Artificial Neural Network

Hierarchical Multi-task Neural Network for Banking Customer Service

Name: S. Aarabi
Index No: COMScDS242P-010
Module: Artificial Intelligence

MSc Data Science

Objectives

- Develop a hierarchical multi-task neural network to classify banking queries by domain, customer intent, and urgency.

- Deploy a functional web application integrating the model with Explainable AI (XAI) for transparent usage.

- Experiment with a Generative AI model to evaluate advanced performance and compare with the custom neural network.

Methodology

1. Data Collection & Preprocessing
2. Exploratory Data Analysis (EDA)
3. Model Development & Evaluation
4. Model Deployment
5. Explainability Integration
6. GenAI Application

01 Data Preprocessing

- Data Collection:
  - Banking77 dataset collected from Hugging Face domain
- Data Cleaning:
  - Ensured no null values
  - Removed special characters, stop words, and lowercased text
- Feature Engineering:
  - Categorized queries into domains: Card, Account, Transfer, Support
  - Created urgency labels: Critical, Normal, Low
  - Created structured mappings for the model
- Tokenization and padding
  - Converted words into a numerical sequence
  - Applied padding to maintain a fixed input length, prevented issues from variable query lengths

01 Data Preprocessing

- Train, Validation, Test Split:
  - Training set to model learns patterns
  - Validation set to evaluate the model
  - Test set to final unbiased performance evaluation
- Label encoding:
  - Convert text-based class labels into numeric format for model compatibility
  - Card Service: 0, Transfer & Payments: 1, Account Management: 2, General Support: 3
  - Critical:0, Medium:1, Low:2
- Key-word mapping:
  - Certain words mapped to categories and urgency

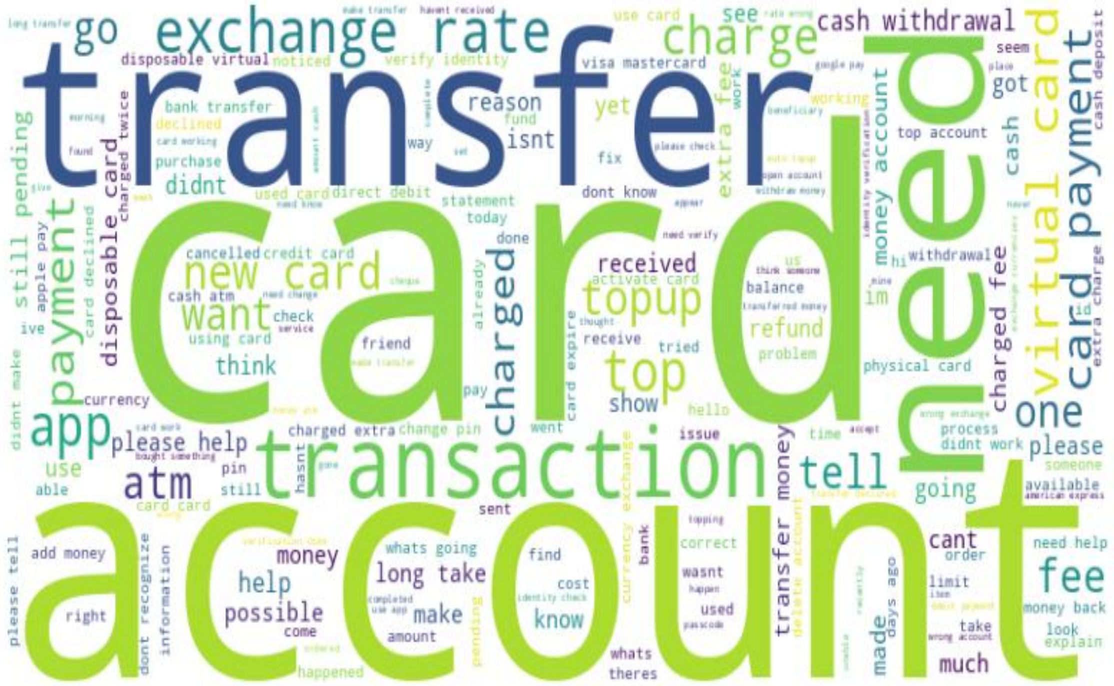
02 Exploratory Data Analysis

- Overview of the dataset

1. Intent Distribution: The Dataset covers 77 distinct intents, well-diversified across categories.

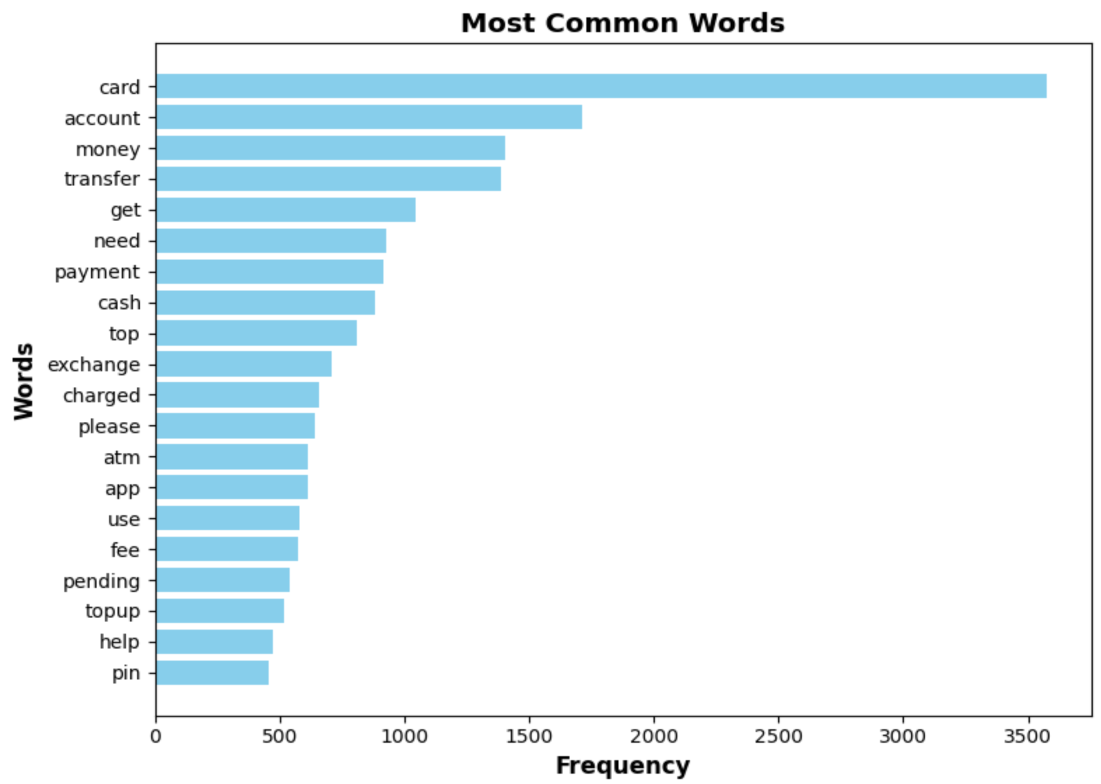
2. Urgency Distribution:
   - Low urgency - 7,425 queries
   - Normal urgency: 3,533 queries
   - Critical urgency: 2,125 queries

