Technical Report

Aakash Raj Dhakal (nvc22)
Bishal Thapa (b_t220)
Department of Computer Science
CS.7389 - Spring 2024

Abstract

An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction is a challenge hosted by the UC Irvine Machine Learning Repository mis (2020). It is a synthetic text dataset and is licensed under Creative Commons Attribution 4.0 International. The goal of the problem is to evaluate various classifiers on out-of-domain performance. The project aims to solve this classification problem using Transformers and LSTM(Long Short Term Memory). We first tried to classify using LSTM models and later tried the **BERT(Bidirectional Encoder Representations from Transformers)** which uses 'attention' to prioritize relevant input features in sentences to create a better result. BERT is based on transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based on their connection.

1 Introduction

CLINC150 dataset is one of the UCI Machine Learning Repository datasets. This dataset is designed for task-oriented dialog systems where the user query call fall outside the scope of supported intents.

2 Problem Description

2.1 Data

The data provided for the competition mis (2020) includes 4 versions of the dataset. We worked with **data_full.json** that consists of full version of the data. Each of the 150 in-domain intent classes has 100 train, 20 val, and 30 test samples and the out-of-domain class has 100 train, 100 val, and 1,000 test samples as depicted in Table 1. It is important to note that the out-of-domain class does not necessarily need to be used at training time. A sample of the data is shown in Figure 2.

Column	Dtype
train	15000 samples
val	3000 samples
test	4500 samples
oos_train	100 samples
oos_test	1000 samples
oos_val	100 samples

Table 1: Description of Columns

Domain	Intent	Query
BANKING	TRANSFER	move 100 dollars from my savings to my checking
WORK	PTO REQUEST	let me know how to make a vacation request
META	CHANGE LANGUAGE	switch the language setting over to German
AUTO & COMMUTE	DISTANCE	tell the miles it will take to get to las vegas from san diego
TRAVEL	TRAVEL SUGGESTION	what sites are there to see when in evans
HOME	TODO LIST UPDATE	nuke all items on my todo list
UTILITY	TEXT	send a text to mom saying i'm on my way
KITCHEN & DINING	FOOD EXPIRATION	is rice ok after 3 days in the refrigerator
SMALL TALK	TELL JOKE	can you tell me a joke about politicians
CREDIT CARDS	REWARDS BALANCE	how high are the rewards on my discover card
OUT-OF-SCOPE	OUT-OF-SCOPE	how are my sports teams doing
OUT-OF-SCOPE	OUT-OF-SCOPE	create a contact labeled mom
OUT-OF-SCOPE	OUT-OF-SCOPE	what's the extended zipcode for my address

Table 2: Sample Queries from dataset (Larson et al. (2019))

2.2 Objective

This text classification problem requires the classification of the intention with 150 in-domain intent classes. The main purpose of the dataset is to evaluate various classifiers on out-of-domain performance. The primary goal of our research was to evaluate how classifiers perform on out-of-domain (OOD) intents that the model has not seen during training. This tests the model's ability to generalize to new, unseen types of queries, which is a critical aspect of building robust AI systems.

3 Methodology

3.1 Data Exploration

Figure 1 shows the length distribution of the texts in the Inscope Texts in the training set. And figure 2 shows the heatmap that we obtained using Seaborn, which shows the frequency information about the data. Figure 3 and Figure 4 depicts the word term distribution as a word cloud. Figure 5 shows the top 10 words in out-of-scope dataset in terms of frequency, and the corresponding frequency in the in-domain intent dataset.

