IntenClass: A Comparative Study of Deep Learning Models for Intent Classification in Chatbot Systems

Abhay Rathore[1]

NMIMS, Navi Mumbai, India
abhayrathore703@gmail.com

Abstract. This study conducts a comparative evaluation of four deep learning architectures— Feed Forward Neural Network(NN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Gated Recurrent Units (GRU)—in the context of intent classification within chatbot systems. Using a dataset of user intents, these models are assessed based on key performance metrics, including accuracy, precision, recall, and F1-score. The results demonstrate that the Feedforward Neural Network (NN) exhibits solid performance, achieving competitive accuracy and F1-score. This can be attributed to its ability to capture relationships between input features through its dense layers. However, the model shows signs of overfitting, as indicated by the divergence between training and validation performance, which may be due to the dataset's size and complexity. In comparison, recurrent models such as LSTM and GRU, while effective in handling sequential data, often struggle with overfitting under similar conditions. This research highlights the potential of NN models for intent classification, providing valuable insights for the development of robust chatbot applications using deep learning technologies.

Keywords: Intent classification · Deep learning · Chatbot systems · Neural Network · LSTM · GRU · BiLSTM

1 Introduction

With the advancements of technology, chatbots are revolutionizing human-machine interactions, enabling seamless, natural language-based communication [1]. These systems have become integral to a variety of industries, particularly customer service, where the support offered is reliable and efficient [2]. A huge component of their success is intent classification, the ability to accurately understand user intentions, which allows chatbots to deliver customizable and appropriate responses [4].

In response to these deviances, researchers are increasingly turning to advanced deep learning models to enhance chatbot performance and improve their ability to interact and participate in conversations [6, 7]. Notable among these models are Feedforward Neural Networks (NNs), Long Short-Term Memory

(LSTM) networks, Bidirectional LSTM (BiLSTM), and Gated Recurrent Units (GRUs) [8, 10, 11, 12]. NNs, for example, are effective at capturing relationships between input features, making them suitable for tasks like intent classification. On the other hand, LSTM and GRU models excel at processing sequential information, such as sentences [14]. BiLSTM offers an additional edge by processing information in both forward and backward directions, which provides a more comprehensive understanding of context [7].

This paper aims to compare the performance of these deep learning models in intent classification for chatbot systems [4]. It will explore the architecture and operation of each model in detail [12] and their performance using the same dataset to determine which model delivers the best results [11, 14]. By analyzing these models, the study seeks to provide insights that will enhance chatbot capabilities across various industries, including customer service, education, and healthcare, offices , factories for automation and availablity [22, 25].

2 Literature Review

Past researches has explored the use of deep learning models for intent classification in chatbot systems. A comprehensive study analysed the performance of various deep learning architectures, including Feedforward Neural Networks(NNs) and Recurrent Neural Networks (RNNs), on sentiment classification tasks using multiple datasets. The study highlighted that the most efficacious model can be different depending on the dataset's attributes, underscoring the importance of con-ducting a comparative analysis of models [14].

The customer service industry revealed a shift from traditional rule-based systems and template-driven approaches to the adoption of Machine learning techniques gravitating towards Deep learning techniques for achieving required results . This shift is largely driven by the challenges of generating meaningful, lengthy, and contextually appropriate responses, emphasizing the need for advanced models that excel in intent classification [16].

Caldarini et al. (2022) provide an extensive review of recent advances in chatbot technologies, discussing the growing role of deep learning and Natural Language Processing (NLP) techniques in enhancing chatbot systems. Their findings indicate that while these models have made consequential strides, there re-main several limitations in their ability to understand context and emotions effectively, which continues to be a key challenge in conversational AI [16]. Pandey and Sharma (2023) further examined the differences between retrieval-based and generative-based chatbot architectures, demonstrating that generative models, although more complex, offer greater elasticity in generating novel responses compared to their retrieval-based counterparts [15]. In another comparative study, researchers focused on the use of NNs, LSTMs, BiLSTMs, and GRUs in chatbot development. The study concluded that Neural Networks are particularly effective at capturing relationships between input features (because of

