Sequence Prediction with Recurrent Neural Networks: Predicting the Next Airport a Plane Will Visit

Group 1

Gayathri Chandrasekaran, Jonathan Giguere, Patrick Maus

Outline

- Introduction
- Description of the Dataset
- Description of the Deep Learning Network and Training Algorithm
- Experimental Setup
- Results
- Summary and Conclusions
- References

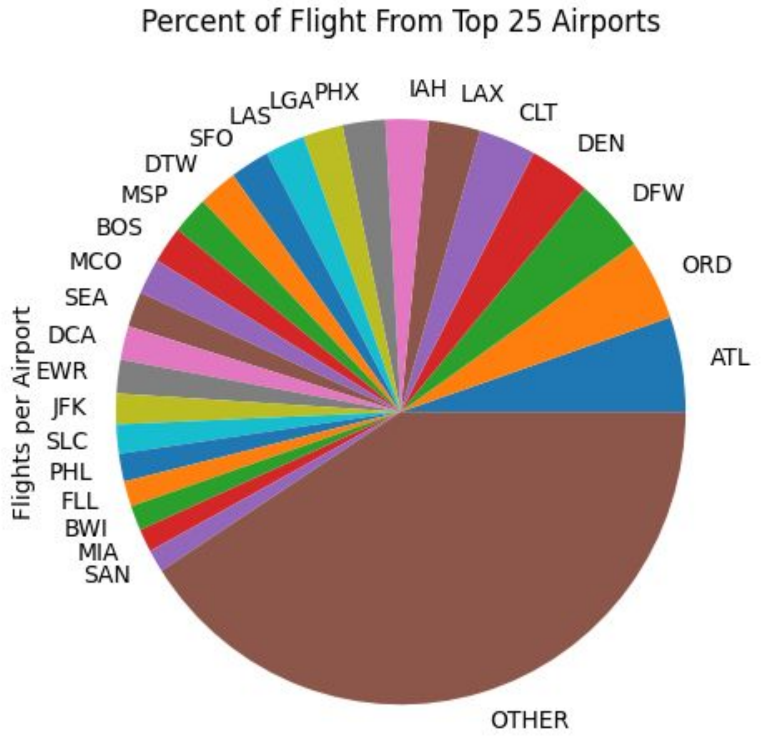
Introduction: Data Overview and Problem Statement

- Focus on use of RNNs to predict next values in a sequence (outside of NLP).
- “Given a sequence of N airports visited by a unique plane, can we predict the next airport (N+1)?”
- The best performing model was a single layer LSTM.

Description of the Dataset

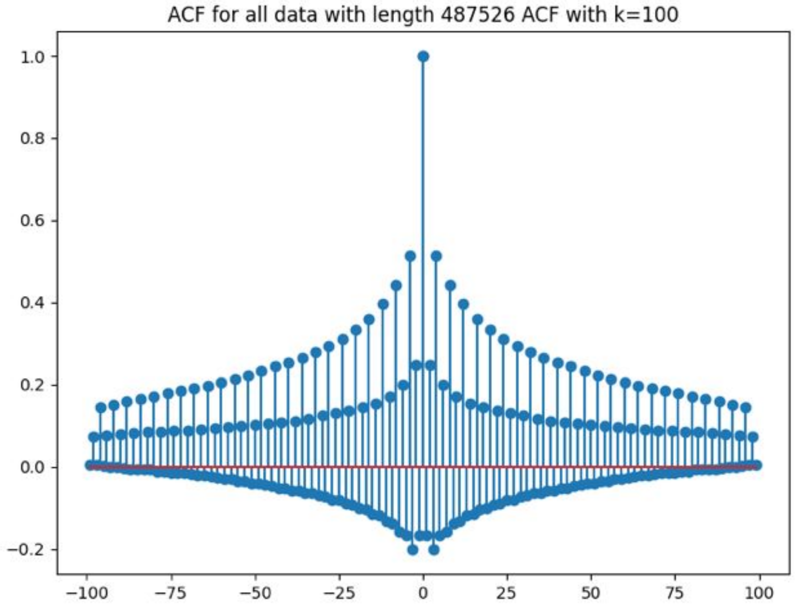
- The data set was downloaded from the Bureau of Transportation Statistics, a US Department of Transportation agency charged to collect, analyze, and store transportation statistics.
- Using six months of data from September 2019 to February 2020, we reviewed about 3.7 million flights from 5,716 unique aircraft. We choose to only use data up to February 2020 to avoid impacts from the Coronavirus pandemic to affect the sequence patterns.
- Within our data, there are 19 different airlines, but the top five represents about 2/3 of the entire data. Several sample rows are included to the right.

