Project Report : CS 5660

Mujahir H Abbasi
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
mabbasi4@calstatela.edu

Abir Sur
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
asur2@calstatela.edu

Rutwik V Wagh
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
rwagh@calstatela.edu

Shivam S Jadhav
California State University, Los Angeles
5151 State University Dr, Los Angeles, CA
sjadhav5@calstatela.edu

Abstract

Intent classification is useful to understand the intentions behind customer queries, automate processes, and gain valuable insights. Most supervised machine learning problems have a target label assigned to that, but what happens when this trained algorithm sees a data which very far from any of the target variable, this is out-of-scope classification. Ultimately, every customer interaction has a purpose, an aim, or intention. Whether they want to make a purchase, request more information, or unsubscribe, you should respond quickly to improve customer retention rates, loyalty, and satisfaction.Recently, many deep learning-based methods have been widely used in skin cancer classification to solve the above issues and achieve satisfactory results.Therefore, in this article, we provide a comprehensive overview of the latest deep learning-based algorithms for customer's intent classification. We begin with an overview of three types of customer intents, followed by a list of publicly available datasets relating to intent classification.This study focuses on the idea of using a deep learning method for intent classification.

1. Introduction/Background/Motivation

The main objective of this project is to classify the customer queries into specific intents, so that customer can be guided to the correct path and it ultimately improves customer satisfaction.There are has been already a lot of work done using machine learning algorithms like Naive Bayes, Logistic Regression, and tree based models like DecisionTree, RandomForest and XgBoost.

We are planning to use Natural Langauge Processing (NLP) technique and a deep learning-based algorithm called Long Short Term Memory(LSTM). Although different algorithm has been applied to solve this problem, but the goal of machine learning is to make any predictions as much accurate as possible. We have focused more on the feature engineering and model tuning part for enhancement of the performance of the model. When a user input is classified as out of scope, it means that the system cannot determine the appropriate intent or category to assign to that input because it doesn't fit within the defined domain or range of topics the system is trained on. Out-of-scope classification helps the system recognize when it encounters inputs that it is not equipped to handle, allowing it to provide appropriate responses such as indicating that it doesn't understand the request or suggesting alternative actions.

Implementing out-of-scope intent classification can be useful for improving the user experience by managing user expectations and avoiding misleading or incorrect responses when faced with unfamiliar or unrelated queries.Out-of-scope intent classification is typically implemented using machine learning techniques, similar to how in-scope intent classification is performed.

The NLU system is trained on a labeled dataset that

includes examples of both in-scope and out-of-scope user queries or statements. These examples are annotated with the appropriate intent or label, indicating whether they are within the system's defined domain or outside of it. While current practice in out-of-scope intent classification has made significant progress like data collection and generalization.

Intent classification are used in conversational AI applications to provide personalized conversation experiences to users. It helps to increase sales and improve overall customer experience. It allows businesses to understand customers' intent and give a more accurate response to their customers.

It can allow businesses to automate the interaction between interested buyers and business representatives. It classifies customers' needs and their special attention then analyzes what customers intend to achieve, and directs them to the relevant representative by categorizing customers' intents.

We have used the dataset mentioned here. It has both the in-scope as well as out-of-scope class. The in-scope datasets are further divided into specific intents.

2. Approach

We have approached this problem as a supervised deep learning classification by preparing the data in such way that is balanced. Gathering a comprehensive and diverse labeled datasets for out-of-scope examples can be challenging. It requires capturing a wide range of possible out-of-scope inputs, which can vary greatly depending on the domain or application.

2.1. LSTM

LSTM is a type of recurrent neural network but is better than traditional recurrent neural networks in terms of memory. Having a good hold over memorizing certain patterns LSTMs perform fairly better.We have a 4-layered architecture consisting of embedding, LSTM and the dense layer.The classification layer that we have use is sigmoid since its a binary classification problem with binary cross entropy as the loss function. The optimising function here is adam.

(5 points) What problems did you anticipate? What problems did you encounter? Did the very first thing you tried work?

