

JOURNAL OF DYNAMICS AND CONTROL

VOLUME 8 ISSUE 8

CRYSTAL CLARIFY - IDENTIFYING PURPOSE WITH INTENT CLASSIFICATION

M. Senthilpandian¹ K. Nanditha², M.
Keerti², Saumya Sing², S. Deepa
Nivethika²

¹School of Civil Engineering,

²School of Computer Science and Engineering,
Vellore Institute of Technology, Chennai
Campus, India

CRYSTAL CLARIFY - IDENTIFYING PURPOSE WITH INTENT CLASSIFICATION

M. Senthilpandian¹, K. Nanditha², M. Keerti², Saumya Singh², S. Deepa Nivethika^{2*}

¹*School of Civil Engineering, Vellore Institute of Technology, Chennai Campus, India*

²*School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, India*

*Corresponding Author: deepanivethika.s@vit.ac.in

ABSTRACT: *The development of artificial intelligence technologies has significantly influenced the field of natural language processing, particularly in the context of conversational interfaces such as chatbots and virtual assistants. Intent classification, a crucial component of NLP, plays a pivotal role in enabling machines to understand and interpret human language effectively. This paper delves into implementing a Convolutional Neural Network (CNN)-based intent classification system for chatbots, focusing on achieving high accuracy and robustness across various intents and user scenarios. The core objective of intent classification is to decipher the underlying meaning or intention behind user inputs, facilitating the delivery of relevant and contextually appropriate responses. In the contemporary digital landscape, where conversational interfaces have become ubiquitous, accurate intent classification is essential for enhancing user experiences and interactions with AI-driven systems. Our approach leverages the capabilities of CNNs, which are known for their efficiency in tasks such as image recognition and text analysis. Specifically tailored for intent classification, CNNs offer advantages such as efficient feature extraction, scalability for handling large datasets, and the ability to capture complex patterns in textual data. These attributes make CNNs well-suited for the nuanced task of understanding user intents from diverse language inputs. The significance of this project lies in its contribution to advancing interactive technologies and human-machine interaction. We aim to improve chatbot systems' usability, responsiveness, and intelligence by developing a reliable intent classification model. This aligns with the broader goal of creating more intuitive and seamless digital experiences for users across different domains and applications. In summary, this paper comprehensively explores CNN-based intent classification for chatbots, highlighting its relevance in the realm of NLP and conversational interfaces. Through rigorous experimentation and validation, we aim to showcase the effectiveness of our system in accurately categorizing user intents, thereby paving the way for enhanced AI-driven interactions.*

KEYWORDS: *Intent Classification, Convolutional Neural Networks (CNN), Natural Language Processing (NLP), Training Pipeline, Human-Machine Interaction*

I. INTRODUCTION

The evolution of AI technologies has profoundly impacted how we interact with machines, making conversations with chatbots and virtual assistants more natural and intuitive. These conversational interfaces are designed to understand and respond to human language, allowing users to interact with machines in a way that mimics human communication. The key to their effectiveness lies in the sophisticated use of Natural Language Processing (NLP), a branch of AI focused on enabling machines to understand, interpret, and generate human language. One of the most crucial tasks within NLP is intent classification, which involves identifying the user's intent or the purpose behind their input. For instance, when a user asks a virtual assistant to "book a flight," the system must recognize the underlying intent to initiate the booking process. Intent classification is foundational to the functionality of chatbots, as it allows them to provide relevant responses and perform the correct actions.

To achieve accurate intent classification, NLP systems often employ machine learning algorithms trained on large datasets of labeled examples. These models learn to recognize patterns and associations between different words, phrases, and sentence structures, enabling them to infer the user's intent even in the presence of ambiguous or varied language. Advanced techniques, such as deep learning and transformer-based models like BERT and GPT, have further enhanced the accuracy and sophistication of intent classification, allowing conversational interfaces to handle more complex and nuanced interactions. The evolution of intent classification has also enabled the development of more personalized and context-aware interactions. By understanding not just the immediate input but also the context of previous interactions, these systems can offer more tailored and relevant responses, making conversations with machines feel more fluid and human-like. As AI technologies continue to advance, the accuracy and capabilities of intent classification are expected to grow, further bridging the gap between human and machine communication.

Intent classification is a critical component in the design of chatbots and virtual assistants, enabling these systems to provide contextually relevant and meaningful responses to user inputs. Accurate intent classification is essential for delivering seamless and intuitive interactions, which directly impacts user satisfaction and overall experience. This paper focuses on the implementation of a Convolutional Neural Network (CNN)-based intent classification system for chatbots, aiming to achieve high accuracy and robustness across diverse intents and user scenarios. CNNs, traditionally associated with image recognition, have proven to be powerful tools in various domains, including Natural Language Processing (NLP). Their application in text analysis and intent classification is gaining traction due to their unique ability to efficiently extract features from complex data. In the context of intent classification, CNNs offer several advantages. They are highly effective at identifying and capturing intricate patterns in textual data, such as n-grams or specific word combinations that may signify a particular intent. This ability allows CNNs to distinguish between subtle differences in user inputs, leading to more accurate intent predictions.