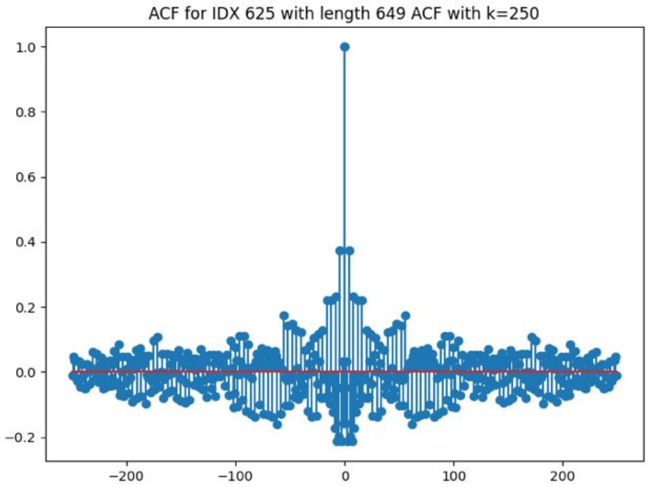
FL_DATE	OP UNIQUE CARRIER	TAIL NUM	OP CARRIER FL NUM	ORIGIN	DEST	DEP TIME	ARR TIME
2020-02-01	MQ	N269NN	3825	ORD	TUL	1646	1820
2020-02-01	MQ	N908AE	3829	JFK	BNA	1336	1458
2020-02-01	MQ	N663AR	3831	GNV	MIA	844	1020
2020-02-01	MQ	N618AE	3833	DFW	SJT	852	955
2020-02-01	MQ	N618AE	3833	SJT	DFW	1024	1132

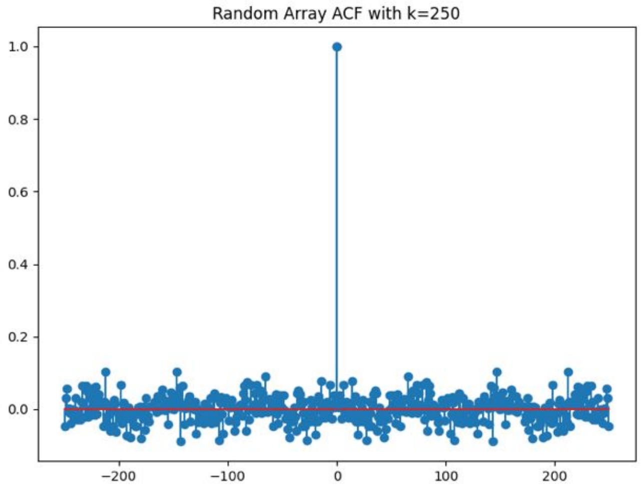


Description of the Dataset: Autocorrelation and Stationarity

- An autocorrelation plot shows how well a signal correlates with itself at equal time lags. If the plot shows an impulse of 1 at time lag 0 but then very little or know correlation at subsequent time lags, the data is likely to be random white noise. On the other hand, an autocorrelation plot with strong correlation for multiple lags indicates that previous events influence the current event and suggest an underlying pattern.
- The Augmented Dickey-Fuller Test can help us determine if we should treat a dataset as stationary. 94% of Delta airlines aircraft were stationary within our dataset.