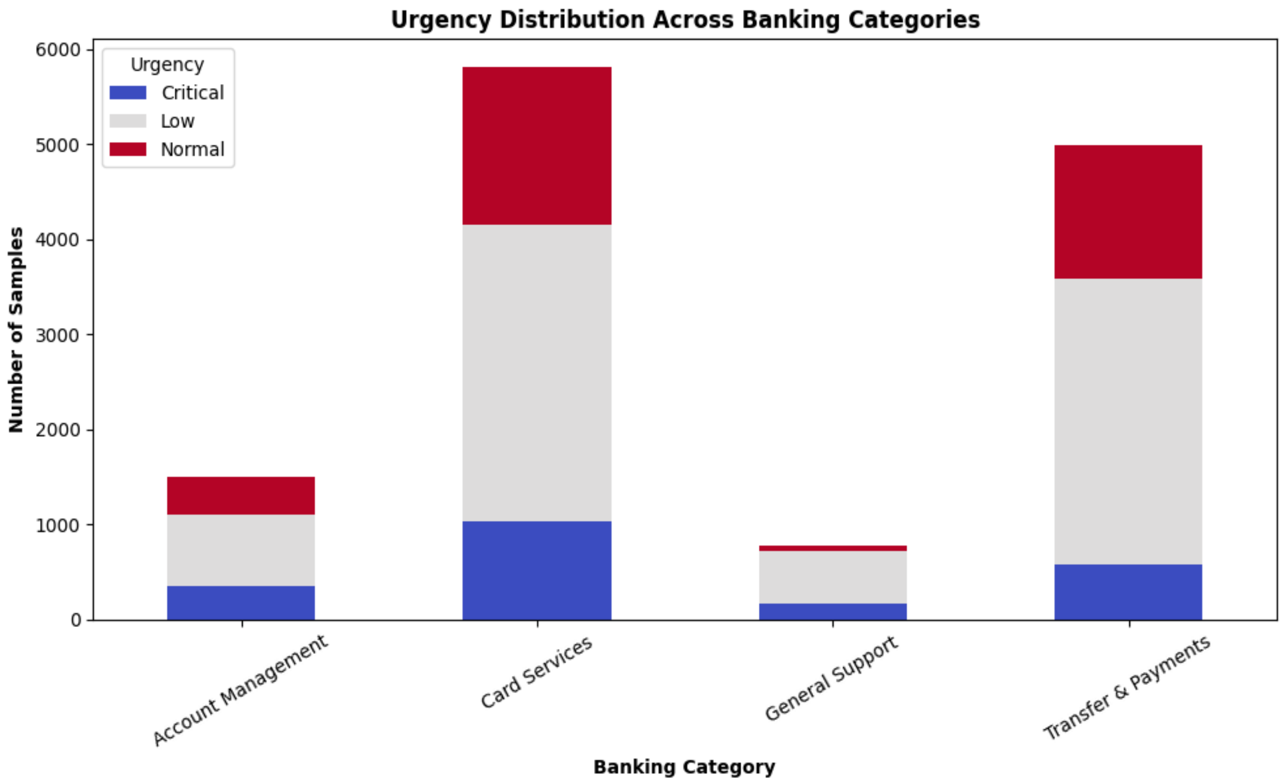
3. Category Distribution:
   - Card Services: 5,816 queries
   - Transfer & Payments: 4,994 queries
   - Other categories less represented

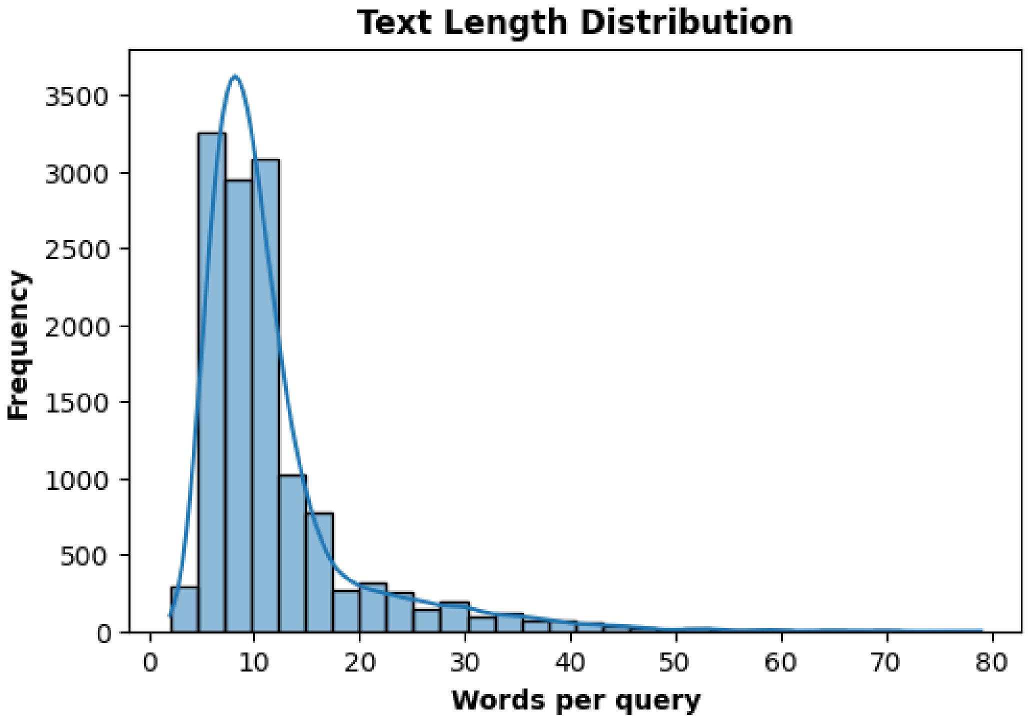


02 Exploratory Data Analysis

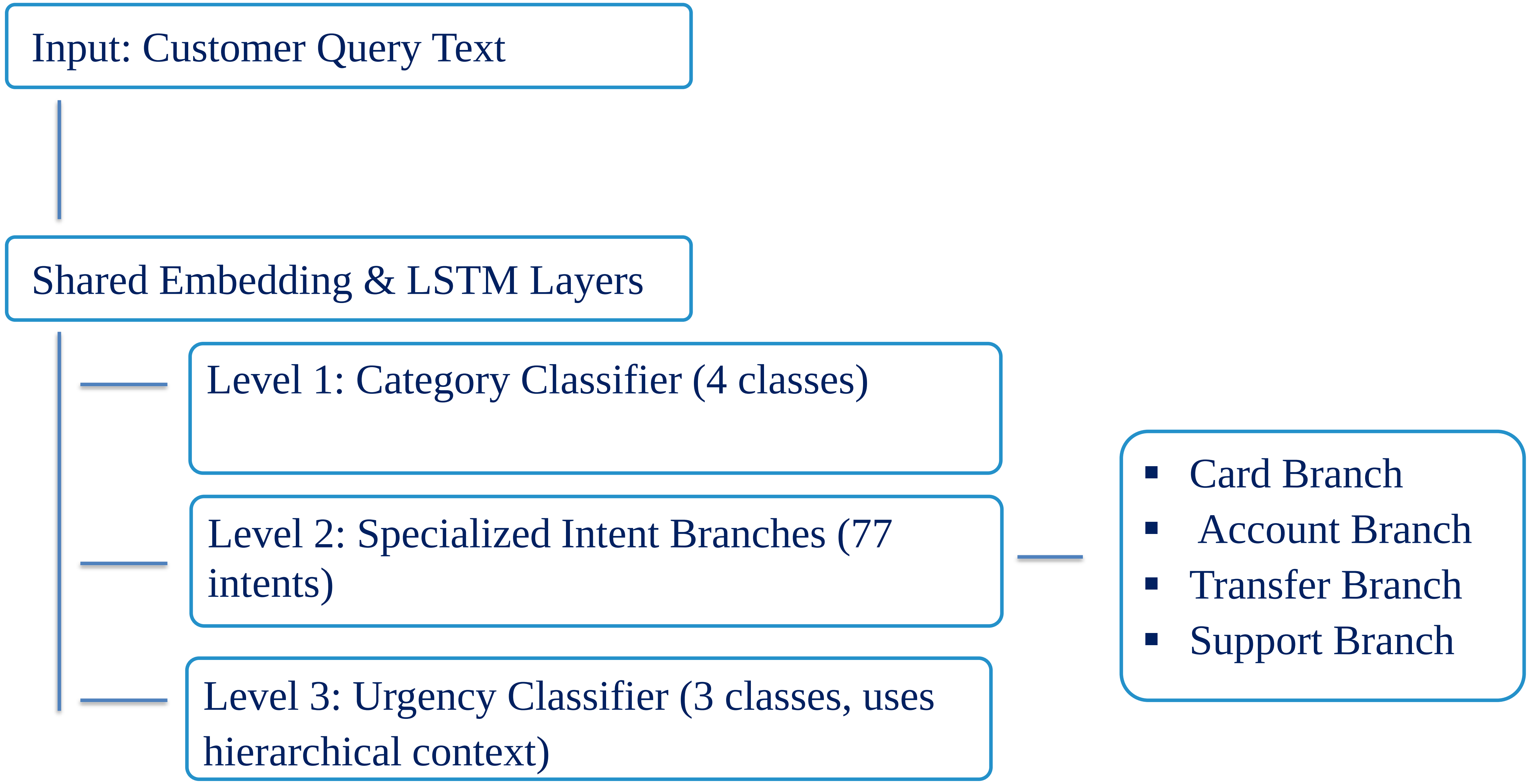
- Frequent word analysis: Identified the most common words across queries, which helps reveal domain-specific vocabulary and recurring customer issues
- Urgency Distribution across Categories: Highlights which services receive more critical or low-priority queries
- Text Length Analysis: Checked the number of words per query