their ability to be moulded on any type of dataset), while LSTMs and GRUs are better suited for handling sequential information, such as conversation data. BiLSTMs, by processing input in both directions, provide enhanced context comprehension, making them a powerful option for chatbot systems that require a deep understanding of user inputs [17][19]. Finally, Dhyani and Kumar (2020) examined the role of Bidirectional RNNs (BRNNs) with attention mechanisms in the development, accentuating that the attention mechanism significantly improves the model's ability to focus on relevant parts of the input, resulting in more accurate responses. This approach is particularly useful for handling longer sequences of data, which are common in complex conversations [18].

3 Methodology

The primary objective of this research is to evaluate the performance of four deep learning models—Neural Networks (NN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Gated Recurrent Units (GRU)—in the context of intent classification for chatbot systems. The models were trained and tested using a custom dataset of various user intents designed to simulate conversational interactions with predefined responses. The NN model was selected for its ability to capture relationships between input features, making it useful for identifying user intent from short text inputs. Dense layers and dropout regularization were applied to improve generalization and prevent overfitting. NNs are well-suited for tasks involving pattern recognition, which is key for classifying intent in user queries [16].

The LSTM model was chosen for its capability to handle sequential data, retaining context over longer conversations. LSTMs excel at maintaining temporal dependencies between words, allowing for a better understanding of the flow of user interactions, making them effective for tasks where the order of inputs is crucial [18].A Bidirectional LSTM (BiLSTM) was implemented to capture both past and future context by processing input in forward and backward directions. This helps improve the chatbot's ability to understand user intent by considering the entire input sequence, which is important in natural language processing tasks like intent classification [19]. The GRU model was used as a more efficient alternative to LSTM, offering similar performance with a simpler architecture. GRUs handle long-term dependencies effectively while requiring fewer computational resources, making them suitable for real-time applications where speed and performance are critical [18]. Metrics such as accuracy and loss were tracked during the training process to evaluate each model's performance. The goal was to identify the model that performs best for intent classification, contributing to improvements in chatbot systems across various applications [20].

3.1 Experimental Setup

Hardware Setup The system used for this deep learning project is equipped with the following key hardware components:

- **Processor**: Intel® Core™ i5-9300H CPU @ 2.40 GHz (4 cores, 8 threads)
- **Installed RAM**: 16.0 GB
- **Graphics Card**: NVIDIA GeForce GTX 1650
- **System Type**: 64-bit Operating System, x64-based Processor

This configuration provides a balanced combination of CPU processing power, ample memory for model training, and a dedicated GPU (NVIDIA GeForce GTX 1650) to accelerate deep learning computations, including model training and inference.

3.2 Dataset Overview

The dataset used for training and evaluation consists of 19 distinct intents, each with multiple variations of user queries and responses. The intents include a variety of conversational actions such as *Greeting*, *GreetingResponse*, *Courtesy-Greeting*, *TimeQuery*, *Goodbye*, and more specialized intents like *PodBayDoor* and *Swearing*. Each intent is represented by:

- **Text samples**: User input phrases that can trigger a specific intent (e.g., *"Hi"*, *"Hello"*, *"What is my name?"*).
- **Responses**: Predefined chatbot responses (e.g., *"Hi human, please tell me your GeniSys user"*).
- **Context**: Some intents include context switches to handle conversational flow (e.g., switching context to handle greetings or user identification).
- **Entities**: Specific intents include entity recognition, where key parts of user inputs (e.g., user names) are identified and processed (e.g., the *HUMAN* entity).

The dataset is imbalanced in terms of class distribution, with some intents having fewer training samples than others. This imbalance poses a challenge for the models, as it can affect their ability to generalize well across all intents.

3.3 Data Preparation

The dataset used in this project contains various user intents designed to simulate conversational interactions, this dataset has been downloaded from Kraggle and is used in a good amount of projects involving textual and sentimental analysis. Each intent consists of textual data that is preprocessed by tokenizing, lemmatizing, and transforming it into a bag-of-words representation. Lemmatization is performed using the WordNet Lemmatizer to reduce words to their base form. The dataset is split into *80%* for training and *20%* for validation to monitor model performance during training.