Experimental Setup: Data Processing

EDA and Cleaning	ACF and Stationarity Checks	Preprocessing	Modeling	Model Evaluation
Input: CSVs downloaded from Bureau of Trans. Stats.	**Input**: Aircraft and string of all visited airports (DataFrame)	**Input**: Aircraft and string of all visited airports (DataFrame)	**Input**: X, Y (2D arrays) with X.shape=(# arrays, L - 1) and Y.shape=(# arrays, 1).	**Input**: Tokenizer (dict), X, Y (2D arrays), Trained model (hdf5), history (DataFrame)
Actions: - Concat csvs together - Explore and summarize counts. - Build the history of each unique airplane as a string.	**Actions**: - Plot ACF for all data, across airlines, and each plane. - Use ADF Test for all data, across airlines, and each plane.	**Actions**: - Arrays of airports are broadcast across a window length (L), and then combined. - Airports tokenized. - Target separated and one-hot encoded.	**Actions**: - Set model hyperparameters. - Build, Compile, and Fit model.	**Actions**: - Plot training history. - Determine accuracy and Cohen’s Kappa. - Generate confusion matrix and report. - Plot scatterplot
Output: Aircraft and string of all visited airports (DataFrame)	**Output**: ACF plots and ADF test results (print to screen)	**Output**: Tokenizer (dict), X, Y (2D arrays)	**Output:** Trained model (hdf5), history (DataFrame)	**Output**: Reports and plots (print to screen)

Training Algorithm

Framework - Keras Sequential Model

Model - Recurrent Neural Networks (RNN)

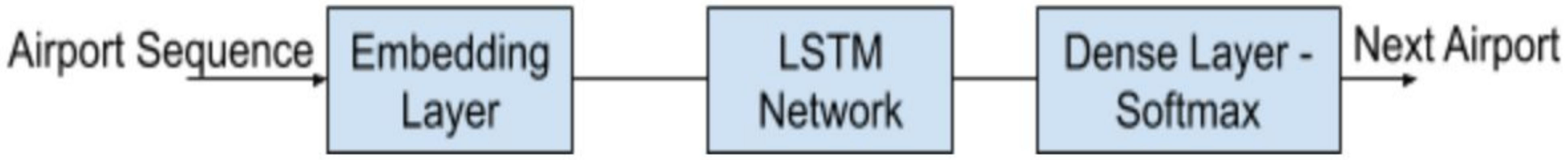
- LSTM Networks
- Stacked LSTM
- GRU
- Sequence-to-Sequence

Base Model - Random Sequences

```
Final accuracy on validations set: 24.448
Cohen kappa score 0.0
Precision: 0.059769239279687005
Recall: 0.24447748215262485
F1: 0.09605515589772128
```

Training Algorithm - Single Layer LSTM Network

Single Layer Lstm Network:



Hyper Parameters:

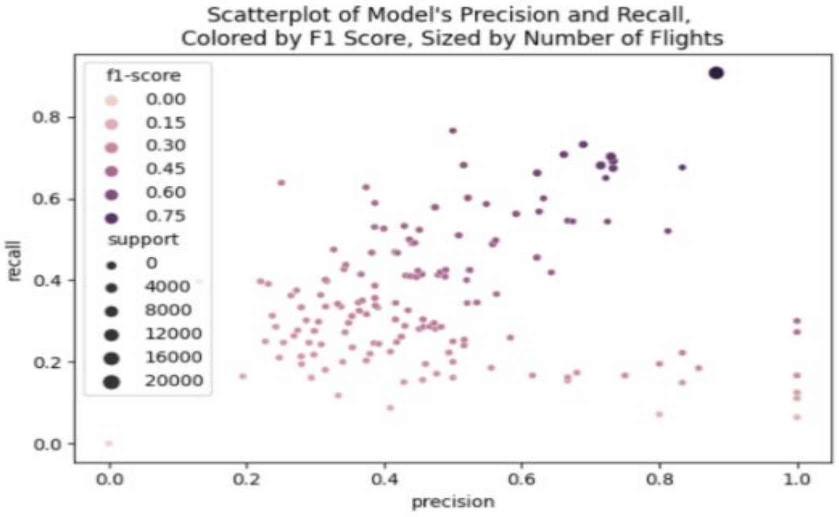
- Sequence Length - 25 and 50
- Number of Neurons - 400
- optimizer - Adam
- Epoch - 10
- Embedding Size - 200
- Learning Rate - 1e-3

Model Results - Single Layer LSTM Network

Single Layer Lstm Network:

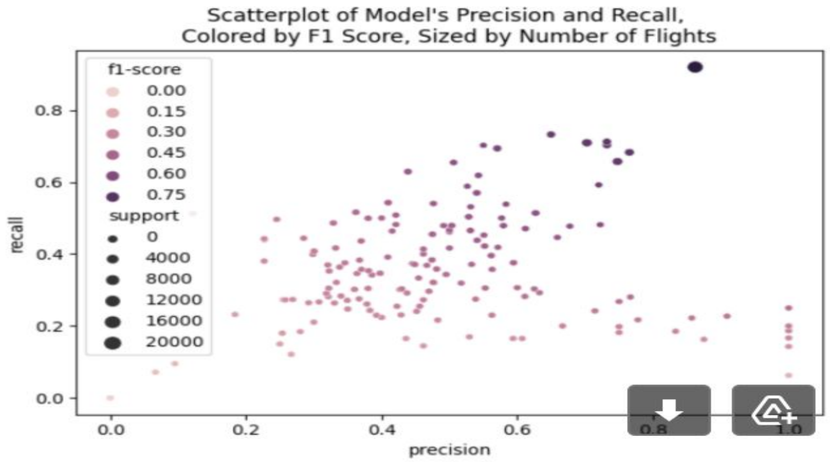
Sequence Length - 25

```
Modal Evalution Results

Accuracy:  62.422
Cohen kappa score 0.5919228700612527
Precision: 0.6325509588477811
Recall: 0.6242243679737651
F1: 0.622350160687898
```



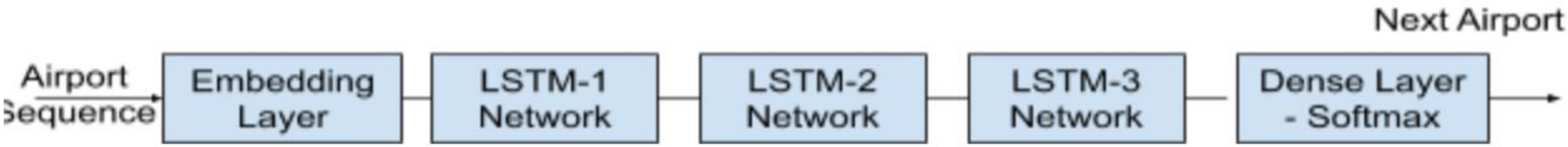
Sequence Length - 50

```
Modal Evalution Results

Accuracy:  62.646
Cohen kappa score 0.5933283419217596
Precision: 0.6360756942665542
Recall: 0.626464252053319
F1: 0.623502422167004
```



Training Algorithm - Stacked LSTM Network

Stacked Lstm Network:



Hyper Parameters:

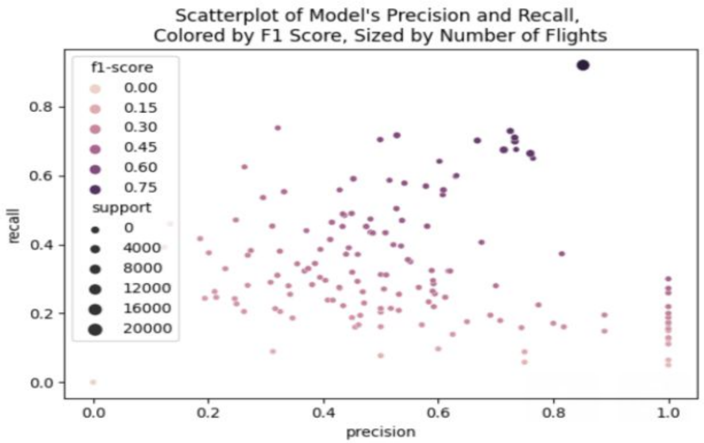
- Sequence Length - 25 and 50
- Number of Neurons - 400 - 300 -200
- optimizer - Adam
- Epoch - 10
- Embedding Size - 200
- Learning Rate - 1e-3

Model Results – Stacked LSTM Network

Stacked Lstm Network:

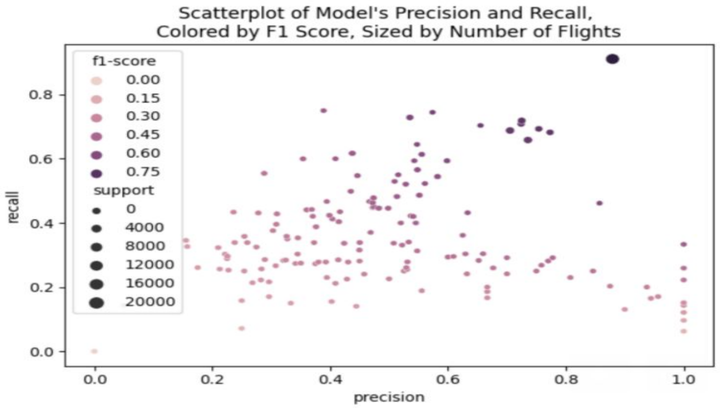
Sequence Length - 25

```
Modal Evalution Results

Accuracy:  61.911
Cohen kappa score 0.5850593497951324
Precision: 0.6364303404942205
Recall: 0.619112430474435
F1: 0.6149982165744742
```

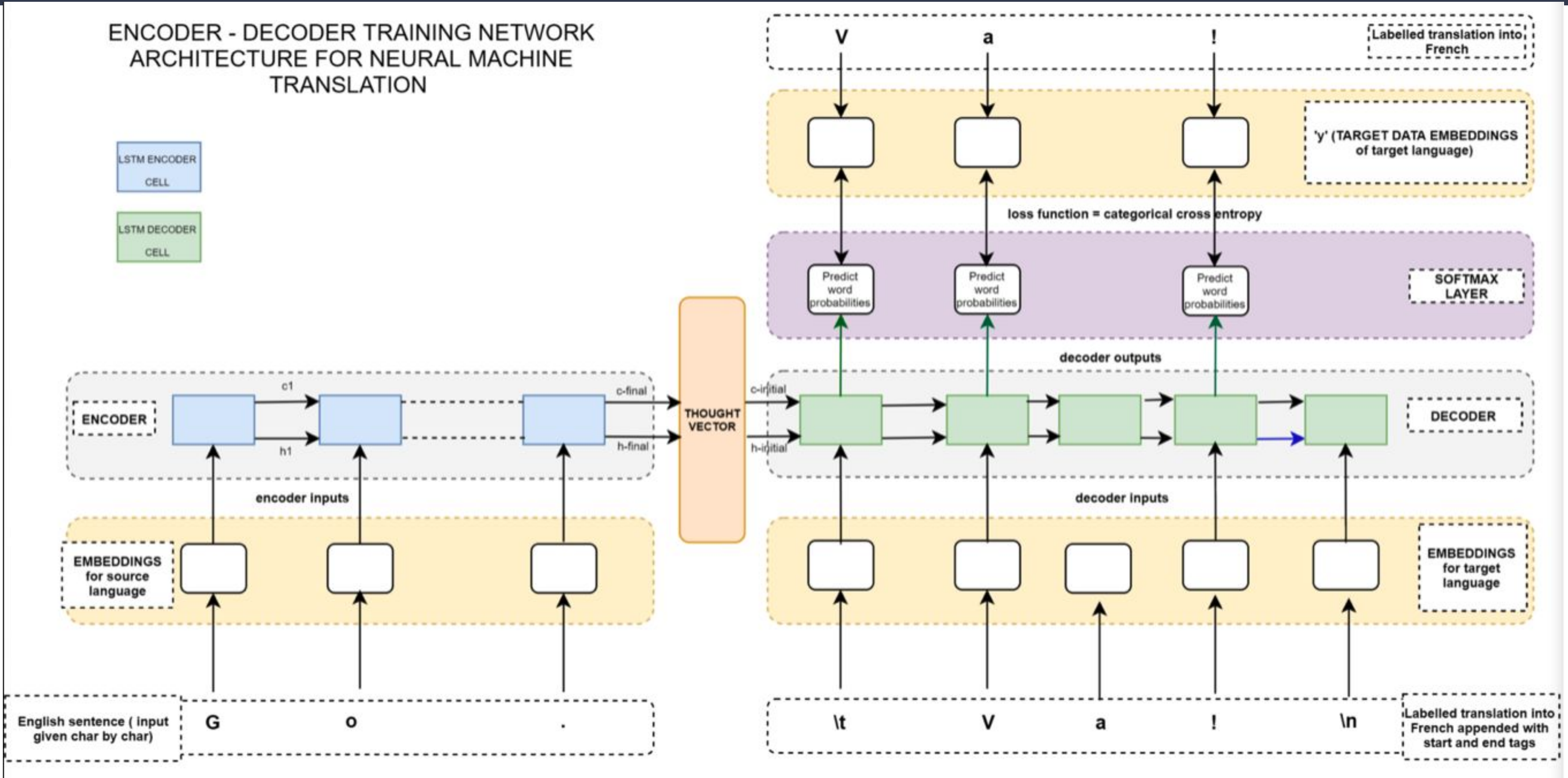