Most Common Words


Urgency Distribution Across Banking Categories


Text Length Distribution

03 Model Development – Model Architecture



Input: Customer Query Text

Shared Embedding & LSTM Layers

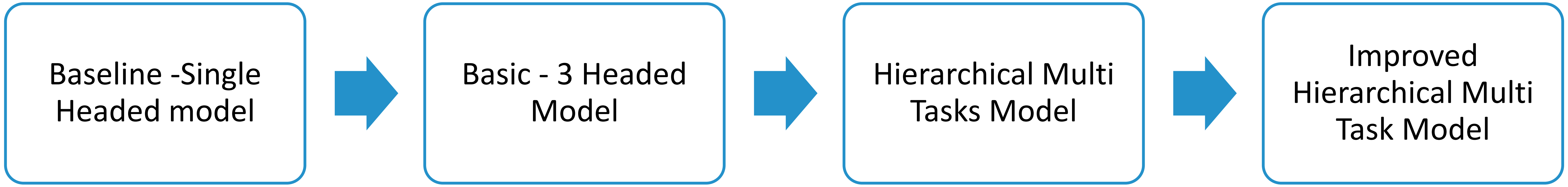
Level 1: Category Classifier (4 classes)

Level 2: Specialized Intent Branches (77 intents)

- Card Branch
- Account Branch
- Transfer Branch
- Support Branch

Level 3: Urgency Classifier (3 classes, uses hierarchical context)

03 Model development – Baseline to Advanced Model Development



Baseline -Single Headed model

Basic - 3 Headed Model

Hierarchical Multi Tasks Model

Improved Hierarchical Multi Task Model

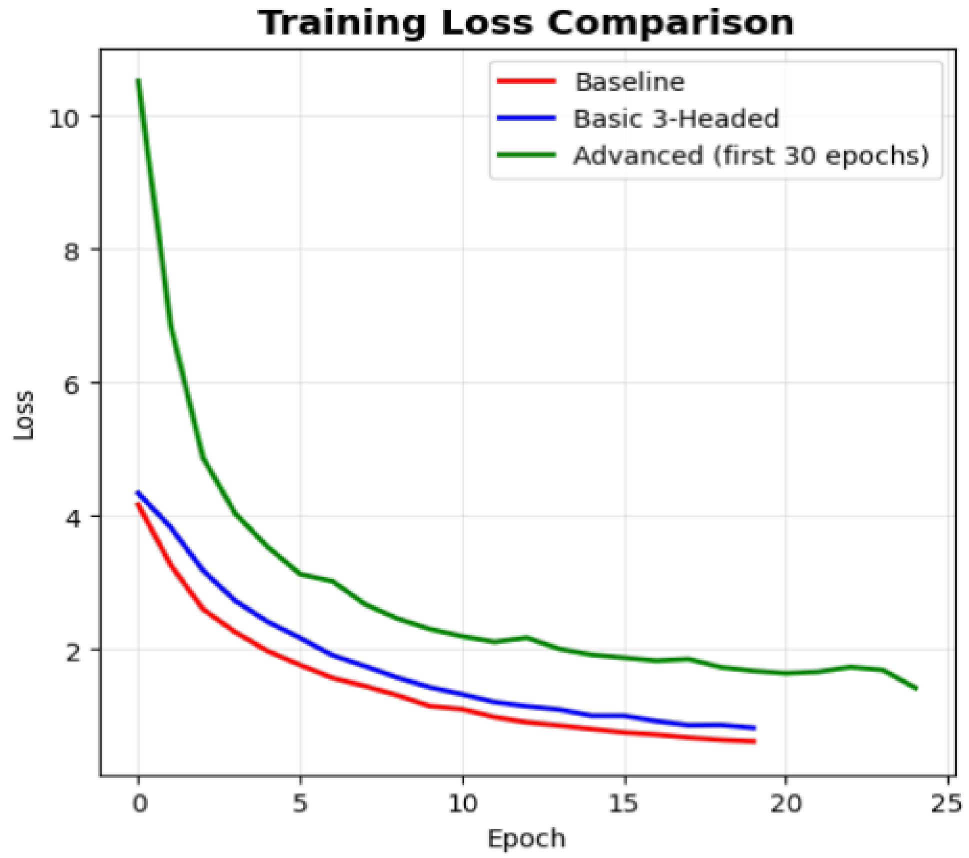
- Baseline Model: Single-head LSTM and intent classification only
- Basic Multi-Task Model: 3 Headed ANN and a shared dense layer with independent heads
- Advanced Hierarchical Model: Shared embeddings with BiLSTM layers and improved contextual prediction with hierarchical flow
- Improved Hierarchical multi-task model: Implemented with optimization Enhancements.

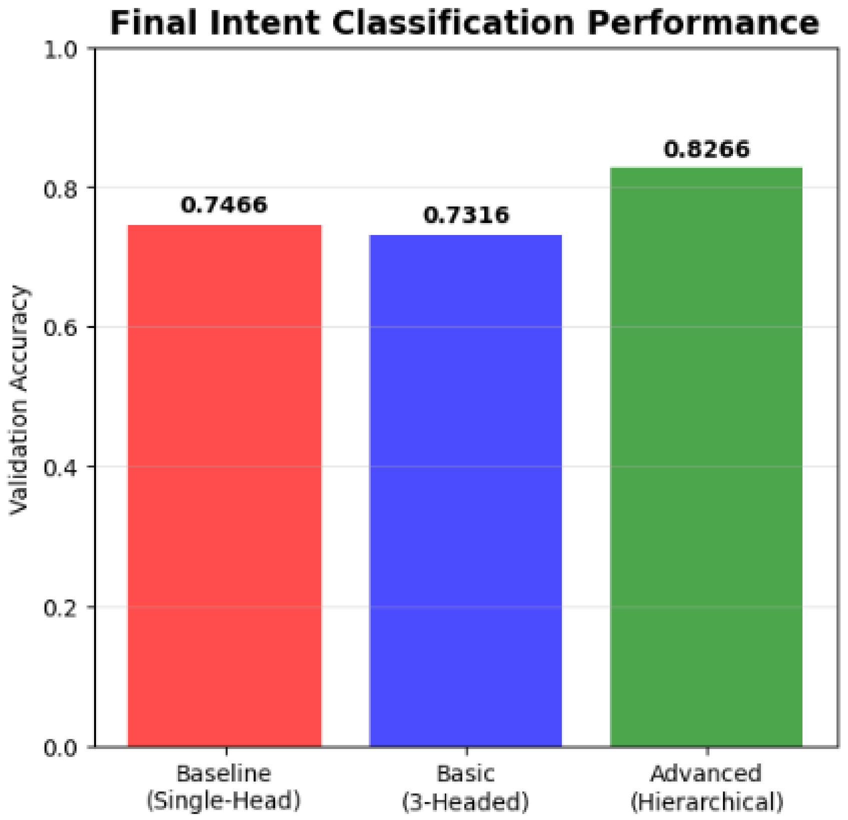
  - Dropout regularization to reduce overfitting