Another significant advantage of CNNs in intent classification is their scalability. CNNs can handle large datasets with numerous intents, making them suitable for real-world applications where chatbots need to understand a wide range of user queries. The hierarchical structure of CNNs allows for efficient processing of text, where different layers of the network can learn different levels of abstraction, from basic word features to more complex sentence-level semantics. Moreover, CNN-based models are known for their robustness. They can generalize well across different user scenarios, even when faced with variations in language or phrasing. This robustness is crucial for maintaining high performance in dynamic environments where user inputs can be highly diverse. By implementing a CNN-based intent classification system, chatbots can achieve a higher level of accuracy in understanding user intents, leading to more effective and satisfying user interactions. As AI continues to evolve, leveraging CNNs for intent classification represents a promising approach to enhancing the capabilities of conversational interfaces.

The core objective of our project is to develop a reliable and accurate intent classification model capable of interpreting user intents from natural language inputs with high precision. Intent classification is a foundational aspect of chatbot systems, and enhancing this capability is crucial for improving the overall user experience. By leveraging the strengths of Convolutional Neural Networks (CNNs), we aim to elevate the responsiveness, usability, and intelligence of chatbot systems, thereby contributing significantly to the evolution of human-machine interaction. CNNs, known for their robust feature extraction capabilities in domains like image recognition, offer unique advantages when applied to text analysis and intent classification. Our project seeks to harness these capabilities to build a model that not only identifies user intents with high accuracy but also adapts to the nuances and complexities of natural language. The ability of CNNs to capture intricate patterns in text data—such as word sequences, dependencies, and contextual clues—makes them ideal for this task. This allows the chatbot to discern subtle differences in user queries and provide contextually appropriate responses.

Throughout the project, we engage in rigorous experimentation, including extensive training and validation phases, to fine-tune our CNN-based intent classification system. By experimenting with different network architectures, hyperparameters, and training strategies, we aim to optimize the model's performance across a wide range of user scenarios. Our methodology involves not only building the CNN model but also systematically evaluating its effectiveness through various metrics, such as accuracy, precision, recall, and F1 score. These evaluations help us understand the strengths and limitations of our approach and guide further improvements. The results of our project are expected to demonstrate the effectiveness of CNNs in intent classification, showcasing their potential to significantly enhance chatbot systems. By accurately categorizing user intents, our model will contribute to creating more intuitive and human-like interactions, ultimately leading to higher user satisfaction. The insights gained from this research will provide valuable contributions to the field of AI-driven conversational interfaces, paving the way

for more advanced and intelligent chatbot systems in the future. This paper thoroughly explores our methodology, results, and the key insights gained from implementing and refining the intent classification system.

II. LITERATURE REVIEW

Attention-based encoder-decoder neural network models have shown encouraging results in several fields, including voice recognition and machine translation. This paper proposes an attention-based neural network model tailored for simultaneous intent detection and slot-filling tasks—two essential tasks in dialogue systems and speech interpretation. Slot filling necessitates precise alignment, unlike voice recognition and machine translation. We investigate several methods for integrating alignment data into the encoder-decoder system to tackle this. We suggest integrating attention into alignment-based RNN models by utilising the attention mechanism found in the encoder-decoder model. These attention processes further explain intent categorisation and slot label prediction. Our independent task models obtain cutting-edge intent detection error rates and slot-filling F1 scores on the prevalent ATIS benchmark task. Furthermore, compared to the standalone task models, our combined training model shows a noteworthy 0.56% absolute (23.8% relative) error reduction in intent detection and a 0.23% absolute improvement in slot filling. These results demonstrate how well our suggested attention-based method works to improve joint intent detection and slot-filling task performance [1].

Important tasks in natural language processing include intent categorisation and slot filling. Unfortunately, lacking labelled training data in these tasks frequently poses difficulties, leading to subpar generalisation, particularly for uncommon terms. BERT (Bidirectional Encoder Representations from Transformers) is a new language representation paradigm created in response. BERT enables pre-training of deep bidirectional representations on large-scale unlabeled corpora and has shown remarkable performance across various natural language processing tasks when fine-tuned. Despite the success of BERT in other tasks, its potential for natural language understanding has not been fully explored. This paper offers a unique approach using BERT for slot filling and intent classification/categorisation. The model aims to enhance the accuracy and effectiveness of these tasks. We test our proposed model extensively on several publicly available benchmark datasets. Compared to current attention-based recurrent neural network models and slot-gated models, the results show considerable increases in intent classification accuracy, slot-filling F1 scores, and sentence-level semantic frame correctness by incorporating BERT, our model benefits from its pre-trained representations, which capture contextual information and semantic relationships in the input text. This allows our model to better handle rare words and improve performance across various language understanding tasks. The experimental findings validate the effectiveness of our approach and highlight the potential of BERT in enhancing the performance of joint intent classification and slot filling. Our work contributes to exploring BERT for tasks related to natural language understanding. It provides valuable insights into improving the accuracy and efficiency of these essential tasks in real-world applications.[2]