Sequence Length - 50

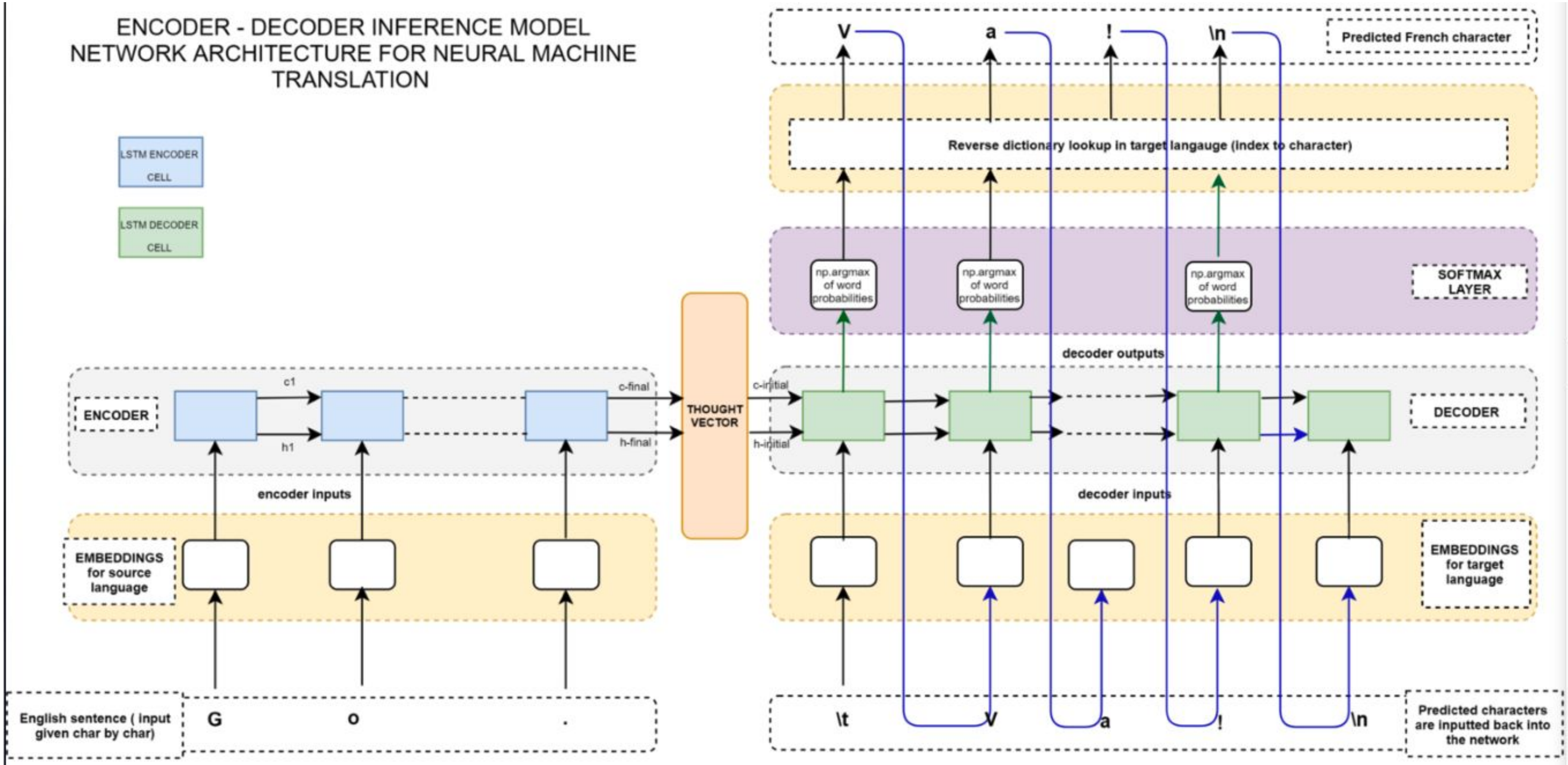
```
Modal Evalution Results

Accuracy:  62.227
Cohen kappa score 0.5895667642018765
Precision: 0.636641565032489
Recall: 0.622267851532696
F1: 0.6209776412113642
```