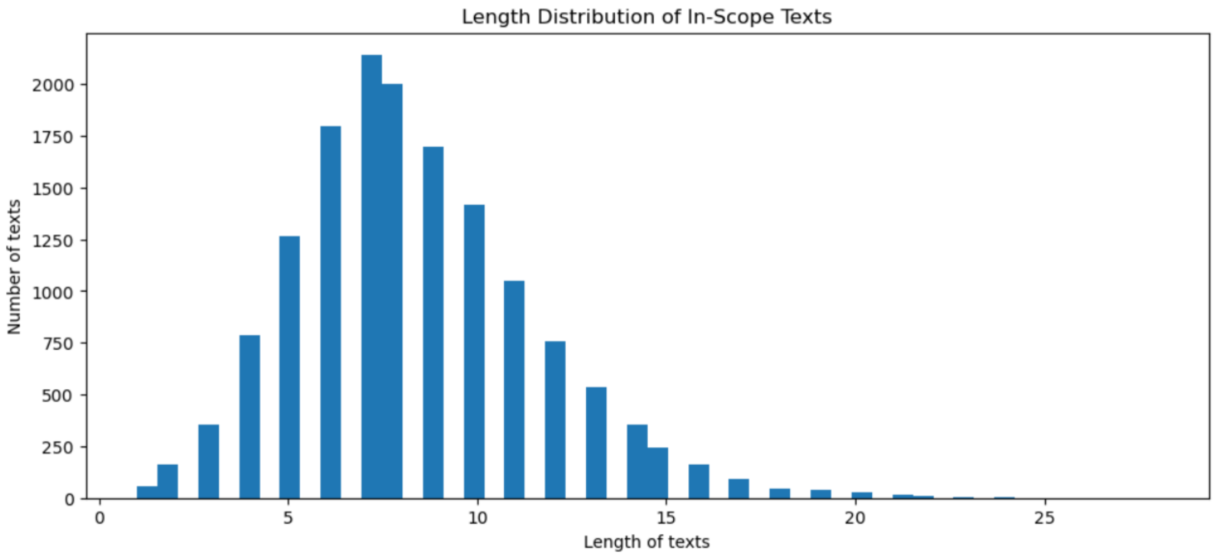


Figure 1: Length Distribution of Inscope Texts

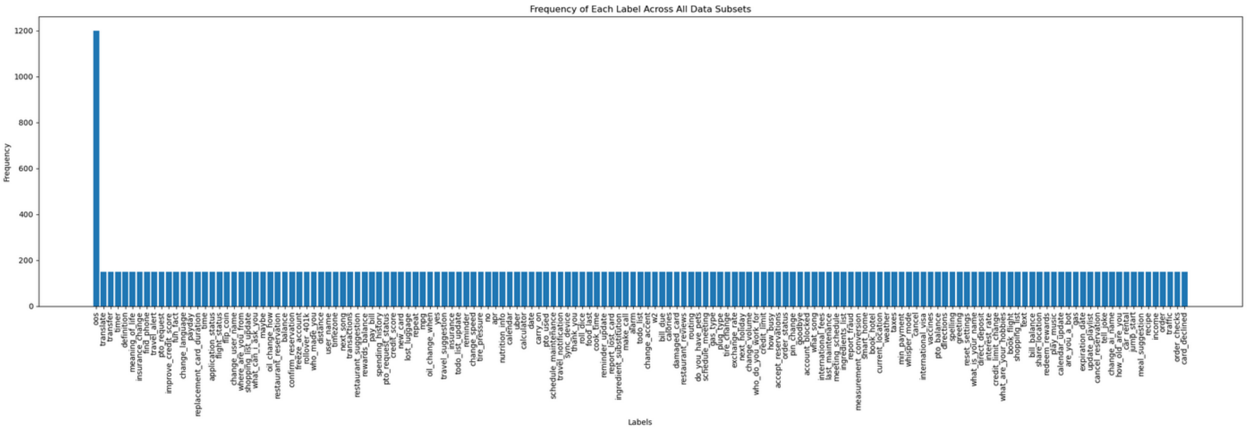


Figure 2: Heatmap of Text Frequency

3.2 LSTM Data Preparation

3.2.1 Tokenization

The textual data has intentions as labels. We converted the text into a sequence of integers. Each unique word is assigned a unique integer. LSTM or Transformers models don't understand textual data but can process numbers. So, this step converts the textual data into a numerical format that the models can interpret.

3.2.2 Sequence Padding

The integer values that we generated from tokenization were of different lengths. So, we plan to keep it to the same lengths using padding. LSTMs or Transformers, like most neural networks, require input dimensions to be consistent. Padding standardizes the length of the sequences, allowing for batch processing.