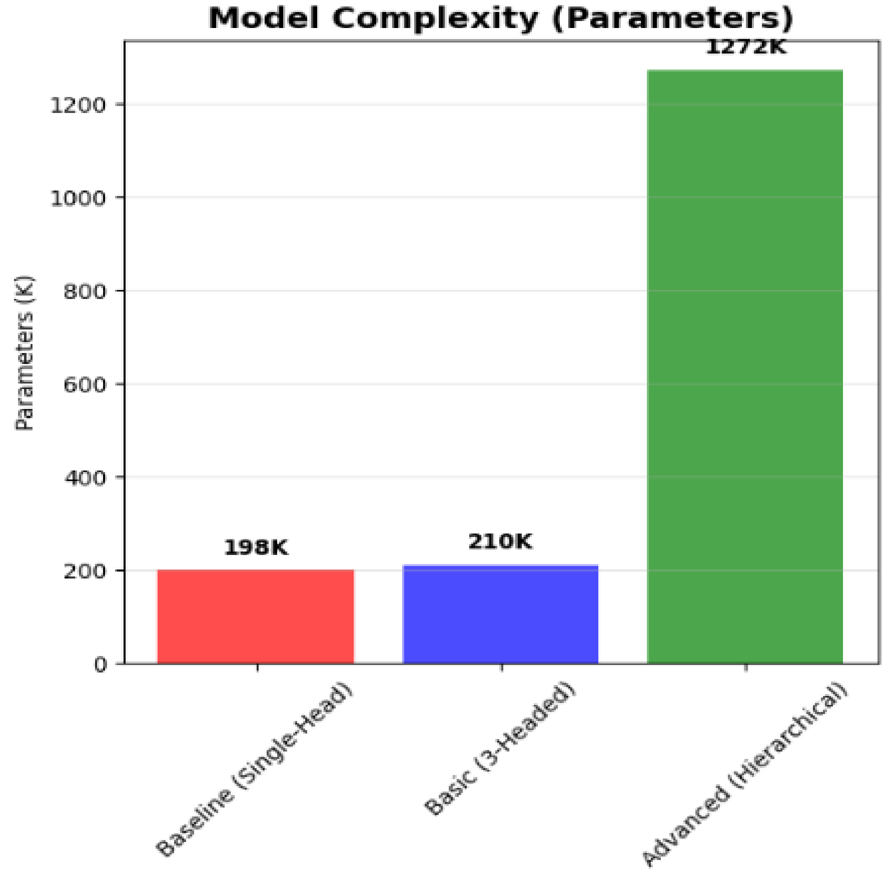
  - Loss weighting to balanced urgency head

