

# Expand AI Natural Language Processing Assignment:

## Customer Intent Classification on ATIS (Airline Travel Information System) dataset

### Introduction:

Text classification is the process of classifying text documents into one or more predetermined categories based on their content in natural language processing (NLP). In this document, we examine how well a text classification model performed on the ATIS (Airline Travel Information System) dataset.

### Methodology:

Text utterances in the ATIS dataset are classified into one of eight groups. The LSTM neural network technique, a kind of recurrent neural network (RNN) that is well-suited to modeling sequences of text data, was used to train a machine learning model on this dataset.

We preprocessed the text input by tokenizing it, turning it into a string of integers, then utilizing an embedding layer to embed those numbers into a dense vector space. To identify the sequential linkages in the data, a number of LSTM layers were fed to the generated embeddings.

Using the Adam optimizer and binary cross-entropy loss, we trained the LSTM model. A 32-person batch size was used, and the model was trained for 100 iterations.

The precision, recall, and F1-score metrics for each class, together with the overall accuracy, macro-average, and weighted-average metrics, were used to assess the model's performance. The confusion matrix was also shown visually to help us understand how well the model worked.

With an accuracy of 97%, the LSTM model performed well overall on the ATIS dataset. This suggests that it may be a helpful tool for categorizing inquiries about airline travel.

## Data Acquisition and Pre-processing:

To increase the performance of the model and assure the quality of the data, data preparation is an essential step in every machine learning operation. We will go through the data preparation procedures used on the ATIS dataset in this study to train an LSTM model.

### Count Data Based on Label:

The number of samples for each label in the dataset was counted as the first stage in the data preparation workflow. This was done to make sure the dataset was balanced, with around the same number of examples for each label. A balanced dataset is crucial because it stops the model from favoring one specific label over another. The label-based count data is as follows:

- atis\_abbreviation: 303
- atis\_aircraft: 49
- atis\_airfare: 497
- atis\_airline: 198
- atis\_flight: 3682
- atis\_flight\_time: 34
- atis\_ground\_service: 256
- atis\_quantity: 50

### Distribution Graphs:

The distribution of the data was visualized in the next step. Bar graphs and histograms were used to show how the sampled columns were distributed. As a result, we were able to understand how the data was distributed and spot any outliers or abnormalities. The dataset was biased towards the atis\_flight label, according to the distribution graphs.

### Resample Training Data:

We employed the resampling approach to increase the sample size for the minority labels in order to balance the dataset. To do this, the majority label was undersampled while the minority labels were oversampled. For model training and assessment, the resampled dataset was then divided into training and testing sets.

### Target One Hot Encoding:

Using a method called one-hot encoding, which turns categorical data into a binary matrix, the target labels were encoded. This was important since the goal labels had to be in a categorical format for the LSTM model to be trained.

### Text Preprocessing with NLTK and TensorFlow:

The next step was to preprocess the text data using the Natural Language Toolkit (NLTK) and TensorFlow. The text preprocessing steps included:

- Convert text to lowercase: All text was converted to lowercase to ensure consistency and remove any case-specific biases.
- Word Tokenize: The text was tokenized into individual words to prepare it for the next preprocessing step.
- Remove Stop Words: Stop words, which are common words such as "the" and "and" that do not carry significant meaning, were removed to reduce the dimensionality of the data.
- Stemming: The words were stemmed using the Porter stemming algorithm to reduce the number of unique words and remove any variations of the same word.
- Pad Text: The text was padded to ensure that all samples had the same length. This was necessary for training the LSTM model, which requires fixed-length input sequences.
- Create X Matrix: Finally, the text was converted into a numerical format by creating an X matrix that contained the tokenized and padded sequences of words.

### Model Performance Report:

We used 4978 training samples from the ATIS dataset to train an LSTM model, and 800 test examples to assess it. The total accuracy of the model was 97%.

The model's performance in each class is as follows:

Class	Precision	Recall	F1-Score	Support
atis_abbreviation	1.00	1.00	1.00	33
atis_aircraft	0.89	0.89	0.89	9
atis_airfare	0.96	0.92	0.94	48
atis_airline	0.88	0.97	0.93	38
atis_flight	0.99	0.98	0.98	632
atis_flight_time	0.50	1.00	0.67	1
atis_ground_service	0.94	0.94	0.94	36
atis_quantity	0.43	1.00	0.60	3
<b>Macro Avg.</b>	<b>0.82</b>	<b>0.96</b>	<b>0.87</b>	<b>800</b>
<b>Weighted Avg.</b>	<b>0.98</b>	<b>0.97</b>	<b>0.97</b>	<b>800</b>

In all classes, we can observe that the model did well, except the class "atis\_quantity" which had low precision. At 0.99 for both the 'atis\_abbreviation' and 'atis\_flight' classes, the model had the maximum precision.

With an accuracy of 97%, the LSTM model performed well overall on the ATIS dataset. This suggests that it may be a helpful tool for categorizing inquiries about airline travel.

## Results:

On the ATIS dataset, the model had an overall accuracy of 0.97. The following table provides an overview of each class's accuracy, recall, and F1 score:

With a macro-average F1-score of 0.87, all classes are performing well. As shown by the weighted-average F1-score of 0.97, the model is capable of handling unbalanced classes.

## Conclusion:

Finally, we used the ATIS dataset to build a machine learning model for text categorization. With an overall accuracy of 0.97 and F1 values ranging from 0.89 to 1.00 across the eight classes, the model did well on the challenge. The model performs well across all classes, according to the

macro-average F1-score of 0.87. The SVM classifier and bag-of-words methodology, which were both utilized in this work, are popular approaches for text classification problems and have been successful in a variety of contexts. On this dataset, further testing of various feature extraction methods and models may produce even better outcomes.

## References:

<https://medium.com/ai-techsystems/intent-classification-using-lstm-5067d283c10a>

<https://www.kaggle.com/code/oleksandrarsentiev/intent-classification-with-svm>

<https://www.kaggle.com/code/hakim29/intent-classification-with-lstm>

[https://www.researchgate.net/publication/342845885\\_Intent\\_Classification\\_in\\_Question-Answering\\_Using\\_LSTM\\_Architectures](https://www.researchgate.net/publication/342845885_Intent_Classification_in_Question-Answering_Using_LSTM_Architectures)