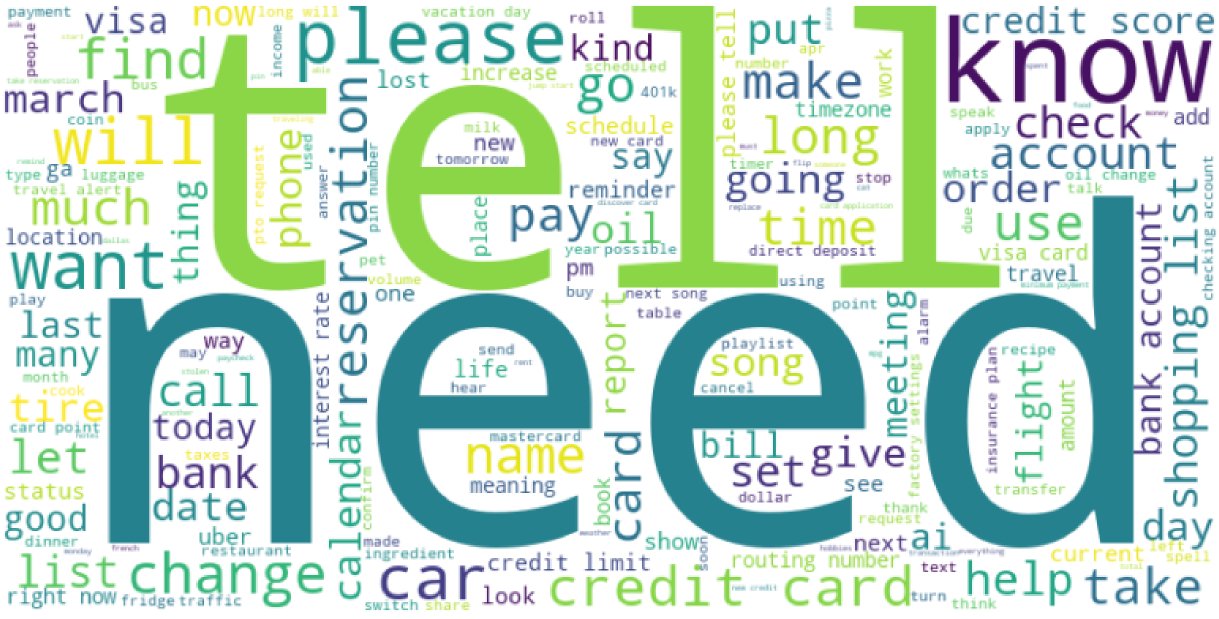


Figure 3: Word cloud for in-scope training set

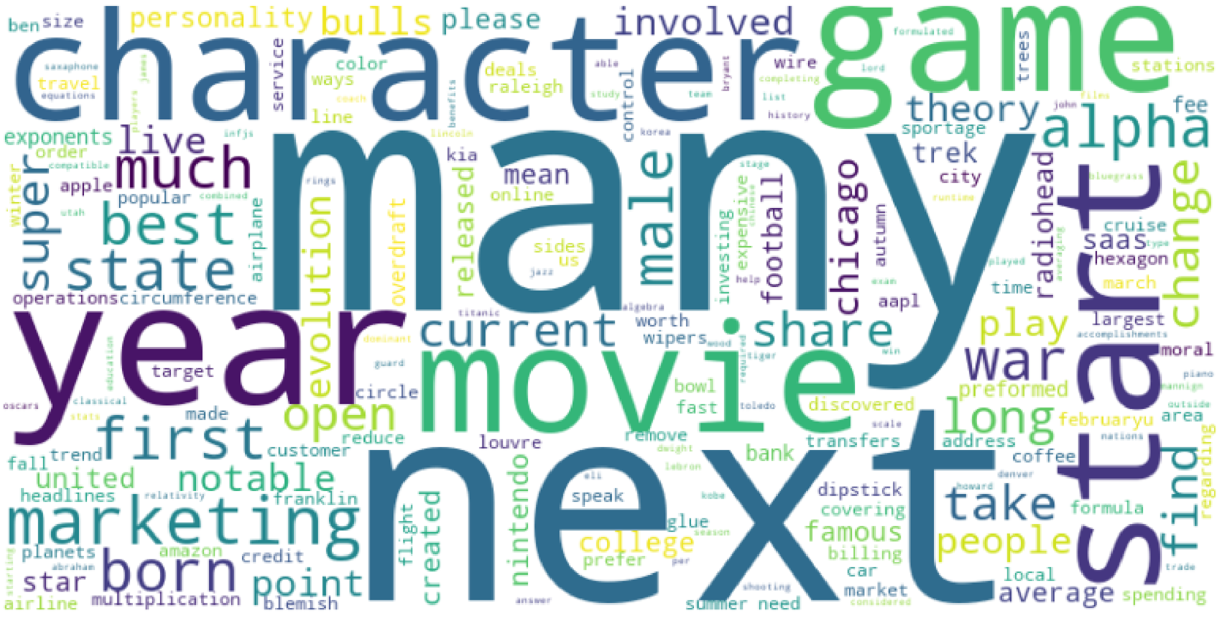


Figure 4: Word cloud for out-of-scope training set

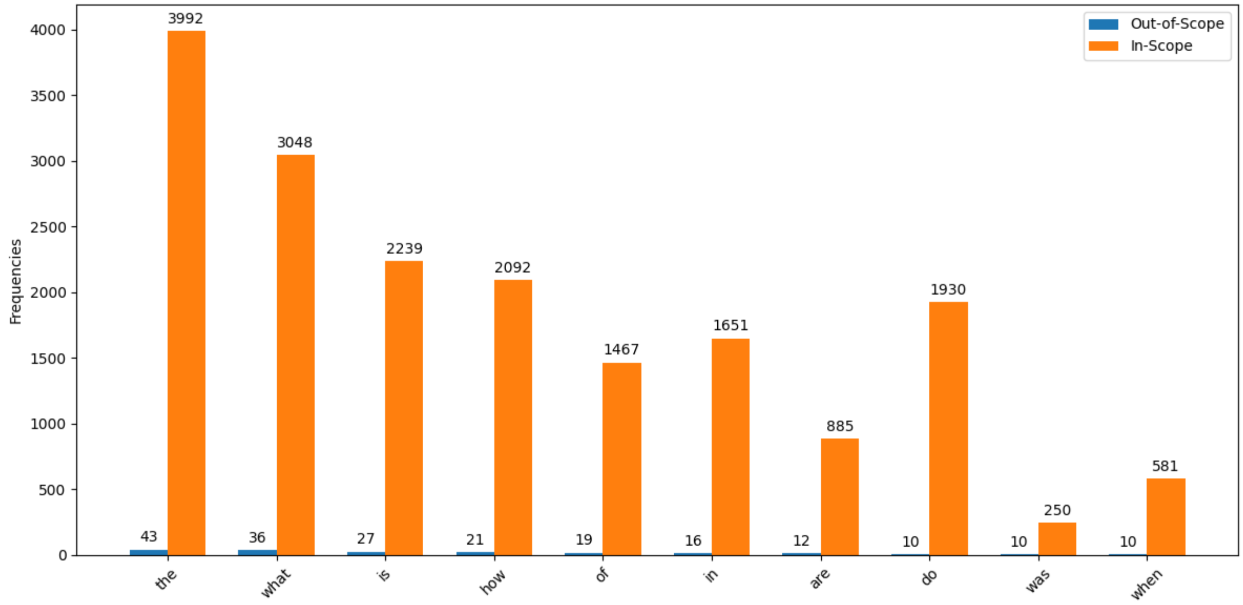


Figure 5: Frequency count for top words in out-of-scope set and corresponding frequency in-scope-set

