Automated Intent Classification for Banking Customers

VAKATI, SAI, VYSHALI

&

SALIM, FATHIMA

APPLIED NATURAL LANGUAGE PROCESSING
FINAL PROJECT IE7500

PROJECT OVERVIEW

Optimize LSTM networks for accurate intent classification.

Utilize the Banking77 dataset, tailored for the banking sector.

Experiment with text preprocessing strategies.

Achieve substantial precision improvements in classifying customer intents.

Importance of Intent Classification in Banking Sector

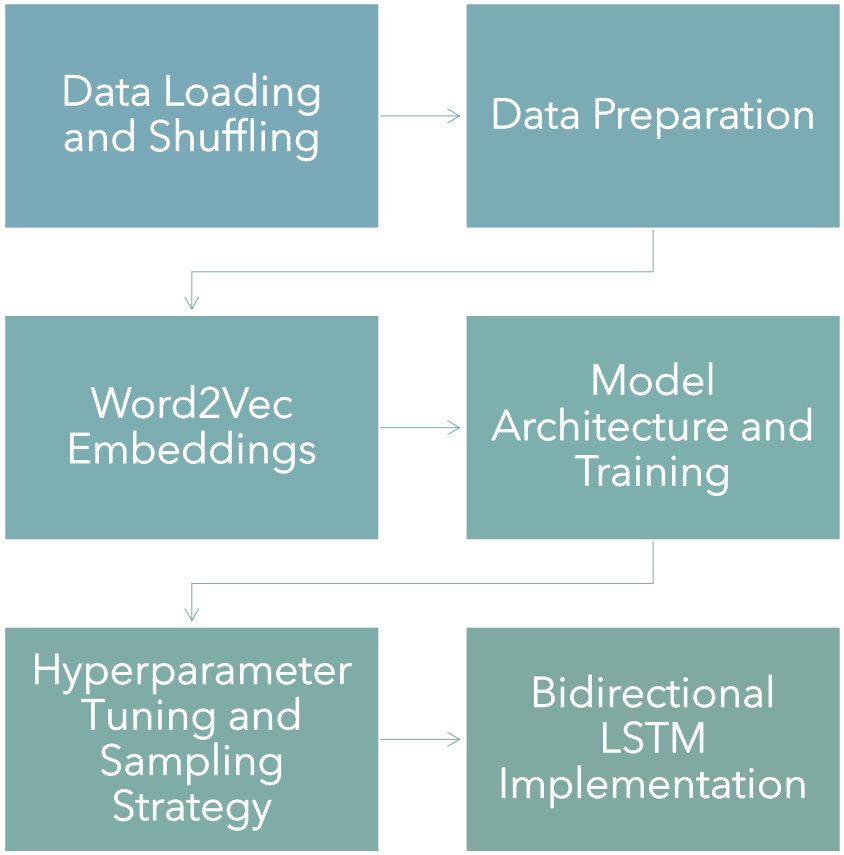
Essential for effective customer service.

Streamlines query routing and response accuracy.

Enhances overall efficiency in handling customer requests.

Provides a foundation for personalized banking experiences.

APPROACH


Data Loading and Shuffling
Data Preparation
Word2Vec Embeddings
Model Architecture and Training
Hyperparameter Tuning and Sampling Strategy
Bidirectional LSTM Implementation

Window Size, and Negative Sampling Effects

Embedding Dimension	Window Size	Negative Samples	Test Accuracy
250	200	20	77.82
250	200	10	76.46
250	100	20	76.40
250	200	5	75.91
250	200	20	75.13
250	300	20	74.97
250	300	10	74.84
250	100	10	74.45
250	100	5	73.41

Impact of text processing on model performance

	WITH PUNCTUATIONS AND STOP WORDS	WITHOUT PUNCTUATIONS AND STOP WORDS
LEMMATIZED	81.69%	75.62%
NOT LEMMATIZED	82.40%	79.48%