Since we had a unbalanced dataset problem, we have used down sampling to balance that.In an unbalanced dataset, one class may have significantly fewer samples than the other. This lack of representation can make it challenging for deep learning models to learn patterns and features associated with the minority class

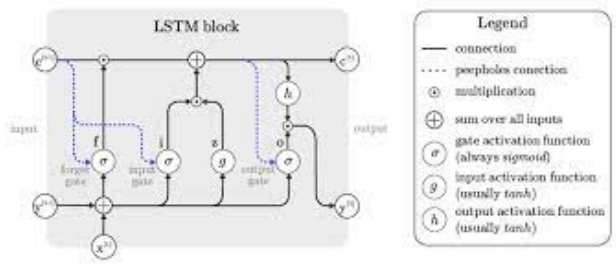

Figure 2.1. A simple LSTM architecture

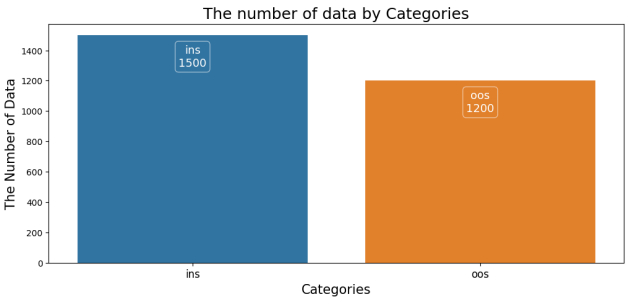

Figure 2.2. Data distribution

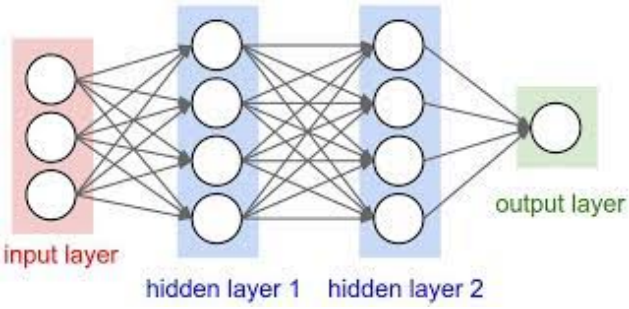

Figure 2.3. Dense Layer

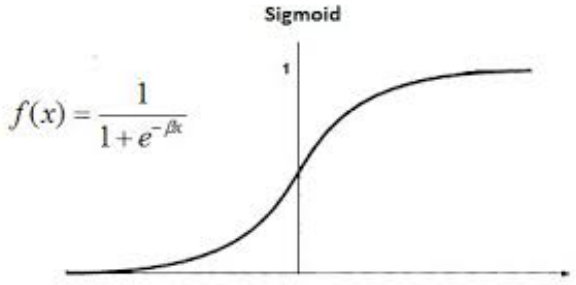

Figure 2.4. Sigmoid function

effectively. The model may become biased towards the majority class, leading to poor performance on the minority class. Downsampling is a technique used to address class imbalance in a dataset by reducing the number of samples from the majority class. It involves randomly selecting a subset of samples from the majority class to match the number of samples in the minority class.

But there is problem associated with this technique it that it may lead to loss of training data. And as unexpected this happened, so we had to trained the model further by increasing the number of epochs.We struggled initially with model but we were able to finedtuned that. And we haven't used any pretrained model, we have trained the LSTM architecture from scratch. We have used the code repository mentioned here here as a reference.

3. Experiments and Results

3.1. Model Training

The datasets that we have used for training has around 2.7K records out of which 1.2K belong to one class and 1.5K belong to the other class. A validation split of 0.1 has been used which makes 10 percent of training data available for cross validation while training.We are also monitoring the validation loss and accuracy during the training phase.At each epoch the validation loss and accuracy is plotted.