3.2.3 Word Embedding

After the data is converted into a fixed size, the embedding operation is performed. Embedding transforms integer tokens into fixed-size dense vectors. Embeddings capture semantic meanings and relationships between the words. Dense vectors are more expressive and efficient for the network to process than sparse one-hot vectors.

3.2.4 Label Encoding

The labels that we have are also in text format. We also convert it to numerical format since it is the only way to compare and perform operations in Machine Learning. Label Encoding converts categorical labels into a numerical format. Neural networks work with numerical data, so categorical class labels need to be transformed into a numerical format, typically one-hot encoded vectors for classification tasks.

3.2.5 Data Shuffling and Splitting

In data shuffling, we randomize the order of data points and divide the dataset into training, validation, and test sets. Instead of keeping the data in the same format as it is, this technique involves randomly shuffling to generate a random order of the data. Shuffling prevents the model from learning any unintended sequential patterns in the data. Splitting helps in training the model (training set), tuning model parameters (validation set), and finally assessing the model's performance (test set).

3.2.6 Inclusion of Out-of-Scope Samples

The inclusion of out-of-scope samples was done to teach the model to recognize inputs that do not belong to any predefined intent classes. The purpose is to enhance the model's robustness by enabling it to correctly identify and handle queries that were outside the defined scope.

3.3 Transformer Data preparation steps

3.3.1 Tokenizer Preparation

We have tokenizer preparation at first to utilize BERT's tokenizer to convert text into tokens that align with the tokenization used during BERT's pre-training. It is done because it ensures compatibility with BERT's pre-trained model, which requires specific formats like special tokens ([CLS] for the start of the sequence, [SEP] for separation and end).

3.3.2 Attention Mask Generation

After tokenization, we create masks to indicate to the model which tokens should be attended to and which are padding. BERT uses attention mechanisms that need to differentiate the real data from padding, ensuring that the model's focus is maintained on meaningful data.

3.3.3 Sequence Padding

Sequence padding standardizes the lengths of input sequences to the maximum length that BERT can handle (typically 512 tokens). Similar to LSTM, BERT requires consistent input shapes and sizes for processing. Padding is essential for handling variable-length texts in batch processing.

3.3.4 Label Encoding

Label Encoding converts class labels into integers. It employs an ensemble learning technique. BERT's output layer uses these integers for calculating loss and making predictions, making it crucial for training and evaluation.

3.3.5 Data Shuffling and Splitting

Same as in LSTM, Data Shuffling and Splitting is done to prevent sequence learning biases and effectively evaluate model generalization. It is critical for validating the model's effectiveness across unseen data and for optimizing model parameters without overfitting.

3.3.6 Inclusion of Out-of-Scope Samples

Same as in LSTM, inclusion of out-of-scope-samples is done Same as in LSTM, to classify queries that don't fit any trained categories. It improves the model's ability to function reliably in real-world scenarios by recognizing and appropriately handling out-of-scope queries.

In both LSTM and Transformer models, these steps are designed to maximize the model's performance by ensuring the data is in a suitable format for learning, improving model stability, and enhancing its ability to generalize from training data to real-world applications.

4 Model Development

4.1 LSTM

LSTM, or Long Short-Term Memory, is a type of recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTMs have feedback connections that make them capable of processing entire sequences of data (e.g., a sentence or a time series). This makes LSTMs ideal for tasks such as speech recognition, time-series forecasting, and natural language processing (NLP).

LSTMs are specifically designed to avoid the long-term dependency problem, allowing them to remember information for long periods of time. This is achieved through their unique structure, which includes three gates: an input gate, a forget gate, and an output gate. These gates collectively decide what information should be passed along to the next time step, what should be remembered or forgotten, and what should be output.

4.1.1 Usage of LSTM in the Given Context

In our case, the LSTM model is used for intent classification and out-of-scope prediction in conversational data. Here's how the LSTM was specifically applied:

- The data was first tokenized into sequences, which were then padded to ensure they had a consistent length for batch processing.
- The LSTM network was configured with an embedding layer, which is critical for NLP tasks as it converts token indices into dense vectors of fixed size. This helps in capturing contextual relationships between words.
- The model was trained on a dataset that included both in-scope intent samples and out-of-scope samples labeled as 'oos'. This training helps the LSTM learn to distinguish between different intents and to identify when an input does not belong to any of the trained categories.

4.1.2 Benefits of Using LSTM in our research

- Due to its recurrent nature, LSTM is capable of understanding the context in a sequence of words, helps in determining the intent behind a user's message.
- LSTM's ability to process sequences makes it ideal for conversational AI where the input is typically in the form of sentences or phrases.
- LSTM can remember important details from earlier in the sequence, which helps in maintaining a continuous understanding of the conversation, beneficial for tasks that require context from previous interactions.
- The use of padding and LSTM's inherent structure allow it to handle inputs of varying lengths, which is common in real-world conversational data.