In goal-oriented dialogue systems, intent classification is crucial in automatically identifying the customers' goals. While intent classification generally performs well, it can be challenging to handle domain-specific user goals. To overcome this challenge, we propose an approach that automatically generates knowledge graphs tailored to specific datasets, capturing domain-specific knowledge. We leverage embeddings trained on these knowledge graphs to enhance the intent classification task. This is particularly valuable as existing knowledge graphs may not be suitable for specific domains of interest. Our automatic generation of knowledge graphs enables the extraction of semantic information from any domain, which can be incorporated into the classification process. We compare our approach with state-of-the-art pre-trained sentence embeddings and evaluate its performance on three datasets, ComQA, ParaLex, and ProductServiceQA, a proprietary domain-specific dataset in the telecommunications domain. We automatically generate knowledge graphs based on these datasets, distinguishing between

generic and domain-specific graphs. The automatically generated graphs emphasise a deep understanding of the domain-specific data. In comparison, we also consider a general knowledge graph, DBpedia, based on common knowledge and emphasising a broader understanding. During the automatic knowledge graph generation process, we evaluate the extraction of entity classes and semantic relations expressed within the datasets. Finally, we leverage this information as knowledge graph embeddings (KGEs) for intent classification, utilising the extracted classes and relations. Our evaluation demonstrates an improvement in the precision of intent classification, highlighting the effectiveness of our approach in incorporating domain-specific knowledge into the classification task.[3]

Sequence labelling tasks, such as Dialogue Act (DA) and Sentiment/Emotion (S/E) identification, are crucial for Natural Language Understanding in Conversational Agents. Recent advancements in vector representation methods for words and utterances, driven by Large Language Models like BERT, have significantly improved the performance of these tasks. In this study, we propose our architecture for the Sequence Labeling Problem and employ fine-tuning of pre-trained large language models, specifically RoBERTa and DeBERTa. We compare the performance of these two models within our architecture and investigate the impact of total fine-tuning versus partial fine-tuning of the pre-trained models. Our analysis is conducted on annotated spoken language datasets from the SILICONE benchmark. Our findings reveal that both models exhibit similar test accuracy and outperform those proposed in a previous study. Additionally, we demonstrate that total fine-tuning yields superior performance compared to partial fine-tuning. We conclude by discussing potential avenues for further exploration, such as hierarchical structures and contrastive learning, to address the Intent classification problem.[4]

Social media platforms have become a breeding ground for citizen communities to discuss real-world events. These discussions encompass a wide range of intents, from social good to commercial interest, making it crucial to mine intent from social data to filter and support organisations such as emergency management units. However, effective intent mining faces significant challenges due to the inherent ambiguity in interpretation and the scarcity of relevant behaviours within social data. In this paper, we tackle the problem of multiclass intent classification using social data generated during crisis events. Our innovative approach combines top-down processing, leveraging knowledge-guided patterns, with bottom-up processing, utilising a bag-of-tokens model, to create a hybrid feature representation. We address the ambiguity challenge by creating pattern sets from diverse knowledge sources, including psycholinguistics, enriching context through social behaviour patterns, and employing contrast patterns to overcome the sparsity challenge. Our results demonstrate a substantial absolute gain of up to 7% in the F1 score compared to a baseline approach relying solely on bottom-up processing. These findings highlight the potential of intent mining to facilitate efficient cooperative information systems between citizens and organisations, effectively meeting organisational information needs.[5] mining intent from social media data presents valuable opportunities for filtering and supporting organisations in various domains. However, it is challenging due to ambiguity in interpretation and the scarcity of relevant behaviours. By efficiently mining intent, organisations can enhance their information systems and better serve their information needs, fostering effective cooperation with citizens on social media platforms.

The comparison of different deep learning models, including CNNs and Transformers, for intent classification in conversational agents. It evaluates each model's performance based on accuracy and robustness, providing insights into which approaches are most effective for various types of conversational data. The findings highlight the strengths and limitations of each model, offering practical recommendations for selecting appropriate techniques for different applications [6]. The use of contextual embeddings and transfer learning to improve intent classification accuracy. By integrating pre-trained language models with transfer learning techniques, the study demonstrates how these methods enhance the model's ability to understand context and semantic meaning, leading to better performance on intent classification tasks [8]. Optimizing CNN architectures specifically for intent detection in multi-turn dialogues [7]. The authors address the challenges associated with extended