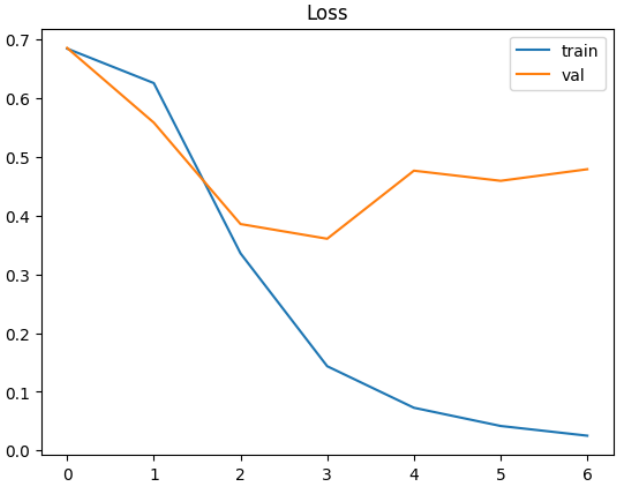

Figure 3.1. Validation Loss

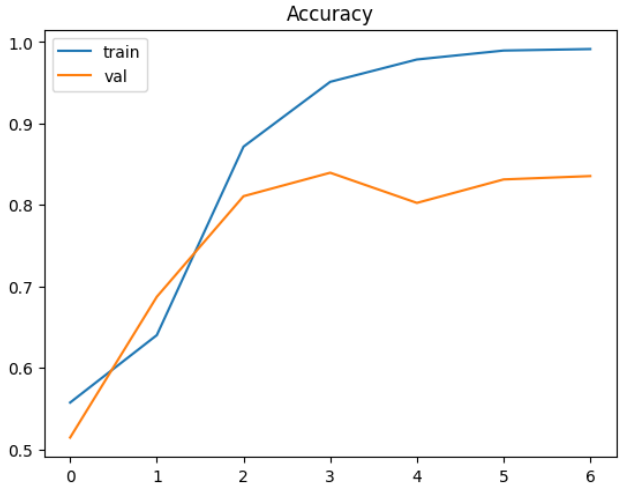

Figure 3.2. Validation Accuracy

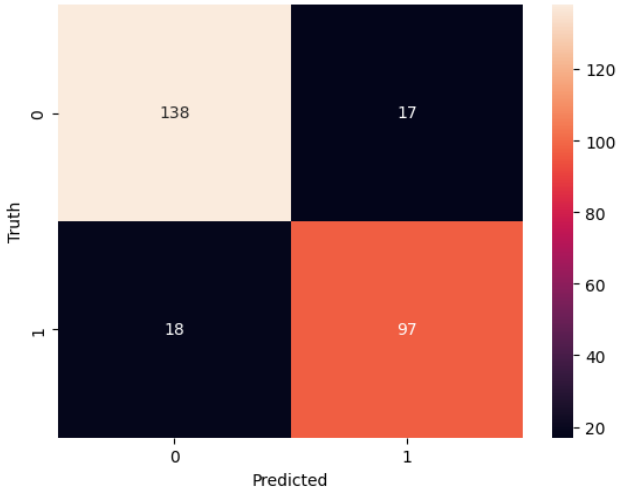

Figure 3.3. Confusion Matrix

3.2. Model Results

In this section, we have discussed about the performance of the model. The highest accuracy achieved is around 89 percent. The test dataset has is 20 percent of the total records.To analyze the result and performance of a classification model, we take the help of confusion matrix (CM). For Binary classification it has two columns and two rows suggesting actual positive, actual negative and predicted positive, predicted negative. The four sections are called True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN). These are essential to correctly analyze the model performance, for example the FP suggests that for how many records the model has classified them

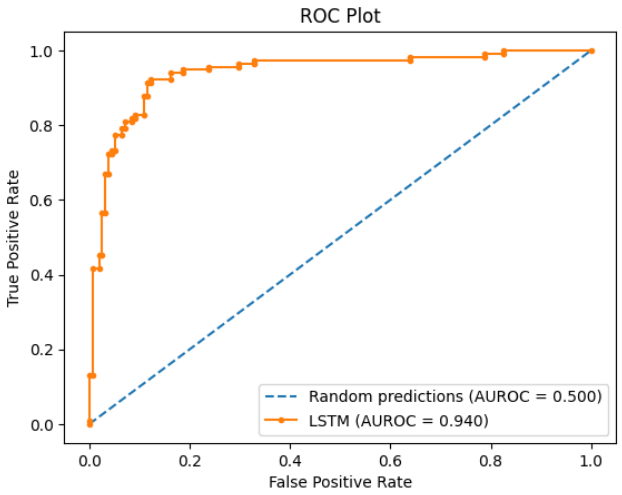

Figure 3.4. ROC-AUC curve

as Positive but they actually are Negative.The Precision, Recall, F1Score and Accuracy is also calculated to validate the performance of the model.