03

Model development – Comparison of all 3 models





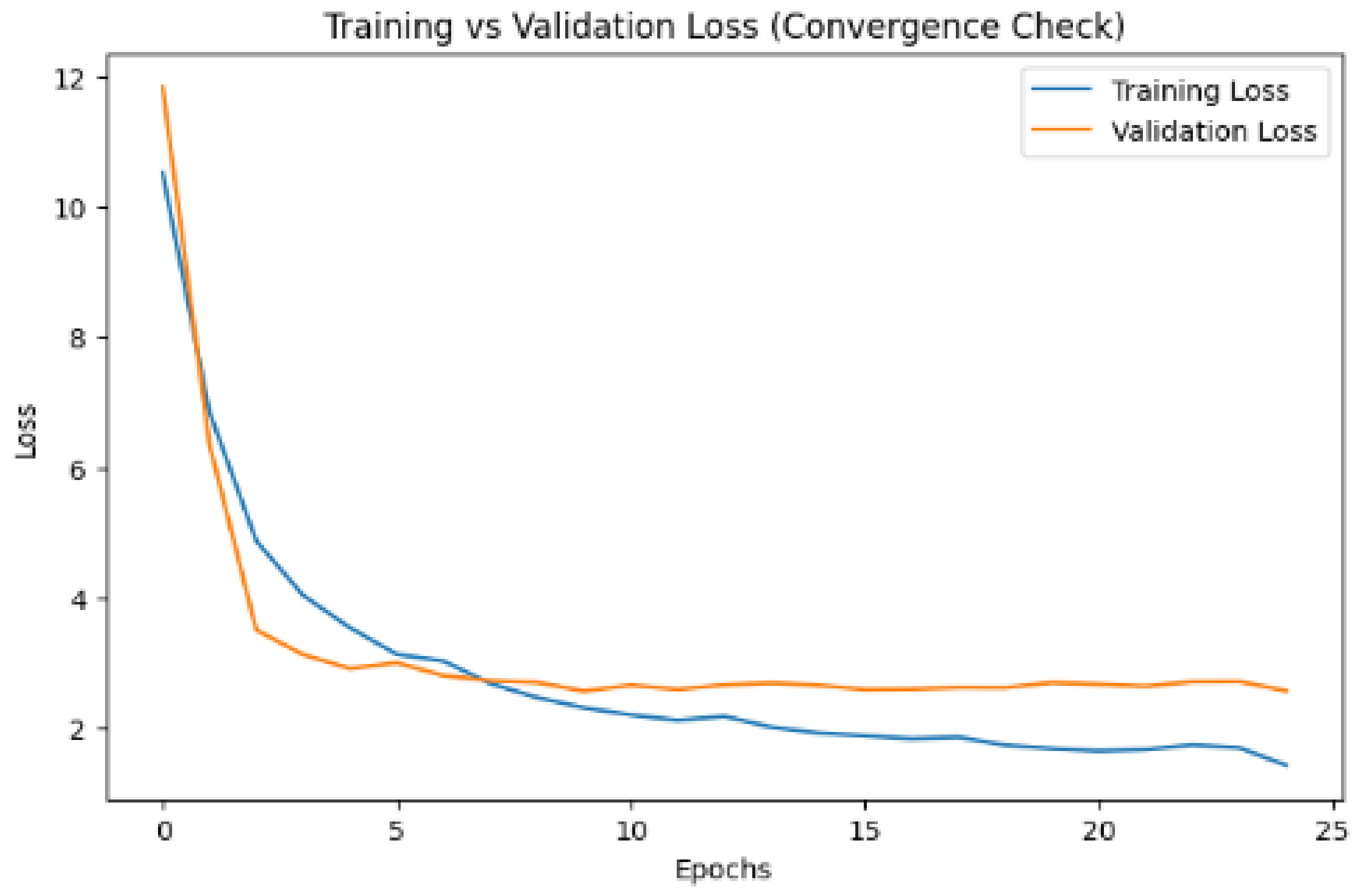


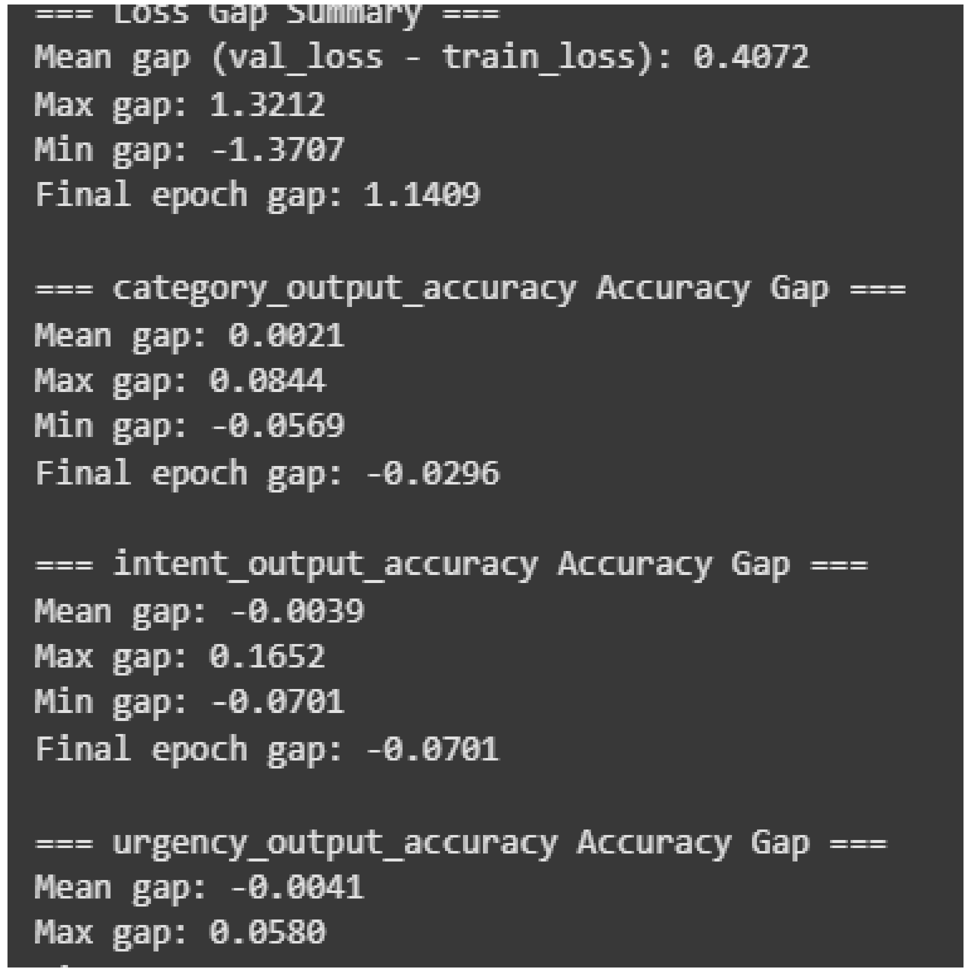
- Training loss comparison shows a steady decline, confirming that each model was able to learn effectively.
- The advanced model shows a higher intent classification accuracy than the other two models, highlighting the stronger learning capability.
- The advanced model with 1.2M parameters achieves substantially higher accuracy, demonstrating that increased complexity, when coupled with a well-structured architecture, yields significant performance gains.

03 Model development - Evaluation of the Advanced Hierarchical Model

For the Advanced Hierarchical Model, a detailed evaluation to assess both its strengths and limitations.

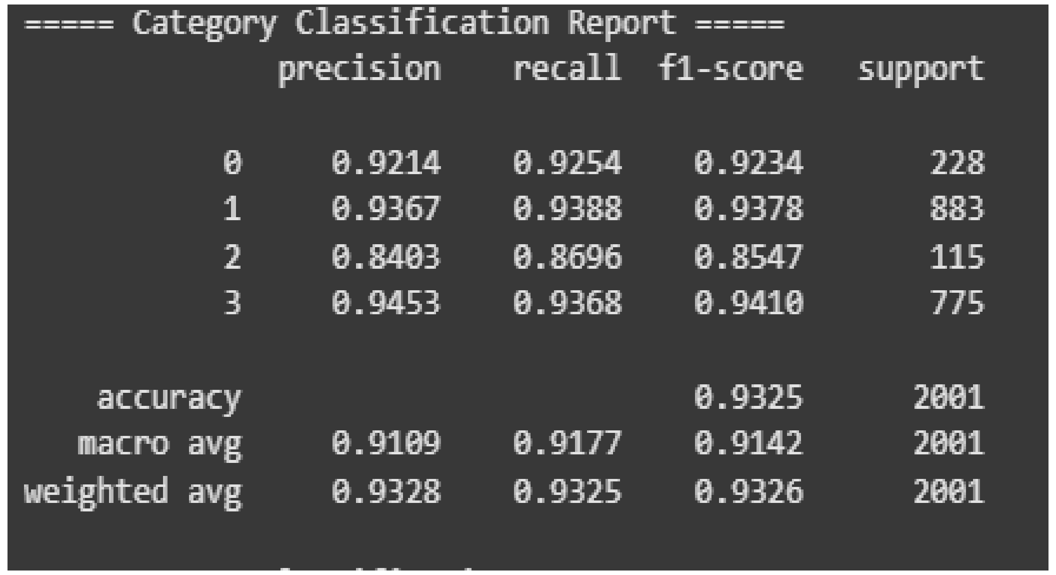
- The loss gap of 0.4072 shows mild overfitting, with the model performing better on training than validation data.
- Task-wise accuracy gap shows that category classification is stable, while intent and urgency show slight overfitting risks**.**



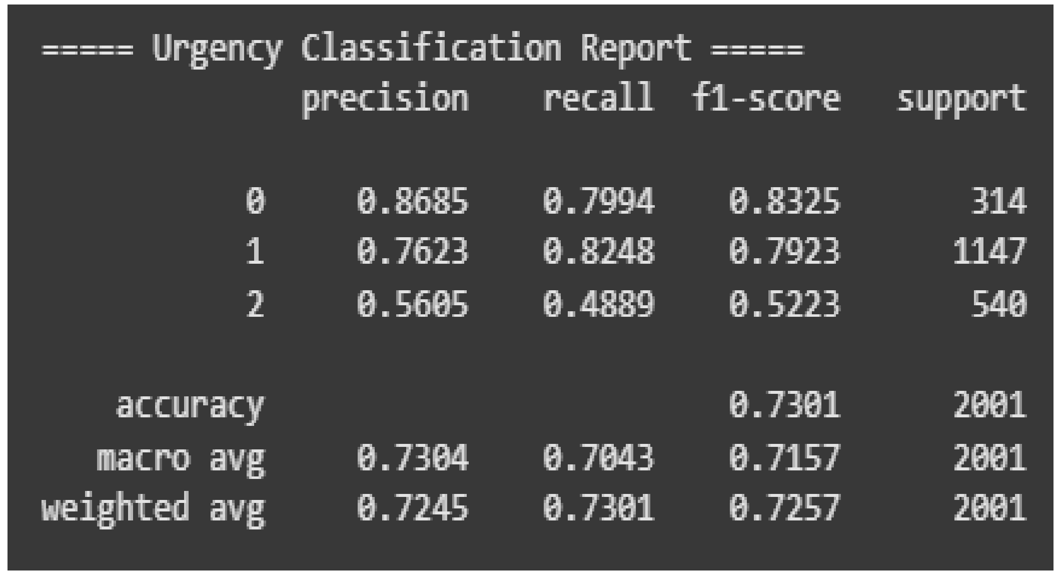


```
=== Loss Gap Summary ===
Mean gap (val_loss - train_loss): 0.4072
Max gap: 1.3212
Min gap: -1.3707
Final epoch gap: 1.1409

=== category_output_accuracy Accuracy Gap ===
Mean gap: 0.0021
Max gap: 0.0844
Min gap: -0.0569
Final epoch gap: -0.0296

=== intent_output_accuracy Accuracy Gap ===
Mean gap: -0.0039
Max gap: 0.1652
Min gap: -0.0701
Final epoch gap: -0.0701

=== urgency_output_accuracy Accuracy Gap ===
Mean gap: -0.0041
Max gap: 0.0580
```