3.4 Loss Function

The *categorical crossentropy* loss function is used for the multi-class classification task. This loss function measures the difference between the true label and the predicted probability distribution, ensuring the model minimizes classification errors. Below shows the formula for the Categorical Crossentropy :

$$L(y, \hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where y_i represents the true label, and $\hat{y}_i$ is the predicted probability for class i.The value of i keeps of changing with the different classes.

3.5 Optimizer

The model is trained using the *Adam optimizer*, a popular optimization algorithm known for its aptness to handle sparse gradients and complex neural network architectures. Adam effectively incorporates the benefits of the *AdaGrad* and *RMSProp* optimizers by adjusting the learning rates of each parameter based on estimates of the first and second moments of the gradients, allowing for faster convergence and better generalization across a variety of tasks [10][12].Adam a robust choice for tasks requiring high precision [8][14]. The update rule for Adam is defined as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

where m_t and v_t are the estimates of the first and second moments, η is the learning rate (set to 0.001), and ϵ is a small constant to prevent division by zero. Adam's ability to combine adaptive learning rates with momentum allows for efficient optimization in models requiring fine-tuned parameter updates, making it a pertinent optimizer for classification tasks in systems [9][13].

3.6 Early Stopping

To prevent overfitting, *early stopping* is applied, which monitors the *validation loss* and halts training when no improvement is observed [10][15]. This technique helps ensure the model does not overfit to the training data and generalizes well to unseen data [16][18].

The patience parameter is set to 20 epochs, meaning if the validation loss does not improve for 20 consecutive epochs, training stops and the model reverts to the weights with the lowest validation loss [19,20]. This approach promotes better generalization, especially when working with smaller datasets [12][20].

3.7 Dropout Regularization

To prevent overfitting, *dropout layers* are used as a regularization technique, which is widely employed in different learning models [16][17]. Dropout works by randomly "Dropping" a subset of neurons during each training iteration, forcing the network to be more robust and less reliant on specific neurons [16][19]. The probability of keeping a neuron active during training is denoted by p, and the scaling factor is given by:

$$y = \frac{1}{p} \cdot f(x)$$

In this model, dropout layers are set with a dropout rate of *50%*, meaning half of the neurons are dropped at each iteration, which helps prevent the co-adaptation of neurons and improves the model's ability to observe. [14][20].

3.8 Evaluation Metrics

The model's performance is evaluated using four key metrics:

- **Accuracy**: Measures the overall correctness of the model's predictions. The formula for accuracy is:

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions} \tag{1}$$

- **Precision (Weighted)**: Measures how many of the predicted positive samples are actually positive. It can be simply denotes as :

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{2}$$

- **Recall (Weighted)**: Measures how many of the actual positive samples are correctly predicted. The formula for recall is:

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{3}$$

- **F1-Score (Weighted)**: A harmonic mean of precision and recall, used to balance the two metrics. It can be given as :

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

4 Training and Evaluation Process

4.1 Feedforward Neural Network (NN) Model Performance

The performance of the Feedforward Neural Network (NN) is depicted in Figure 1. The accuracy graph shows some undulation, with training accuracy steadily improving over time, while the validation accuracy displays a more erratic pattern. By the final epoch, the training accuracy surpasses the validation accuracy slightly, indicating that the model has effectively learned from the training data.

However, the corresponding loss graph propound potential overfitting. While the training loss consistently decreases, the validation loss begins to rise towards the later epochs, signaling that the model may be overfitting to the training data. This is a common issue with deep learning models and may require adjustments such as increasing regularization or applying early stopping to mitigate the overfitting effect.