This parameters that can be calculated from the Confusion Matrix- Accuracy = (TP+TN)/(TP+FP+TN+FN) Precision (P) = TP / TP+ FP Recall(R) = TP/ TP+FN F1 score = 2*P*R/(P+R) The term Accuracy is quite obvious and needs little or no explanation, Recall is the ability of the model to detect the required class and Precision gives a measure of False Positive cases detected by the model. F1 score is the harmonic mean of Precision and Recall. Better precision, recall and F1 score are signs of a good classifier model.

We have also plotted the ROC-AUC curve,An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

Class	Precision	Recall	F1Score	Accuracy
oos	88	89	89	87
ins	85	84	85	87

Table 3.1. Metric Table

4. Work Division

The following shows the contribution of each of the group members:

- Mujahir Abbasi : Worked on designing and training of the LSTM model and Project Report. Defined the architecture of the LSTM model, incorporating additional layers like dropout or recurrent dropout to prevent overfitting. Utilized techniques such as mini-batch gradient descent and backpropagation through time to optimize the model's parameters.
- Abir Sur : Worked on data collection and preprocessing of the data and Project Report. Collect a labeled dataset that includes examples of both in-scope and out-of-scope intents. Preprocess the data by removing noise, normalizing text, and performing tokenization. Apply techniques like word embedding (e.g., Word2Vec, GloVe) to represent words as dense vectors and padding.
- Rutwik Wagh : Worked on evaluating and testing of the model. Assessed the model's performance using evaluation metrics such as accuracy, precision, recall, and F1 score on the validation set. Adjusted hyperparameters or modified the model if necessary to improve performance. Calculated the final evaluation metrics to assess the model's effectiveness in out-of-scope intent classification.
- Shivam Jadhav: Worked on testing and fine-tuning of the model. Calculated the final evaluation metrics to assess the model's effectiveness in out-of-scope intent classification. Refined the model based on insights from the evaluation results. Experimented with techniques like hyperparameter tuning, regularization, or different LSTM architectures to further optimize performance.

5. Conclusion

The project successfully developed an LSTM-based classification model for out-of-scope intent detection. The LSTM architecture demonstrated strong performance by effectively capturing sequential patterns in the input data. Through careful data preprocessing and

evaluation, the model exhibited good generalization capabilities and showed promise for practical applications in domains such as customer support chatbots and virtual assistants. However, further improvements can be made to enhance the model's robustness, particularly in handling rare or ambiguous out-of-scope intents.

6. Future Scope of Work

The future scope of out-of-scope intent classification using LSTM includes several potential areas of improvement and expansion. Here are a few possibilities:

- Dataset augmentation: Increasing the diversity and quantity of the training data can enhance the model's ability to handle a wider range of out-of-scope intents. Techniques such as data augmentation, data synthesis, or incorporating external datasets can be explored to improve model performance.

- Transfer learning: Leveraging pre-trained language models, such as BERT or GPT, as a starting point for training the LSTM model can potentially enhance its performance. Transfer learning allows the model to benefit from the knowledge learned from large-scale language modeling tasks and can be particularly beneficial when labeled out-of-scope intent data is limited.

- Ensemble models: Combining the predictions of multiple LSTM models or combining LSTM with other classifiers, such as random forests or support vector machines, can lead to improved accuracy and robustness. Ensemble methods can help mitigate biases or limitations of individual models.

- Handling rare and ambiguous intents: Addressing the challenge of rare or ambiguous out-of-scope intents is an important area for future research. Techniques such as active learning, where the model selectively seeks labeled examples for uncertain intents, or incorporating user feedback in real-time can help improve the model's ability to handle these cases effectively.

- Online learning: Implementing online learning techniques can allow the model to continuously adapt and update itself as new data becomes available. This is particularly useful in scenarios where the distribution of out-of-scope intents may change over time.

- Deployment considerations: Considerations such as model size, inference speed, and resource efficiency should be taken into account when deploying the LSTM model in real-world applications. Optimizing the model architecture or implementing techniques like model compression can help ensure efficient and practical deployment.

7. References

- BERT for Joint Intent Classification and Slot Filling. 1

- Deep Bi-Directional LSTM Network for Query Intent Detection. 2

- Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot 3