By integrating out-of-scope samples during the training phase, the LSTM model not only learns to classify various intents accurately but also effectively identifies queries that are not covered by the training data. This dual capability is particularly beneficial in practical applications where unexpected user inputs are common.

4.2 Transformers

The Transformer architecture, introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017, represents a significant departure from earlier sequence processing models like RNNs and LSTMs. Unlike these older models, which process sequences step-by-step, Transformers use a mechanism called self-attention to process all parts of the input data simultaneously. This parallel processing capability allows Transformers to achieve remarkable efficiency and effectiveness in tasks involving large datasets and complex dependencies.

Transformers consist of two main parts: an encoder that processes the input data and a decoder that produces the output. Each of these parts is made up of multiple layers of self-attention and position-wise feedforward neural networks. For NLP tasks, often only the encoder part is used, as in models like BERT (Bidirectional Encoder Representations from Transformers).

4.2.1 Usage of Transformer (BERT) in the Given Context

In our context, a Transformer model, specifically BERT, was used for intent classification and out-of-scope prediction. Here's how BERT was applied:

- BERT has been pre-trained on a large corpus of text in a self-supervised manner, learning a deep representation of language. This pre-training is leveraged by fine-tuning BERT on the specific task of intent classification with additional out-of-scope labels.
- BERT requires a specific format for its inputs, which includes tokenization using a WordPiece model, addition of special tokens ([CLS] at the beginning and [SEP] at the end), and padding to a fixed sequence length.
- Utilizing attention, BERT can weigh the importance of different words relative to others in a sentence, which helps in better understanding the context and nuances of language.
- After loading the pre-trained BERT model, it's fine-tuned on the labeled data containing both in-scope intents and out-of-scope examples. This fine-tuning adjusts the model weights slightly to adapt to the specifics of the task.

4.2.2 Benefits of Using Transformer in our research

- Unlike LSTMs, Transformers can manage long-range dependencies in text more effectively, allowing them to understand more complex and longer sentences without losing performance.
- Due to its bidirectional nature, BERT can integrate context from both left and right sides of a word in a sentence across the whole dataset, providing a richer understanding of language.
- The parallel processing capabilities of the Transformer allow for faster training times, especially on large datasets with modern hardware (GPUs/TPUs).
- Transformer models often achieve state-of-the-art results in many NLP tasks, including intent classification, due to their deep and powerful architecture.

The Transformer model, particularly in its implementation as BERT, offers a powerful tool for NLP tasks due to its deep contextual understanding and efficient training characteristics. For tasks like intent classification and out-of-scope prediction, BERT provides robust performance and scalability, making it a popular choice for developers and researchers in the field of artificial intelligence.

5 Results

5.1 Evaluation Measures

5.1.1 Accuracy

It denotes the ratio of correctly predicted observations (both true positives and true negatives) to the total observations. In evaluating the predictive models, several performance metrics were employed to assess their efficacy in classification tasks. The following key metrics were utilized: Precision, Recall, F1 Score, and Accuracy. These metrics provide a comprehensive understanding of the models' abilities to classify positive and negative instances, considering both correct and incorrect classifications.

5.1.2 Precision

It measures the ratio of correctly predicted positive observations to the total predicted positives. Higher precision values indicate fewer false positives. Notably, the models exhibited high precision across the board, ranging from approximately 82% to 89%. This signifies the models' proficiency in accurately identifying positive cases while minimizing false positive predictions.

5.1.3 Recall

Recall, also known as sensitivity, quantifies the ratio of correctly predicted positive observations to the actual positives in the dataset.

5.1.4 F1 Score

It a harmonic mean of precision and recall, provides a balanced assessment of a model's accuracy in classification tasks. Similar to precision and recall, the F1 Scores were notably high, ranging between 98.9% and 99.6%. These scores indicate a balanced performance in terms of both precision and recall.

5.1.5 Area Under the ROC (AUC-ROC)

is a common metric used to assess the performance of binary classification models. It measures the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity). The range of ROC value from the values was 0.87 to 0.93.

5.2 Model Performance

We ran two different types of models for sample classification, LSTM and Transformer. We further worked on two different versions of the LSTM model.