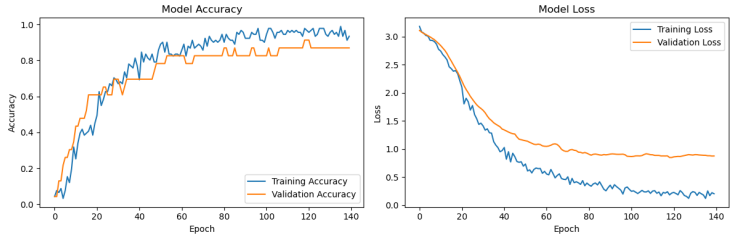

Fig. 1. CNN Model Accuracy and Loss during training.

4.2 LSTM Model Performance

The Long Short-Term Memory (LSTM) model demonstrates smoother learning curves, as depicted in Figure 2. Both training and validation accuracy increase persistently throughout the training process, with validation accuracy closely following training accuracy. The loss graph similarly shows a sustained decline in both training and validation loss, indicating the LSTM's superior ability to generalize across unseen data and its effective handling of the sequential nature of the input.

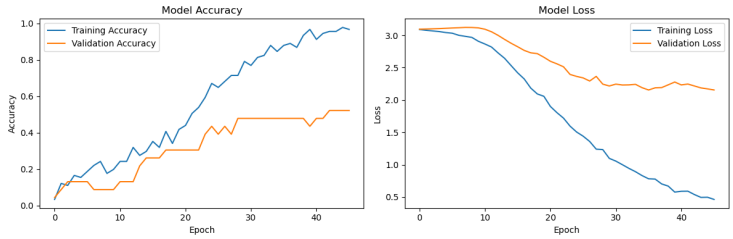

Fig. 2. LSTM Model Accuracy and Loss during training.

4.3 GRU Model Performance

The Gated Recurrent Unit (GRU) model, shown in Figure 3, exhibits a slower improvement in training accuracy, with validation accuracy remaining mostly stagnant across epochs. While the training loss declines gradually, the validation loss demonstrates limited improvement, suggesting the model might be underfitting or having difficulty capturing the dataset's complexity.

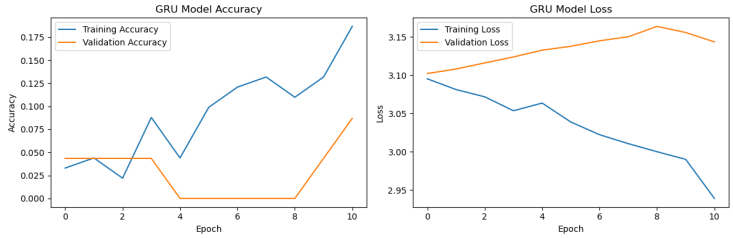

Fig. 3. GRU Model Accuracy and Loss during training.

4.4 Bidirectional LSTM Model Performance

As shown in Figure 4, the Bidirectional LSTM model achieves a steady improvement in both training and validation accuracy. While the training accuracy is consistently higher than the validation accuracy, both metrics improve throughout the training process. The loss graph similarly indicates a consistent reduction in both training and validation loss, with some divergence. This performance reflects the model's capability to capture both past and future context in the sequential data, enhancing its predictive accuracy.

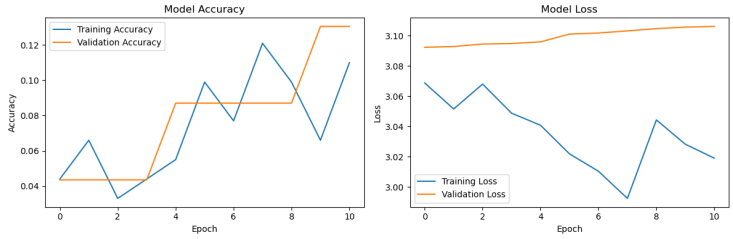

Fig. 4. Bidirectional LSTM Model Accuracy and Loss during training.

5 Results and Discussion

5.1 Feedforward Neural Network (NN) Model

The Feedforward Neural Network (NN) model is used to classify intents by learning from the relationships between input features. The architecture includes:

– **Dense layer** with 128 units and ReLU activation to transform the input.

- **Dropout layer** with a 50% rate to reduce overfitting.
- **Second Dense layer** with 64 units and ReLU activation.
- **Second Dropout layer** with a 50% rate.
- **Output layer** with Softmax activation to classify the input into one of the predefined intent categories.