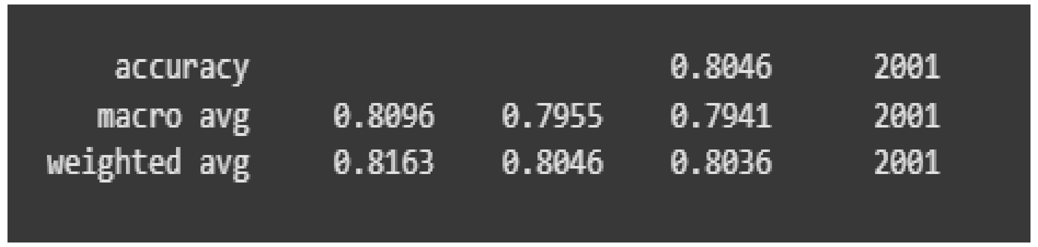
03 Model development - Evaluation of the Advanced Hierarchical Model



```
===== Category Classification Report =====
              precision    recall  f1-score   support

           0     0.9214    0.9254    0.9234       228
           1     0.9367    0.9388    0.9378       883
           2     0.8403    0.8696    0.8547       115
           3     0.9453    0.9368    0.9410       775

    accuracy                         0.9325      2001
   macro avg     0.9109    0.9177    0.9142      2001
weighted avg     0.9328    0.9325    0.9326      2001
```



```
===== Urgency Classification Report =====
              precision    recall  f1-score   support

           0     0.8685    0.7994    0.8325       314
           1     0.7623    0.8248    0.7923      1147
           2     0.5605    0.4889    0.5223       540

    accuracy                         0.7301      2001
   macro avg     0.7304    0.7043    0.7157      2001
weighted avg     0.7245    0.7301    0.7257      2001
```



```
    accuracy                         0.8046      2001
   macro avg     0.8096    0.7955    0.7941      2001
weighted avg     0.8163    0.8046    0.8036      2001
```

- Overall category accuracy of this is 93.25% and precision, recall, and F1 scores are consistently high across all four categories, still reasonably good

- Overall intent accuracy is 80.46% but shows imbalance. Strong for common intents and weaker for rare ones

- Overall urgency accuracy is 73.01% but underperforms on low urgency detection and highlights a need for further fine-tuning and class balance

03

Model development - Evaluation of the Advanced Hierarchical Model

The advanced model achieved strong results but showed slight overfitting, particularly reflected in the loss gap and certain misclassification in the urgency and rare intent classes.

To address this, the following strategies are applied:

- **Added L2 Regularization: T**o all dense layers to discourage large weights
- **Dropout increased:** Dropout increased for shared layer from 0.5 to 0.6 and Dense layer from 0.3 to 0.4 to drop the neurons during training, forcing the network to not rely on specific connections
- **Increased Loss weight**: For urgency head from 1.5 to 2.0 to make the model focus on correctly predicting the urgency level
- **Reduced the patience:** To stop the training earlier when the validation stops improving, which reduces continuous memorization of training data.
- **Monitored weighted validation loss:** Monitored the validation loss to ensure learning rate adapts based on overall multitask balance
- **Reduced model architecture:** To prevent excess modal complexity

03

Model development - Evaluation after applying the strategies

Task	Metric	Before	After	Change / Insight
Category	Accuracy	93.25%	93.75%	Small but stable gain: already strong, now slightly more robust
	Macro F1	91.42%	92.02%	Macro average improved: better balance across all 4 classes
Intent	Accuracy	80.46%	81.31%	+0.85% improvement: better handling of minority intents
	Macro F1	79.41%	80.53%	More balanced intent prediction: fewer extreme weak spots
Urgency	Accuracy	73.01%	74.71%	+1.7% improvement, especially improved precision/recall for Class 0
	Macro F1	71.57%	73.49%	Minority Class 2 still challenging, but Class 0 and 1 more stable
Loss Gap	Final Epoch Gap	1.14	0.91	Gap reduced: overfitting controlled
	Mean Gap	0.41	-0.12	Training vs validation loss aligned better (generalization improved)

03 Model development - Test Data Prediction and Evaluation

Task	Dataset	Accuracy	Weighted Precision	Weighted Recall	Weighted F1-score
Category	Training	0.9375	0.9376	0.9375	0.9375
Category	Test	0.9455	0.9455	0.9455	0.9454
Intent	Training	0.8131	0.8210	0.8131	0.8123
Intent	Test	0.8140	0.8246	0.8140	0.8129
Urgency	Training	0.7471	0.7373	0.7471	0.7386
Urgency	Test	0.7571	0.7464	0.7571	0.7462

Urgency Class	Training Precision	Training Recall	Training F1	Test Precision	Test Recall	Test F1
Critical	0.9043	0.8726	0.8882	0.9110	0.8748	0.8925
Normal	0.7629	0.8500	0.8041	0.6028	0.4341	0.5047
Low	0.5857	0.4556	0.5125	0.7642	0.8685	0.8130
Weighted Avg	0.7373	0.7471	0.7386	0.7464	0.7571	0.7462

- Category classification and intent classification show consistent performance across all classes.

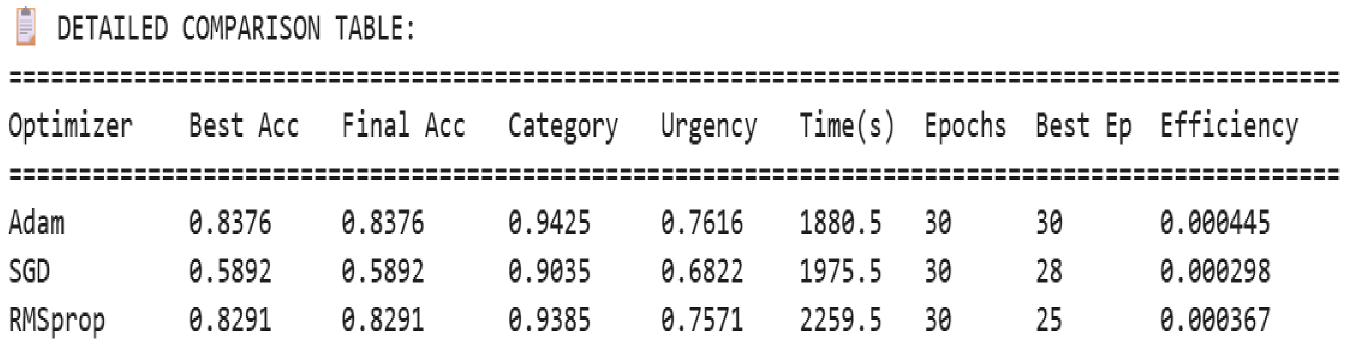
- Urgency classification shows slightly lower performance compared to category and intent due to class imbalance or task complexity. Still shows reasonable predictive capability for priority levels.

- The model is strong at predicting critical urgency, but struggles with normal urgency in the test set (F1:0.8041 to 0.5047) may be due to misclassification or class imbalance, or overlapping vocabulary.

- For low urgency, performance improved in the test set compared to training (F1:0.5125 to 0.8130), suggesting either better learning or dataset distribution differences.