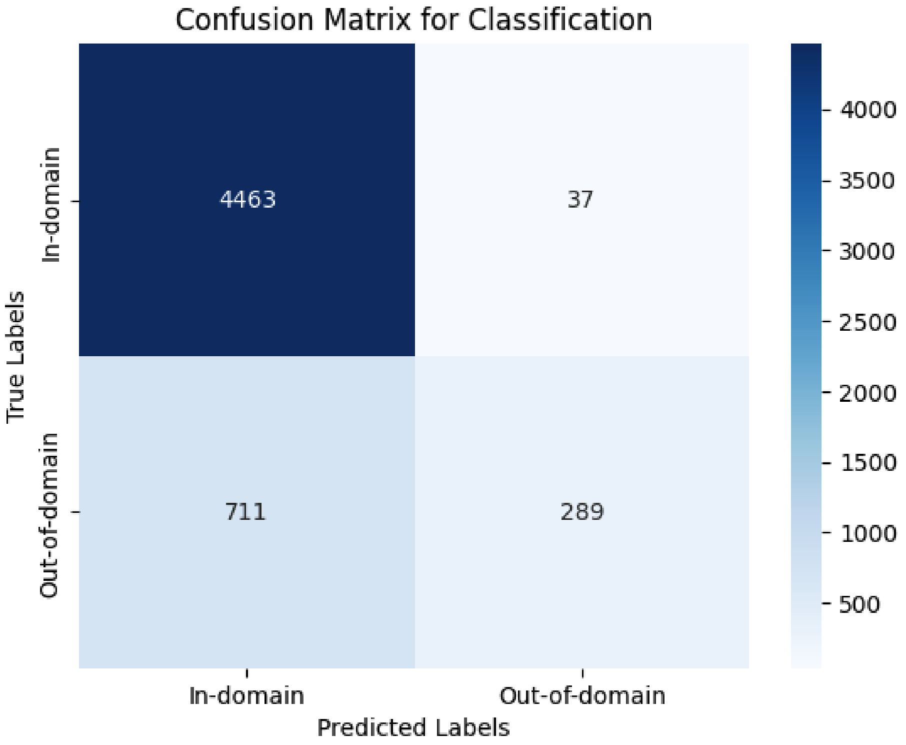
5.2.1 LSTM

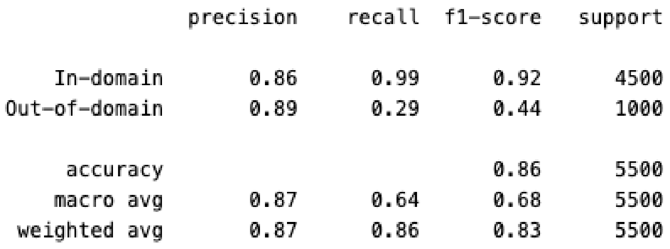
The first LSTM model was trained with in-domain-class training samples, as well as out-of-domain training samples. The validation accuracy for this model was around 85.9%. Since there were 150 in-domain-classes, we grouped all the in-domain-classes before generating the confusion matrix for simplicity in visualization. The confusion matrix for the classification is depicted in 6, and we can see that 2970 out of 3000 in-domain samples are correctly classified, but most of the out-of-domain samples are misclassified.

5.2.2 LSTM without training the Out-of-domain samples

The second LSTM model was trained only with the in-domain-class training samples. So, the model was not introduced to the out-of-domain samples in the training phase. The validation accuracy for this model was around 88.00%. We introduced the out-of-domain test samples in the testing phase only. So, in order to label these out-of-domain samples, we created a threshold probability value (threshold), which we manually declared. If the maximum probability output from the model (softmax layer) for all the in-domain-class is less than the threshold, then we classify it as out-of-domain class.

Similar to 1st model, we grouped all the in-domain-classes before generating the confusion matrix for simplicity in visualization. The confusion matrix for the classification is depicted in 8, and we can see that 37 out of 4500 in-domain samples are correctly classified, but 819 out of 1000 samples were misclassified for the out-of-domain samples which makes sense as the model was not trained on out-of-domain samples.





```
               precision    recall  f1-score   support

    In-domain       0.86      0.99      0.92      4500
Out-of-domain       0.89      0.29      0.44      1000

     accuracy                           0.86      5500
    macro avg       0.87      0.64      0.68      5500
 weighted avg       0.87      0.86      0.83      5500
```

