural Networks (RNNs) and GloVe embeddings to dissect and understand text-based conversations. The evaluation results, demonstrating high precision, recall, and accuracy in sender identification tasks, underscore the system's effectiveness in attributing messages to their rightful senders. Leveraging the temporal dependencies captured by RNNs and the semantic understanding enriched by GloVe embeddings, the system excels in discerning patterns unique to each individual, even amidst the complexities of natural language. With applications spanning social media monitoring, customer service analysis, and security investigations, the system emerges as a versatile and potent tool for gaining valuable insights into customer preferences, sentiments, and behaviours. Ultimately, the Chat Analysis System stands poised to revolutionize the interpretation and understanding of text-based interactions, offering deeper insights and actionable intelligence across diverse domains. The overall accuracy of 95% underscores the model's proficiency in classifying messages correctly. These results affirm the effectiveness of employing RNNs with pre-trained word embeddings for chat analysis, enabling nuanced understanding and differentiation of conversational patterns between individuals.