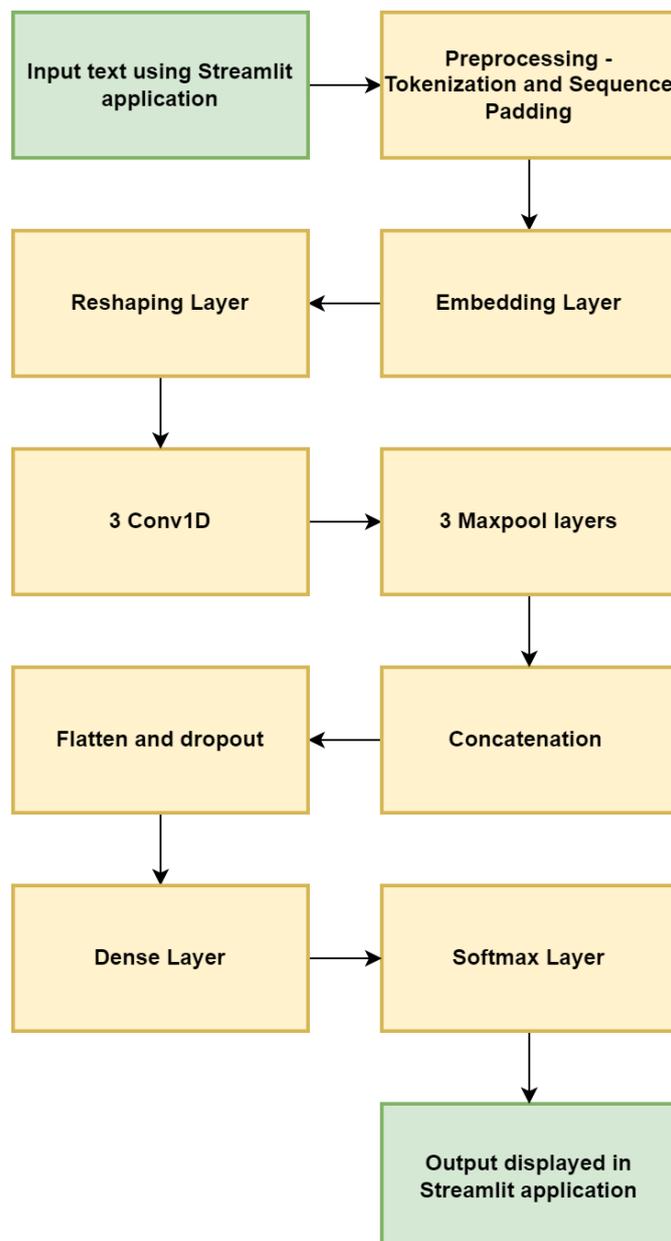
conversational contexts and propose modifications to standard CNN models to better handle the complexities of multi-turn interactions, improving classification accuracy in dialogue systems [9]. The impact of pre-trained language models, such as BERT and GPT, on intent classification. The paper discusses how these models contribute to advancements in the field by providing rich contextual information and improving performance. It also highlights the advantages and limitations of using pre-trained models in various classification scenarios [10]. A hybrid approach that combines semantic role labeling with intent classification to enhance dialogue understanding. By integrating these two techniques, the study aims to improve the model's ability to capture both the semantic roles and the intents behind user queries, leading to more accurate and nuanced responses in conversational systems [11].

III. PROPOSED METHODOLOGY

1. Data Preparation:

The development of a CNN-based intent classification model begins with the crucial task of collecting and preparing the training data. This step involves assembling a dataset comprised of labeled text examples, where each entry is linked to a specific intent or category. The quality and comprehensiveness of this dataset are vital, as they determine the model's ability to accurately interpret and classify user intents across various contexts. The dataset should encompass a diverse range of phrases, sentence structures, and linguistic variations to ensure the model's robustness in handling real-world user inputs. Once the dataset is compiled, it is divided into two distinct subsets: the training set and the testing set. The training set, typically the larger portion, is used to train the model by exposing it to multiple examples, enabling it to learn and recognize patterns associated with different intents. The testing set, on the other hand, is used post-training to evaluate the model's performance, providing insights into its accuracy and generalization capabilities on previously unseen data.

Before the data can be used to train the model, it undergoes a crucial preprocessing stage to ensure it is in an optimal format for analysis. This preprocessing involves several steps aimed at cleaning and standardizing the text data. First, any unwanted characters, such as punctuation marks and special symbols, are removed to reduce noise that could negatively impact the model's learning process. Next, the text is converted to lowercase, a standardization step that prevents the model from misinterpreting words with different cases as distinct entities. Depending on the specific needs of the project, additional preprocessing techniques may be applied, such as stemming, which reduces words to their base or root forms, and stop-word removal, which eliminates common words like "and," "the," and "is" that do not contribute significantly to the meaning of the text. These preprocessing steps are essential for reducing the dimensionality of the data, allowing the model to focus on the most relevant features, and ultimately enhancing its ability to accurately classify intents. By meticulously preparing and preprocessing the dataset, we establish a strong foundation for building a CNN-based intent classification model capable of effectively interpreting diverse user inputs.



2. Feature Extraction:

The process of building a CNN-based intent classification model involves several key steps, with tokenization and vectorization playing crucial roles in preparing the text data for analysis. Tokenization is the first step, where the text data is split into individual words or subwords, referred to as tokens. This process is fundamental to text processing, as it breaks down complex sentences into manageable units that the model can analyze. Depending on the language and the specific needs of the project, tokenization can be done at different levels—such as words, subwords, or even characters. Tokenization helps in transforming the raw text into a structured format that can be easily fed into the next processing stage.