Figure 6: Confusion Matrix for LSTM trained with out-of-domain samples

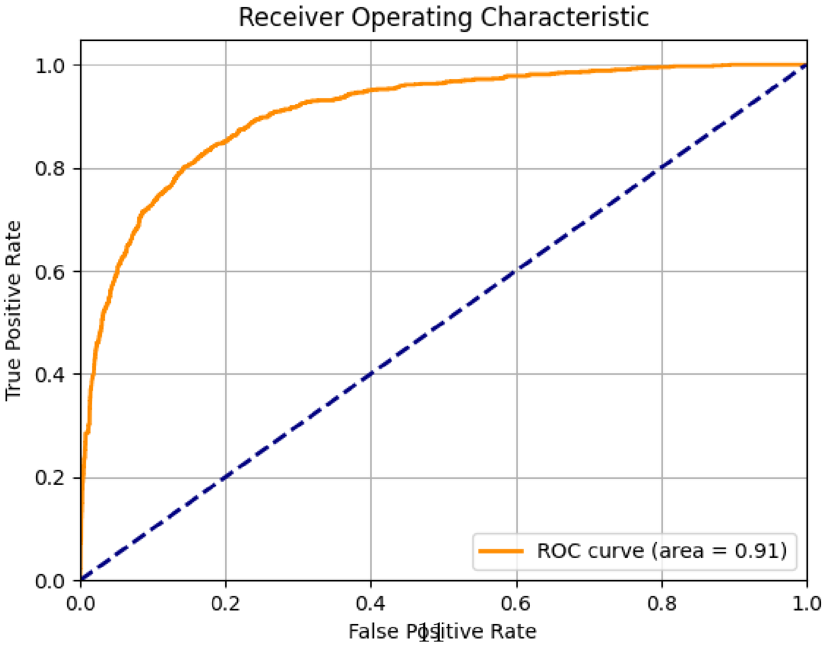
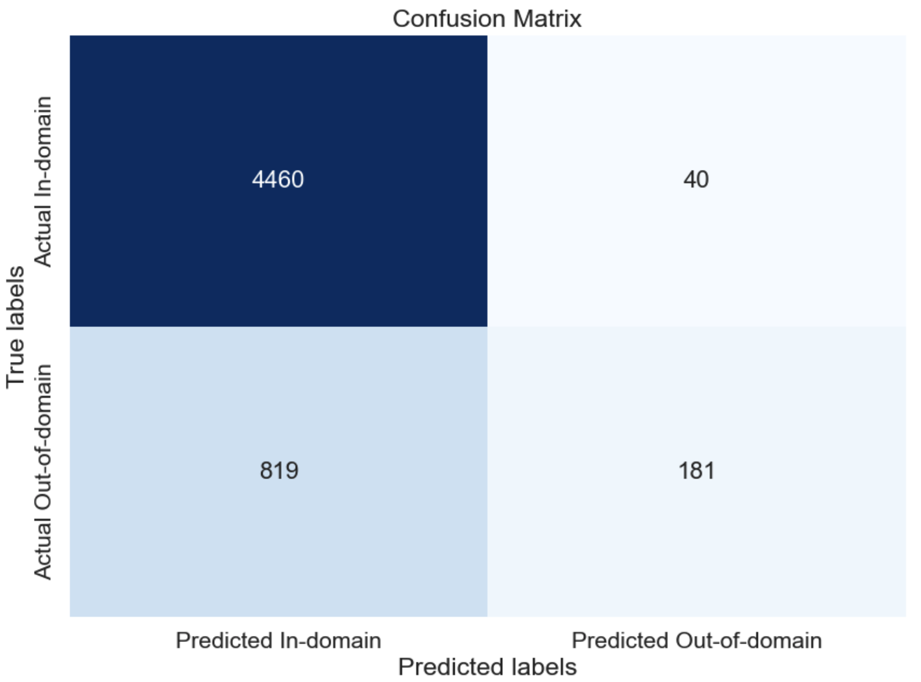
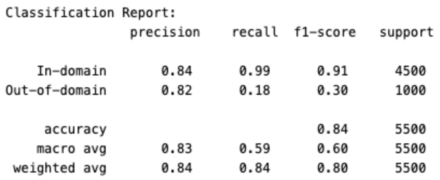


Figure 7: ROC AUC for LSTM trained with out-of-domain samples





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

    In-domain       0.84      0.99      0.91      4500
Out-of-domain       0.82      0.18      0.30      1000