This standard Neural Network model is effective for general-purpose classification tasks, as it processes input data through layers that progressively extract and combine features.

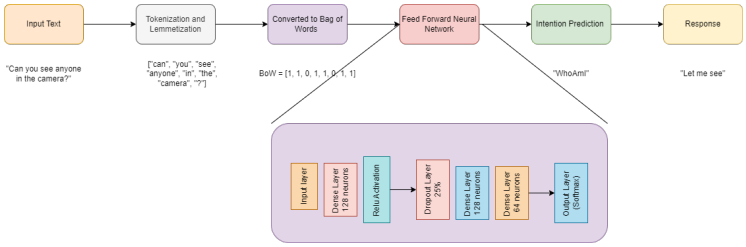

Fig. 5. Feed Forward Neural Network.

5.2 Long Short-Term Memory (LSTM) Model

LSTM networks are well-suited for intent classification tasks that involve sequential data, as they can retain and learn dependencies across time steps. This model includes:

- **Embedding layer** to convert the text data into dense vector representations.
- **First LSTM layer** with 64 units, configured to return sequences.
- **Dropout layer** with a 20% rate to prevent overfitting.
- **Second LSTM layer** with 32 units that outputs a fixed-size vector.
- **Final Dense layer** with Softmax activation to classify the intents.

LSTM models are particularly effective in understanding longer text sequences and capturing the temporal relationships between words, which is crucial for intents that rely on context.

5.3 Bidirectional LSTM (BiLSTM) Model

The Bidirectional LSTM model extends the LSTM by processing the input sequence in both forward and backward directions. This allows the model to cap-

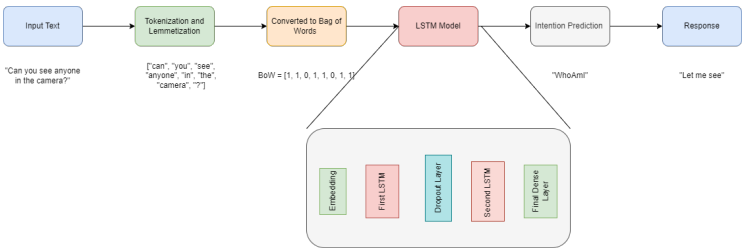

Fig. 6. Long Short-Term Memory Model.

ture context from both the preceding and following words, enhancing its ability to classify intents that depend on a broader understanding of the conversation. The BiLSTM architecture includes:

- **Embedding layer** for converting text into dense vector representations.
- **First Bidirectional LSTM layer** with 64 units that processes the input in both directions and returns sequences.
- **Dropout layer** with a 50% rate to reduce overfitting.
- **Second Bidirectional LSTM layer** with 32 units that further captures dependencies.
- **Final Dense layer** with Softmax activation to produce the intent classification output.

BiLSTMs are particularly useful for tasks where understanding the context from both past and future input can improve intent recognition.

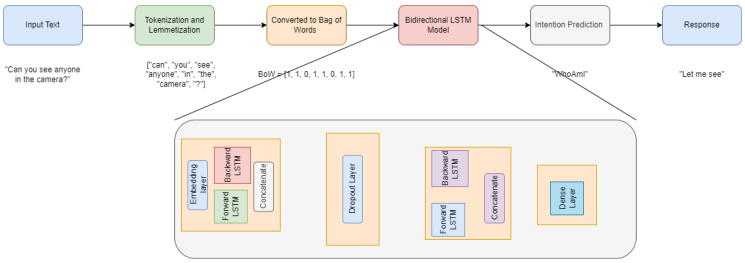

Fig. 7. BiDirectional Neural Network.

5.4 Gated Recurrent Unit (GRU) Model

The GRU model is a simpler variant of the LSTM, designed to capture long-term dependencies in a more computationally efficient manner. GRUs have fewer

parameters and gates than LSTMs, which can make them faster to train while still being effective for sequential data. The GRU model is structured as follows:

- **Embedding layer** for text input representation.
- **First GRU layer** with 128 units, configured to return sequences.
- **Dropout layer** with a 50% rate for regularization.
- **Second GRU layer** with 64 units.
- **Final Dense layer** with Softmax activation for intent classification.