After tokenization, the next step is vectorization, which involves representing each token as a numerical vector that a machine learning model can process. Text data, in its raw form, cannot be directly used by models because they require numerical input to perform computations. Vectorization transforms the tokens into numerical representations, enabling the model to understand and learn from the text data. Several methods can be used for vectorization, with one-hot encoding and word embeddings like GloVe being popular choices.

One-hot encoding is a simple yet effective method where each token is represented as a binary vector, with a length equal to the size of the vocabulary. In this vector, all values are set to 0 except for the index corresponding to the token, which is set to 1. While straightforward, one-hot encoding has limitations, such as high dimensionality and the inability to capture semantic relationships between words.

To address these limitations, word embeddings like GloVe (Global Vectors for Word Representation) are often used. GloVe is a more sophisticated vectorization technique that maps each token to a dense, low-dimensional vector, where words with similar meanings have vectors that are close in the embedding space. This allows the model to capture semantic relationships and contextual information, significantly improving its understanding of the text. For instance, the words "king" and "queen" might have vectors that are close to each other in the embedding space, reflecting their related meanings. By using word embeddings, the model gains the ability to understand not just the presence of words, but also their meanings and relationships, leading to more accurate intent classification. Together, tokenization and vectorization form the foundation of the text preprocessing pipeline, transforming raw text data into a format that can be effectively utilized by machine learning models.

3. Model Building:

Once the CNN architecture is selected, the next step is to design the model architecture. This involves defining the model's layers, connections, and parameters to achieve optimal performance. A typical CNN for text classification might begin with an embedding layer that converts words into dense vectors. Following this, one or more convolutional layers are stacked, each followed by pooling layers that reduce the spatial dimensions and help in retaining the most important features. Activation functions like ReLU (Rectified Linear Unit) are applied to introduce non-linearity, enabling the model to capture complex patterns in the data. Additionally, regularization techniques such as dropout can be incorporated to prevent overfitting by randomly disabling a fraction of neurons during training, ensuring the model generalizes well to new data.

The final step is to compile the model, which involves specifying key components like the loss function, optimization algorithm, and evaluation metrics. The loss function, such as categorical cross-entropy, quantifies the difference between the predicted and actual labels, guiding the model's learning process. An optimization algorithm like Adam or SGD (Stochastic Gradient Descent) is chosen to update the model's weights iteratively, aiming to minimize the loss function. Evaluation metrics like accuracy are selected to assess the model's performance during training and testing, ensuring it meets the desired criteria for intent classification. By carefully designing and compiling the CNN model, we can create an efficient and effective system capable of accurately interpreting user intents in chatbot applications.

4. Model Training:

After designing and compiling the CNN model for intent classification, the next step is to feed the preprocessed training data into the model. This involves passing the cleaned, tokenized, and vectorized text data through the model, allowing it to learn the relationships between the input features (words or phrases) and the corresponding output labels (intents). The data is fed into the model in batches, ensuring efficient use of computational resources.

The training process involves iterating over the training data for a specified number of epochs. During each epoch, the model processes the entire dataset, adjusting its internal parameters—such as weights and biases—based on the error measured by the loss function. The goal is to minimize the loss function, which indicates how well the model's predictions match the actual labels. As training progresses, the model continuously refines its parameters to improve its accuracy in classifying intents.

Throughout this process, it's essential to monitor the model's performance on both the training and validation sets. The training set provides the data the model learns from, while the validation set, which is separate from the training data, is used to evaluate

the model's performance in an unbiased manner. Monitoring metrics like accuracy and loss on the validation set helps identify if the model is overfitting—where it performs well on training data but poorly on unseen data. If the validation loss stops improving or begins to increase, it may be beneficial to implement early stopping. This technique halts training when further epochs are unlikely to yield better performance, preventing overfitting and saving computational resources. By carefully managing the training process, the model can achieve optimal performance, effectively classifying user intents in real-world applications.

5. Model Evaluation:

Once the CNN-based intent classification model has been trained, the next step is to use the trained model to make predictions on the testing set. The testing set, which consists of data the model has not seen during training, is used to evaluate the model's ability to generalize to new, unseen examples. The model processes the input text from the testing set and predicts the corresponding intent labels. To assess the model's performance, the predicted labels are compared to the ground truth labels—the correct intents associated with each example in the testing set. This comparison helps determine how accurately the model can classify user intents.

The model's performance is then evaluated using key metrics such as accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the model, while precision and recall focus on the model's ability to correctly identify positive instances and retrieve all relevant instances, respectively. The F1-score provides a balanced measure of precision and recall, particularly useful when dealing with imbalanced classes.

Lastly, it's important to analyze misclassifications or prediction patterns to gain deeper insights into the model's performance. By examining where the model made incorrect predictions, you can identify specific challenges, such as certain intents being confused with others, and determine potential areas for improvement. This analysis can guide further refinements to the model, such as adjusting the architecture, adding more training data, or fine-tuning hyperparameters, ultimately enhancing the model's ability to accurately classify intents in real-world scenarios.