     accuracy                           0.84      5500
    macro avg       0.83      0.59      0.60      5500
 weighted avg       0.84      0.84      0.80      5500
```

Figure 8: Confusion Matrix for LSTM trained without out-of-domain samples

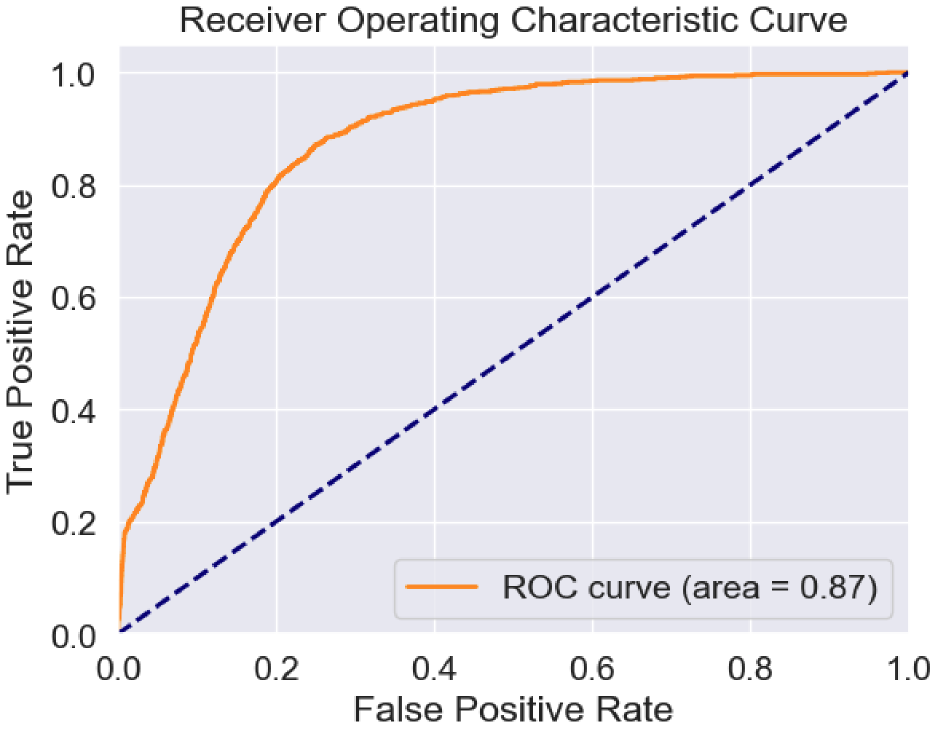


Figure 9: ROC AUC for LSTM trained without out-of-domain samples

5.2.3 Transformers

We used the BERT neural network as our 2nd model, and used self-attention mechanisms to process the context of words. The confusion matrix for the classification is depicted in 10, and we can see that all of the in-domain samples were correctly classified. However, most of the out-of-domain samples were misclassified. The number of epochs was 20, the validation accuracy of the model was 97% and precision was 0.93.

6 Possible Improvements

The number of samples in each of the classes could be increased to improve the performance of the model. This would help the model to better recognize the queries which are out-of-domain. Generally, these models can classify the in-scope intent classes but fail with the out-of-scope queries, so increasing the out-of-domain samples could also help better the model's accuracy in classifying out-of-domain samples.

7 Conclusion

In this research, we explored the capability of LSTM and BERT in the context of the classification of in-domain intent classes and out-of-domain classes. We used two different models of LSTM. The first model was trained with both in-domain class and out-of-domain class, while the second model was trained exclusively on in-domain intent classes. For the 2nd model, we created a threshold value for the classification probability to classify out-of-domain samples. This meant that, if the maximum probability classification any sample was less than the threshold, then it would classify it as an out-of-domain sample. We used the testing dataset to validate the performance of the model based on accuracy, precision, recall, and ROC curve. We evaluated and compared the efficacy of these three AI models to classify queries on various classes.

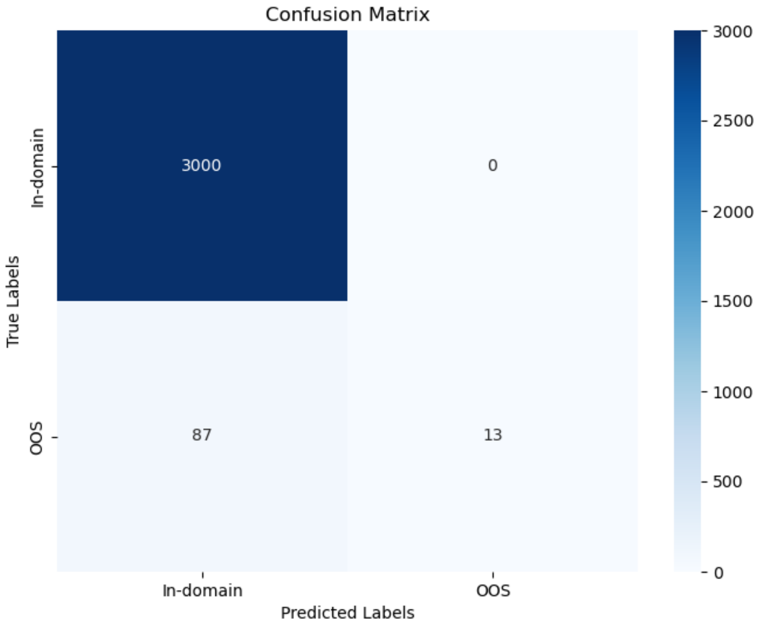


Figure 10: Confusion Matrix for BERT

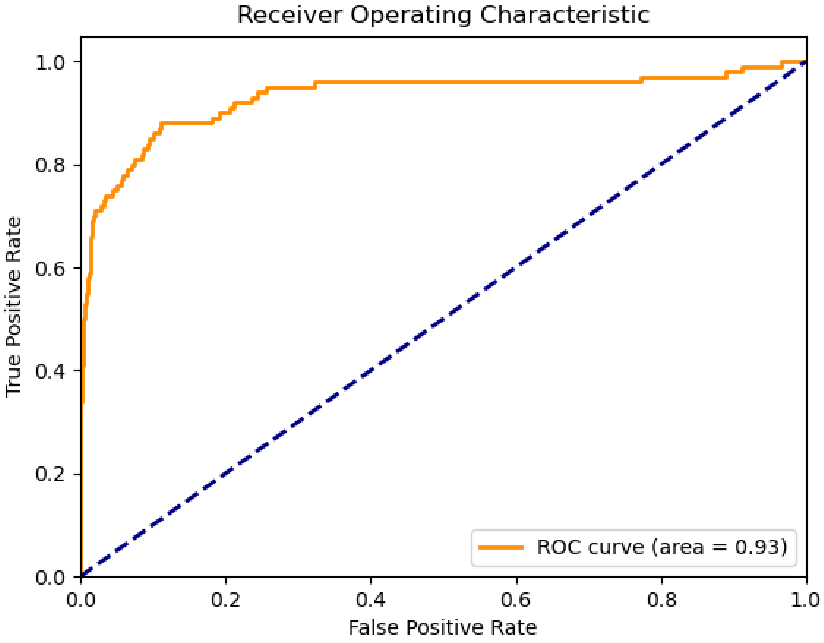


Figure 11: ROC AUC for BERT

References

2020. CLINC150. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5MP58.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. https://www.aclweb.org/anthology/D19-1131