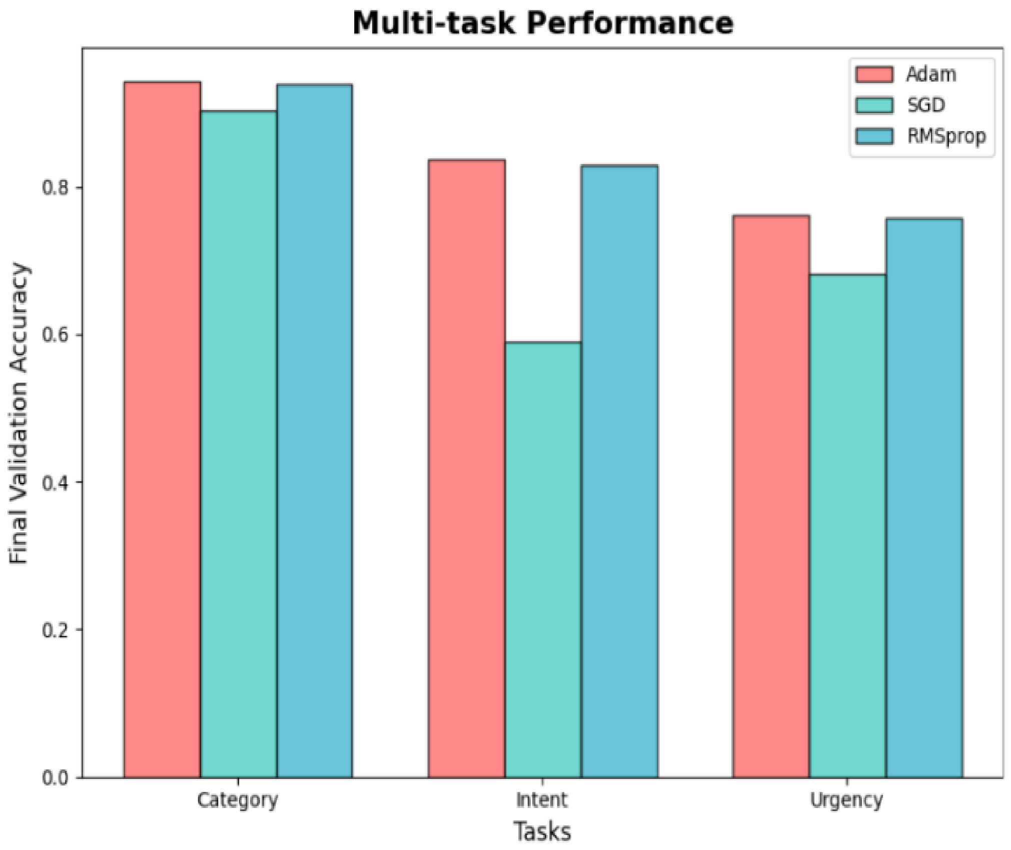
03

Model development - Optimizer performance comparison



```
📋 DETAILED COMPARISON TABLE:
==========================================================================================
Optimizer    Best Acc   Final Acc   Category   Urgency   Time(s)  Epochs  Best Ep  Efficiency
==========================================================================================
Adam         0.8376     0.8376      0.9425     0.7616    1880.5   30      30       0.000445
SGD          0.5892     0.5892      0.9035     0.6822    1975.5   30      28       0.000298
RMSprop      0.8291     0.8291      0.9385     0.7571    2259.5   30      25       0.000367
```

Optimizer	Total Loss	Category Loss	Intent Loss	Urgency Loss	Min Total Loss
Adam	1.2621	0.0998	0.2592	0.4306	1.2621
RMSprop	1.5588	0.1125	0.3916	0.4477	1.5588
SGD	4.9273	0.3389	1.7314	0.7683	4.2672



- Adam: Model achieved highest intent accuracy of 83.76%, the lowest total loss of 1.26, and fast convergence by 30 epochs

- RMSprop: Model achieved 82.91% intent accuracy and 93.85% category accuracy, which is close to Adam's performance and reported a loss of 1.56, better than SGD but slightly behind Adam.

- SGD: Model achieved 58.92% intent accuracy with a much higher total loss of 4.9 compared to Adam and RMSprop

Overall, Adam proved to be the best optimizer for this multi-task model due to its balance of speed, accuracy, and stability.

03 Model development - Learning rate comparison

Learning Rate	Training Time (s)	Best Epoch	Intent Acc	Category Acc	Urgency Acc
0.0005	3495.65	48	0.8326	0.9390	0.7791
0.001	3404.44	40	0.8371	0.9370	0.7571
0.002	3058.80	30	0.8451	0.9340	0.7621

- LR 0.0005: Good generalization with balanced performance across all tasks and indicates stable but slower convergence.
- LR 0.002: Achieved the highest intent accuracy of 0.8451 and a Slight drop in Category (0.9340) and Urgency (0.7621) performance. Has the fastest training time and early convergence in 30 epochs**.**
- LR 0.001: Most balanced trade-off between accuracy and training speed. Converges faster (epoch 40) with good stability.

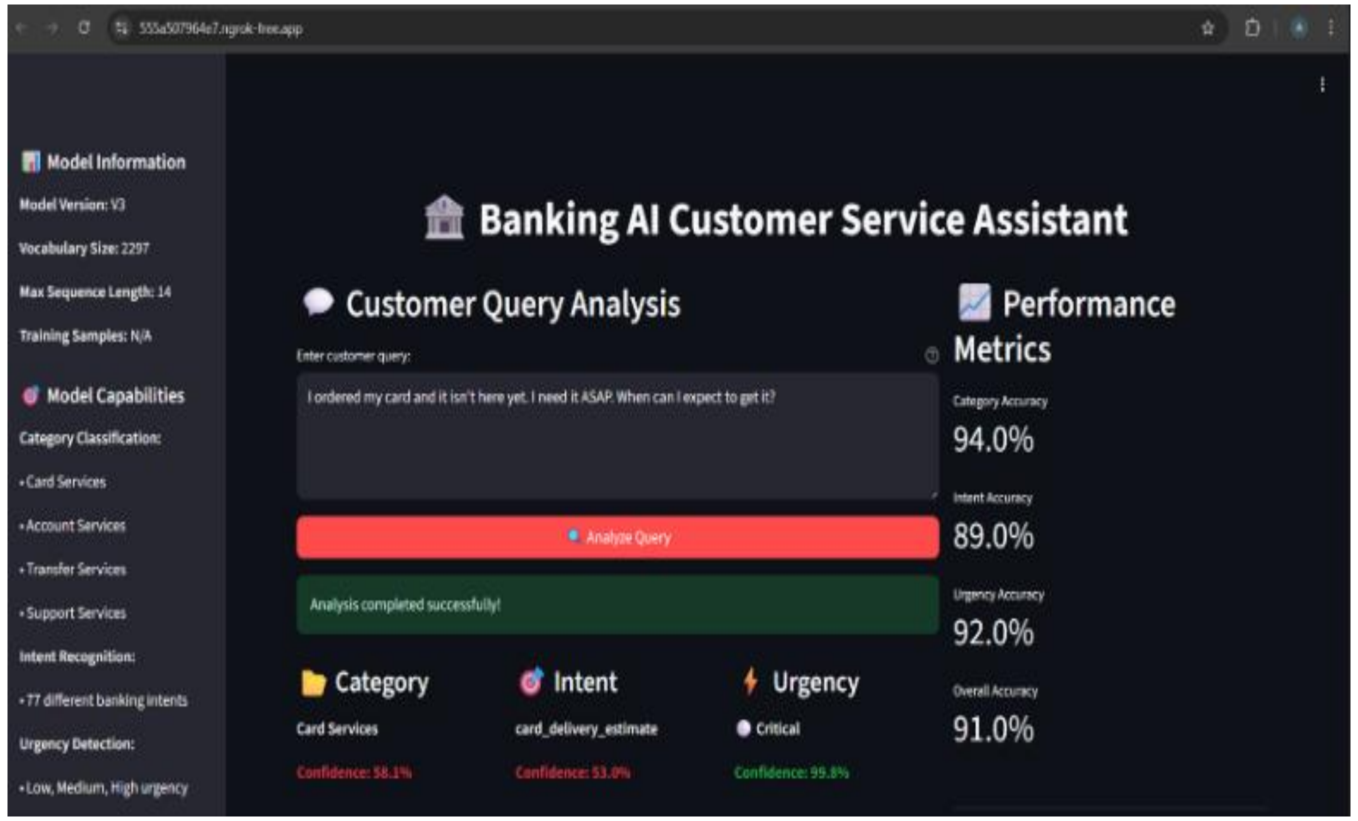
04 Model Deployment - Web Application

- Web application was implemented in Streamlit, providing a frontend and backend in a single Python script.