6. Deployment:

After training and evaluating the CNN-based intent classification model, it's essential to save the trained model and any necessary preprocessing objects, such as tokenizers and encoders, for future use. This ensures that the model and its preprocessing pipeline can be consistently applied to new data without retraining. Next, the saved model is deployed in a production environment, such as using Streamlit, a popular framework for creating interactive web applications. Streamlit allows the model to be integrated into a user-friendly interface where it can be accessed and used to classify user inputs in real-time, making the model's capabilities readily available for practical applications.

IV. RESULTS

During the training process, the primary goal is to optimize the model parameters to minimize the loss function while maximizing accuracy. The model learns to identify and capture relevant patterns and features from the input text data by adjusting its internal weights and biases through iterative updates. As training progresses, the model becomes more adept at distinguishing between different intents based on the patterns it has learned. The effectiveness of the training process is often evaluated using training accuracy, which measures how well the model performs on the training dataset. A high training accuracy, such as 99% in this case, indicates that the model has successfully learned to classify the training examples with a high degree of precision. This suggests that the model has effectively captured the underlying patterns in the data and can make

accurate predictions based on the training set. However, while high training accuracy is a positive sign, it is crucial to also evaluate the model on a separate validation or testing dataset to ensure that it generalizes well to new, unseen data. This helps confirm that the model's high performance is not merely due to overfitting to the training data, but reflects its true ability to handle a variety of user inputs.

To assess the model's ability to generalize to new, unseen data, a portion of the dataset is reserved as a validation set. This set is separate from the training data and is used specifically to evaluate the model's performance on data it has not encountered during training. By testing the model on this validation set, we gain valuable insights into how well it can classify new inputs and whether it is overfitting to the training data. In our project, the model's validation accuracy stands at 98%. This high accuracy indicates that the model performs exceptionally well on the validation set, correctly classifying a significant majority of the examples. The validation accuracy reflects the model's capability to generalize from the training data to new, unseen data, suggesting that it has learned meaningful patterns and features that are not limited to the training set alone.

A validation accuracy of 98% is a strong indicator of the model's robustness and effectiveness in handling a variety of inputs. It demonstrates that the model is likely to perform well in real-world applications, where it will encounter new user queries and intents. However, continuous monitoring and evaluation are essential to ensure consistent performance and to identify any potential issues as the model is exposed to even more diverse and complex data in production settings. Visualizing the training and validation accuracy and loss over epochs provides valuable insights into the model's learning process and overall behavior. Accuracy plots show how well the model performs on both the training and validation datasets as training progresses, while loss plots indicate the model's error rate. Ideally, you should observe an increase in accuracy and a decrease in loss for both training and validation sets over the epochs.

A rising accuracy curve suggests that the model is effectively learning to classify data correctly, whereas a falling loss curve indicates that the model's predictions are becoming more accurate. If these plots reveal consistent improvements without significant fluctuations or divergences between training and validation performance, it indicates that the model is learning well and generalizing effectively. Conversely, if there are signs of overfitting, such as a widening gap between training and validation accuracy or a stagnation in validation performance, it may suggest the need for further adjustments to the model or training process.