Implementing

OOV words

handling and

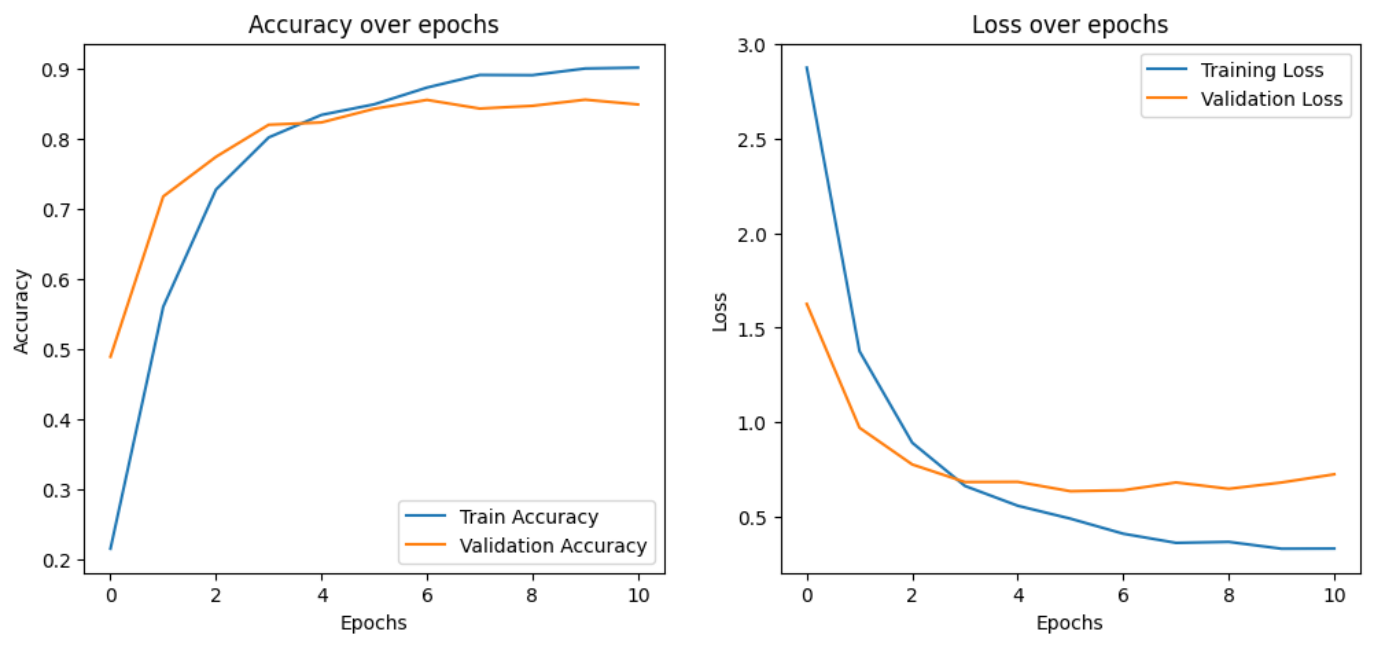
Sentence

contractions

ACCURACY: 83.86%

FINAL MODEL WITH HYPERPARAMETER TUNING AND IMPLEMENTATION OF BIDIRECTIONAL LSTM

```
[4] Epoch 1/30
313/313 [==============================] - 207s 607ms/step - loss: 3.0021 - accuracy: 0.1936 - val_loss: 1.6335 - val_accuracy: 0.5052
Epoch 2/30
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file forma
  saving_api.save_model(
313/313 [==============================] - 219s 699ms/step - loss: 1.4060 - accuracy: 0.5663 - val_loss: 1.0467 - val_accuracy: 0.6860
Epoch 3/30
313/313 [==============================] - 193s 616ms/step - loss: 0.9017 - accuracy: 0.7213 - val_loss: 0.8280 - val_accuracy: 0.7747
Epoch 4/30
313/313 [==============================] - 189s 606ms/step - loss: 0.6772 - accuracy: 0.7933 - val_loss: 0.7710 - val_accuracy: 0.7893
Epoch 5/30
313/313 [==============================] - 181s 578ms/step - loss: 0.5901 - accuracy: 0.8248 - val_loss: 0.7433 - val_accuracy: 0.8162
Epoch 6/30
313/313 [==============================] - 189s 604ms/step - loss: 0.4998 - accuracy: 0.8500 - val_loss: 0.7364 - val_accuracy: 0.8136
Epoch 7/30
313/313 [==============================] - 188s 602ms/step - loss: 0.4345 - accuracy: 0.8694 - val_loss: 0.7376 - val_accuracy: 0.8247
Epoch 8/30
313/313 [==============================] - 188s 600ms/step - loss: 0.3943 - accuracy: 0.8820 - val_loss: 0.7559 - val_accuracy: 0.8227
Epoch 9/30
313/313 [==============================] - 189s 601ms/step - loss: 0.3728 - accuracy: 0.8865 - val_loss: 0.7491 - val_accuracy: 0.8393
Epoch 10/30
313/313 [==============================] - 190s 606ms/step - loss: 0.3593 - accuracy: 0.8925 - val_loss: 0.7222 - val_accuracy: 0.8468
Epoch 11/30
313/313 [==============================] - 190s 607ms/step - loss: 0.3193 - accuracy: 0.9073 - val_loss: 0.7696 - val_accuracy: 0.8445
Epoch 12/30
313/313 [==============================] - 194s 619ms/step - loss: 0.3148 - accuracy: 0.9062 - val_loss: 0.7490 - val_accuracy: 0.8432