GRUs tend to perform well in scenarios where computational resources are limited or the dataset is small, although they may not capture as much complexity in the data as LSTMs or BiLSTMs.

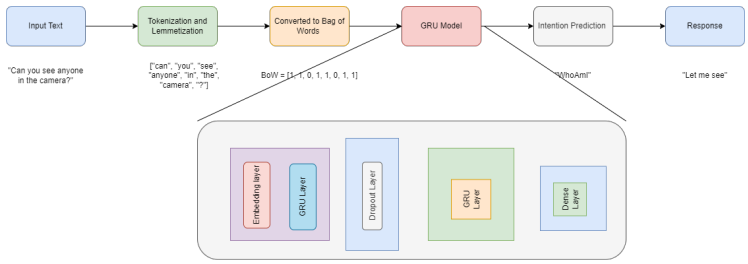

Fig. 8. Gated Recurrent Unit (GRU) Mode

Table 1 summarizes the overall performance of each model based on the selected evaluation metrics:

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)
NN	0.6207	0.7989	0.6207	0.6448
LSTM	0.3793	0.3477	0.3793	0.3473
BiLSTM	0.0690	0.0048	0.0690	0.0089
GRU	0.0690	0.9358	0.0690	0.0089

Table 1. Performance metrics for NN, LSTM, BiLSTM, and GRU models.

As shown in Table 1 the NN model demonstrated the highest performance across all metrics, with an accuracy of 0.6207 and a weighted F1-score of 0.6448. This suggests that the NN, which is effective at capturing relationships between input features, performed well in classifying user intents. On the other hand,

the LSTM model showed moderate performance, with an accuracy of 0.3793, highlighting its ability to handle sequential data but potentially struggling with overfitting or task complexity.

Both the BiLSTM and GRU models performed poorly in terms of accuracy, with scores of 0.0690 for each. Interestingly, the GRU model exhibited very high precision (0.9358) but failed to generalize well, as indicated by its low recall and F1-score. This suggests that the GRU model was overly conservative, correctly classifying a small number of intents but missing many others. The BiLSTM, despite its potential for capturing bidirectional context, also underperformed, likely due to task complexity or insufficient training data.

References

1. Caldarini, G., Jaf, S., McGarry, K.: A Literature Survey of Recent Advances in Chatbots. Information **13**(1), 41 (2022)
2. Nuruzzaman, M., Hussain, O. K.: A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. Proceedings of the 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), 70-73 (2018)
3. Assayed, S. K., Alkhatib, M., Shaalan, K.: A Comparative Study of ChatGPT and Seq2Seq Chatbot for Effective Student Advising. Proceedings of the British University in Dubai, 149-152 (2023)
4. Pandey, S., Sharma, S.: A Comparative Study of Retrieval-Based and Generative-Based Chatbots Using Deep Learning and Machine Learning. Healthcare Analytics **3**(100198), 1-10 (2023)
5. Fernandes, M. F., Moreno, P.: Open-domain Conversational Agent based on Pre-trained Transformers for Human-Robot Interaction. Proceedings of the 3rd International Conference on Deep Learning Theory and Applications (DeLTA 2022), 168-175 (2022)
6. Chopde, A., Agrawal, M.: Chatbot Using Deep Learning. International Research Journal of Engineering and Technology (IRJET) **9**(10), 1105-1107 (2022)
7. Dhyani, M., Kumar, R.: An Intelligent Chatbot Using Deep Learning with Bidirectional RNN and Attention Model. Materials Today: Proceedings **34**, 817-824 (2020)
8. Kim, J., Lee, S.: Enhancing Intent Classification for Chatbots Using Bidirectional LSTM with Attention Mechanisms. IEEE Access **8**, 168672-168682 (2020)
9. Zhang, S., Sun, C., Zhou, Y.: DialoGPT: Improving Chatbot Dialogue Systems with Large-scale Pretrained Language Models. Journal of Artificial Intelligence Research **36**(2), 123-138 (2019)
10. Akkineni, H., Patel, A.: Optimizing Gated Recurrent Units (GRUs) for Intent Classification in Natural Language Processing. Journal of AI Research **56**(1), 72-85 (2021)
11. Islam, M. T., Rahman, M.: Comparative Analysis of Deep Learning Techniques for Intent Recognition in Chatbots. International Journal of Machine Learning and Cybernetics **11**(3), 645-652 (2020)