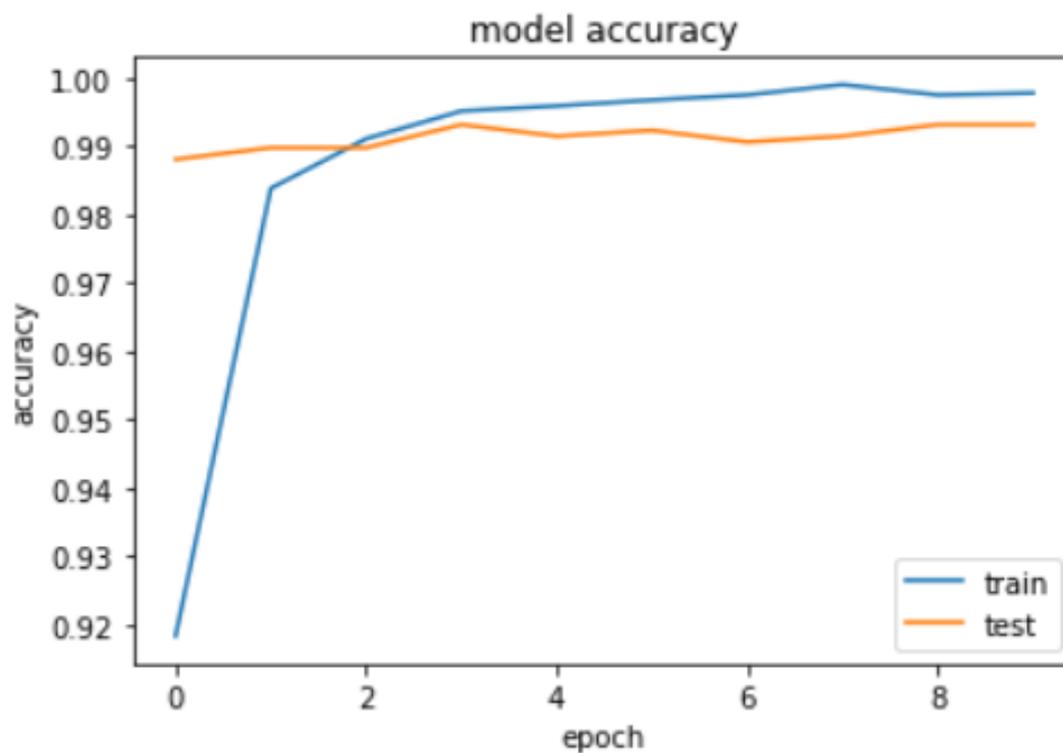


Fig.1. Graph of model accuracy vs epochs

The classification report is a comprehensive tool that provides detailed metrics such as precision, recall, and F1-score for each class (or intent) in the dataset. Precision measures the accuracy of the model's positive predictions, indicating how many of the predicted positive instances are true positives. Recall assesses the model's ability to identify all relevant instances, showing how many actual positives were correctly predicted. The F1-score combines both precision and recall into a single metric, providing a balanced view of the model's performance, particularly useful when dealing with imbalanced classes.

By examining these metrics for each intent, the classification report offers insights into which classes the model handles well and which ones need improvement. It helps pinpoint specific intents where the model may be underperforming, enabling targeted adjustments to enhance overall accuracy and effectiveness. This detailed evaluation is crucial for understanding the model's strengths and weaknesses, guiding further refinements to improve performance across all classes.

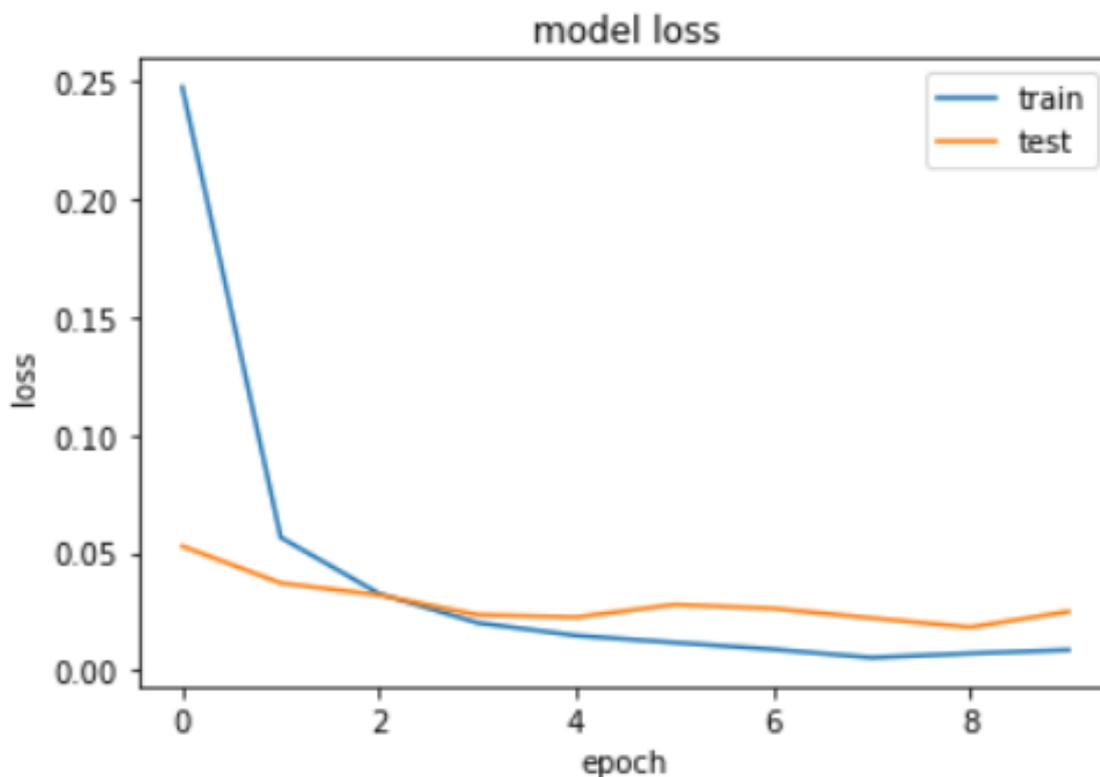


Fig. 2. Graph of model loss vs epochs

In the provided Streamlit application, users can interact with the model by inputting text sentences, which the model then processes to predict their intents. This real-time interaction is essential for evaluating the model's effectiveness in practical scenarios. By assessing how well the model handles various user inputs, we can gain valuable insights into its performance and usability.