Epoch 13/30
313/313 [==============================] - 190s 609ms/step - loss: 0.3083 - accuracy: 0.9117 - val_loss: 0.7370 - val_accuracy: 0.8442
Epoch 14/30
313/313 [==============================] - 187s 599ms/step - loss: 0.2850 - accuracy: 0.9134 - val_loss: 0.8515 - val_accuracy: 0.8435
Epoch 15/30
313/313 [==============================] - 187s 599ms/step - loss: 0.2909 - accuracy: 0.9128 - val_loss: 0.8130 - val_accuracy: 0.8432
Epoch 15: early stopping
97/97 [==============================] - 19s 133ms/step - loss: 0.7222 - accuracy: 0.8468
Test Accuracy: 84.68%
Test Loss: 0.7222
```


Accuracy over epochs
Accuracy
Epochs
Train Accuracy
Validation Accuracy
Loss over epochs
Loss
Epochs
Training Loss
Validation Loss

	precision	recall	f1-score	support
activate_my_card	0.93	0.95	0.94	40
age_limit	0.95	0.93	0.94	40
apple_pay_or_google_pay	0.98	1.00	0.99	40
atm_support	0.91	0.97	0.94	40
automatic_top_up	0.92	0.82	0.87	40
balance_not_updated_after_bank_transfer	0.67	0.82	0.74	40
balance_not_updated_after_cheque_or_cash_deposit	0.94	0.85	0.89	40
beneficiary_not_allowed	0.86	0.75	0.80	40
cancel_transfer	0.89	0.85	0.87	40
card_about_to_expire	0.97	0.90	0.94	40
card_acceptance	0.93	0.70	0.80	40
card_arrival	0.84	0.90	0.87	40
card_delivery_estimate	0.71	0.75	0.73	40
card_linking	0.95	0.93	0.94	40
card_not_working	0.80	0.88	0.83	40
card_payment_fee_charged	0.80	0.80	0.80	40
card_payment_not_recognised	0.76	0.85	0.80	40
card_payment_wrong_exchange_rate	0.89	0.85	0.87	40
card_swallowed	0.96	0.68	0.79	40
cash_withdrawal_charge	0.88	0.90	0.89	40
cash_withdrawal_not_recognised	0.70	0.88	0.78	40
change_pin	0.90	0.90	0.90	40
compromised_card	0.77	0.60	0.68	40
contactless_not_working	1.00	0.80	0.89	40

```
text1 = ["Someone grabbed my card and ran away,now I dont have access to the card"]
predictclass(text1)
```

```
1/1 [==============================] - 0s 184ms/step
'lost_or_stolen_card'
```


THANKYOU!