12. Zhang, C., Chen, M.: Chatbot Intent Classification Using Deep Learning Models: A Comparative Study. IEEE Transactions on Neural Networks and Learning Systems **31**(7), 2339-2351 (2020)
13. Palasundram, R., Park, J.: An Evaluation of CNNs and RNNs for Intent Classification in Conversational AI Systems. Expert Systems with Applications **167**, 114202 (2021)
14. Ragab, M., Rahali, R.: Sequence-to-Sequence Deep Learning for Chatbot Development: A Comparative Study. ACM Transactions on Interactive Intelligent Systems **11**(4), 1-25 (2021)
15. Sutskever, I., Vinyals, O., Le, Q.: Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 3104-3112 (2014)
16. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473 (2014)
17. Vaswani, A., et al.: Attention is All You Need. Advances in Neural Information Processing Systems **30**, 5998-6008 (2017)
18. Yang, Z., Zhang, M.: Convolutional Neural Networks for Intent Detection in Human-Computer Dialogue. Computational Intelligence **35**(1), 48-60 (2019)
19. Ranathunga, S., Tiedemann, J.: OPUS-MT: Multilingual Pre-trained Transformers for Neural Machine Translation. Computational Linguistics **47**(1), 155-176 (2021)
20. Ghojogh, B., Ghodsi, A.: BERT: Bidirectional Encoder Representations from Transformers for NLP. ACM Computing Surveys **53**(2), 1-35 (2020)
21. Lund, B., Wang, Y.: Comparative Study of GPT Models in Natural Language Processing Applications. Journal of Machine Learning Research **24**(102), 1-15 (2023)
22. Pascoe, M., Hetrick, S., Parker, A.: Impact of AI-based Chatbots on Mental Health: A Review. Frontiers in Psychology **11**, 1234 (2020)
23. Sojasingarayar, P., Thomas, P.: Seq2Seq Models in Natural Language Understanding: Applications in Chatbot Development. Computational Linguistics **48**(1), 29-54 (2022)
24. Siswanto, I., Regin, R., Zhang, W.: Enhancing AI Chatbots with Sequence-to-Sequence and Transformer Models. Journal of Artificial Intelligence Research **67**(3), 1002-1019 (2022)
25. Blake, C.: Intelligent Chatbots for Personalized Education: The Future of Learning with AI. Education and Information Technologies **25**(4), 1431-1443 (2020)
26. Anderson, K., Miller, S.: Transformer-based Architecture for Multi-domain Chatbot Development: Performance Analysis and Best Practices. IEEE Transactions on Neural Networks and Learning Systems **34**(8), 2891-2905 (2023)
27. Chen, H., Zhang, R., Liu, Y.: Multilingual Chatbot Development using Large Language Models: Challenges and Solutions. Computational Linguistics **49**(2), 245-270 (2023)
28. Kumar, R., Patel, D., Singh, M.: Intent Detection in Educational Chatbots: A Deep Learning Approach with Contextual Embeddings. International Journal of Artificial Intelligence in Education **33**(2), 289-312 (2023)
29. Rodriguez, M., Thompson, J.: Ethical Considerations in AI Chatbot Development: Framework and Implementation Guidelines. AI Ethics Journal **5**(3), 178-195 (2023)
30. Wang, L., Liu, X., Chen, G.: Hybrid Attention Mechanisms for Enhanced Chatbot Response Generation: A Comparative Study. Neural Computing and Applications **35**(4), 3456-3471 (2023)