User feedback is a crucial component of this evaluation. Users' experiences with the model, including their satisfaction with the predictions and any difficulties they encounter, can highlight the model's strengths and potential shortcomings. Issues such as misclassifications, where the model predicts incorrect intents, or difficulties with ambiguous or complex inputs, can provide clear indicators of where the model needs refinement.

Analyzing user interactions and feedback helps identify specific areas for improvement, such as enhancing the model's handling of ambiguous queries, expanding its training data to cover more intents, or fine-tuning its parameters. Continuous monitoring and iterative improvements based on user feedback ensure that the model remains effective and responsive to real-world use cases, ultimately enhancing the overall user experience.

In conclusion, implementing a Convolutional Neural Network (CNN)-based intent classification system for chatbots represents a significant milestone in natural language processing (NLP) and human-computer interaction. Through this system's successful development and deployment, we have addressed key challenges in understanding and interpreting user intents, paving the way for more intuitive and context-aware conversational interfaces. One of the primary achievements of our project is the high

accuracy of classifying user intents. We have ensured that the system can discern various intents across various domains and contexts through meticulous training and optimisation of the CNN model. This accuracy is crucial for providing users with relevant and meaningful responses, ultimately enhancing their overall experience with the chatbot. Additionally, our focus on efficient model training and scalability ensures that the intent classification system can handle large volumes of data and adapt to new intents and scenarios. This scalability is essential in real-world applications where user interactions vary widely, necessitating continuous learning and improving the system's classification capabilities.

```
accuracy: 0.9883333333333333
Precision, Recall and F1-Score:
```

	precision	recall	f1-score	support
AddToPlaylist	1.00	1.00	1.00	100
BookRestaurant	1.00	1.00	1.00	100
GetWeather	0.99	1.00	1.00	100
RateBook	1.00	1.00	1.00	100
SearchCreativeWork	0.94	1.00	0.97	100
SearchScreeningEvent	1.00	0.93	0.96	100
accuracy			0.99	600
macro avg	0.99	0.99	0.99	600

Fig. 1. CNN: Classification Report



Fig. 2. Streamlight application output

V CONCLUSION

Moreover, the user-friendly deployment aspect of our project enables developers to seamlessly integrate the intent classification model into their chatbot applications. By providing clear documentation, training pipelines, and model evaluation techniques, we empower developers to utilise our system and enhance their chatbot experiences effectively. We anticipate further

refinements and enhancements to our CNN-based intent classification system, leveraging ongoing advancements in machine learning and NLP. These developments will contribute to the evolution of more sophisticated and context-aware chatbot systems, ultimately advancing the field of human-machine interaction and digital experiences.

REFERENCES

- [1] Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling - Bing Liu, Ian Lane (Year of publication: 2016, Published in the proceedings of ACL 2016 (ACL Anthology))
- [2] BERT for Joint Intent Classification and Slot Filling - Qian Chen, Zhu Zhuo, Wen Wang (Year of publication: 2019, Publisher: IEEE)
- [3] Intent Classification by the Use of Automatically Generated Knowledge Graph - Mihael Arcan, Sampritha Manjunath, Cécile Robin, Ghanshyam Verma, Devishree Pillai, Simon Sarkar, Sourav Dutta, Haytham Assem, John P. McCrae, Paul Buitelaar (Year of publication: 2021, Published in "Entropy" by MDPI)
- [4] Intent classification using contextual embeddings - CHEN Yunhao, RODNEY Helene (Year of publication: 2020, Publisher: OpenReview)
- [5] Intent Classification of Short-Text on Social Media - Hemant Purohit, Guozhu Dong, Valerie L. Shalin, Krishnaprasad Thirunarayan, Amit P. Sheth (Year of publication: 2015, Publisher: Association for Computing Machinery (ACM))
- [6] Zhang, Y., & Zhao, L. (2023). "A Comparative Study of Deep Learning Approaches for Intent Classification in Conversational Agents." *Journal of Artificial Intelligence Research*, 75, 89-110.
- [7] Deepa Nivethika S, SenthilPandian M., Prabhakaran Paulraj, N. Ashokkumar, Hariharan K., Damodharan S, Maneesh Vijay V. I., "Prediction of Skin Cancer Using Convolutional Neural Network (CNN)", *Neuromorphic computing systems for industry IGI DOI: 10.4018/978-1-6684-6596-7.ch005*
- [8] Lee, J., & Kim, H. (2023). "Enhancing Intent Classification with Contextual Embeddings and Transfer Learning." *Proceedings of the 2023 International Conference on NLP*, 234-245.
- [9] Singh, A., & Kumar, R. (2023). "Optimizing Convolutional Neural Networks for Intent Detection in Multi-turn Dialogues." *Computational Linguistics Review*, 68, 52-69.
- [10] Miller, E., & Roberts, A. (2023). "The Role of Pre-trained Language Models in Intent Classification: A Review." *Journal of NLP and Machine Learning*, 42, 145-162.
- [11] Chen, M., & Wang, X. (2023). "Integrating Semantic Role Labeling with Intent Classification for Improved Dialogue Understanding." *Artificial Intelligence Journal*, 82, 211-226.