- The app was hosted using ngrok to create a public URL that enables easy access from any device without to a permanent server

- The interface was designed to be user-friendly with a clear input area, results display and example queries are given for guidance. Confidence levels are given to enhance interpretability and trust in the AI prediction.

- Video Link: [Q4-Web Application.mp4]



Model Information

Model Version: V3

Vocabulary Size: 2297

Max Sequence Length: 14

Training Samples: N/A

Model Capabilities

Category Classification:

- Card Services
- Account Services
- Transfer Services
- Support Services

Intent Recognition:

- 77 different banking intents

Urgency Detection:

- Low, Medium, High urgency

Banking AI Customer Service Assistant

Customer Query Analysis

Enter customer query:

I ordered my card and it isn't here yet. I need it ASAP. When can I expect to get it?

Analysis completed successfully!

Category	Intent	Urgency
Card Services	card_delivery_estimate	● Critical
Confidence: 58.1%	Confidence: 53.0%	Confidence: 99.8%

Performance Metrics

Category Accuracy
94.0%

Intent Accuracy
89.0%

Urgency Accuracy
92.0%

Overall Accuracy
91.0%

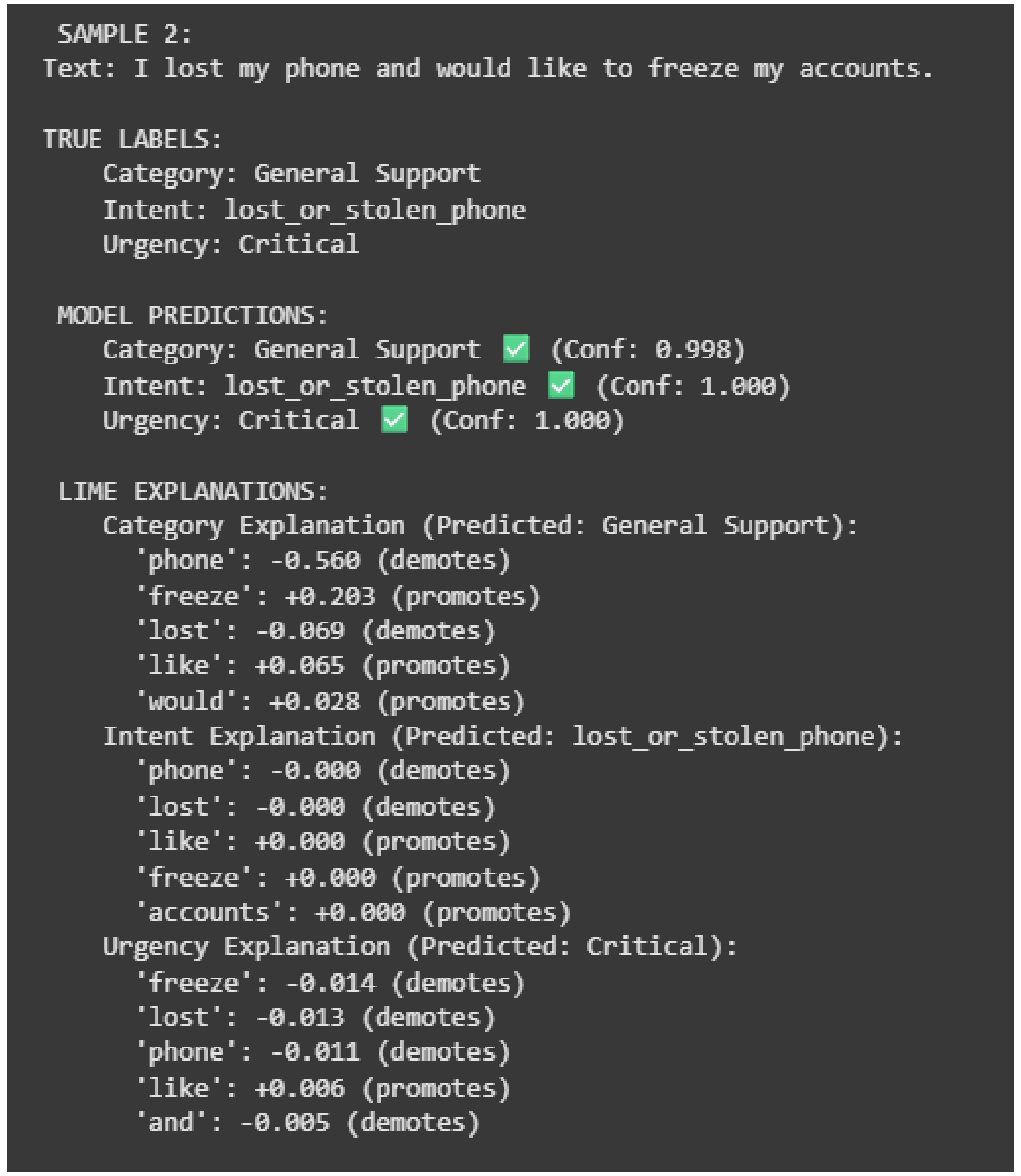
05 Explainability Integration

- Implemented an Explainability feature for the model using LIME (Local Interpretable Model Agnostic Explanations) to analyze the multi-task classification predictions by the model

- Selected few samples and generated explanations for Category, Intent and Urgency predictions.

- Then integrated the explainable feature into the web application in a user-friendly manner.

- Video Link: Q5-Web APP with Explainability.mp4



```
 SAMPLE 2:
Text: I lost my phone and would like to freeze my accounts.

TRUE LABELS:
    Category: General Support
    Intent: lost_or_stolen_phone
    Urgency: Critical

 MODEL PREDICTIONS:
    Category: General Support ✅ (Conf: 0.998)
    Intent: lost_or_stolen_phone ✅ (Conf: 1.000)
    Urgency: Critical ✅ (Conf: 1.000)

 LIME EXPLANATIONS:
    Category Explanation (Predicted: General Support):
      'phone': -0.560 (demotes)
      'freeze': +0.203 (promotes)
      'lost': -0.069 (demotes)
      'like': +0.065 (promotes)
      'would': +0.028 (promotes)
    Intent Explanation (Predicted: lost_or_stolen_phone):
      'phone': -0.000 (demotes)
      'lost': -0.000 (demotes)
      'like': +0.000 (promotes)
      'freeze': +0.000 (promotes)
      'accounts': +0.000 (promotes)
    Urgency Explanation (Predicted: Critical):
      'freeze': -0.014 (demotes)
      'lost': -0.013 (demotes)
      'phone': -0.011 (demotes)
      'like': +0.006 (promotes)
      'and': -0.005 (demotes)
```

06 GenAI Application

- The model used for GenAI application was FLAN-T5, a transformer based larger language model from Hugging Face.

- A smaller version “flan-t5-small” was used for demonstration purposes due to hardware limitations. The model takes a text query as input and predicts three outputs in structured JSON format which was guided by proper prompt engineering.

- The model often returned “General Support” and “Medium” regardless of the query because small model has limited capacity and prompt alone doesn’t provide enough examples for the model to learn and identify the queries to the correct predefined categories and urgency level

- Video link of working demonstration: Q6_GenAI.mp4

Thank you!