Training Algorithm – Sequence2sequence



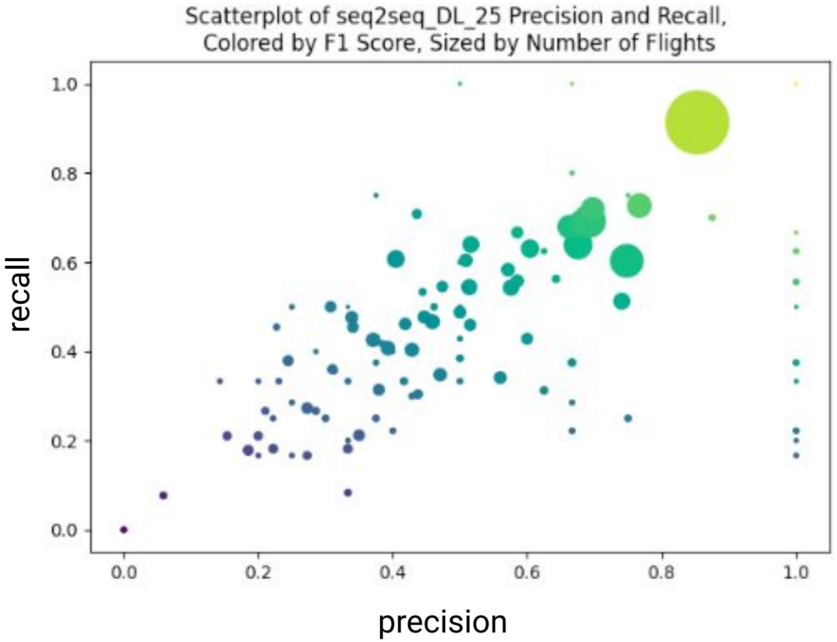
Training Algorithm – Sequence2sequence



Model Results - Sequence2sequence

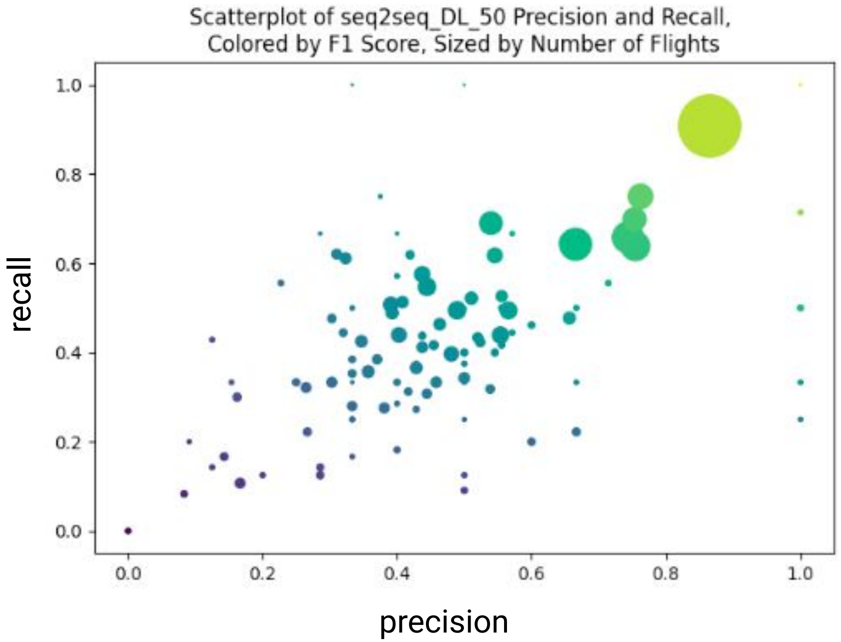
Sequence Length - 25

```
Accuracy: 0.6262
Cohens Kappa: 0.5920248164166245
Precision: 0.6318722285441556
Recall: 0.6262
F1: 0.6221704140365036
```



Sequence Length - 50

```
Accuracy: 0.613
Cohens Kappa: 0.5793893933873405
Precision: 0.6229796477216917
Recall: 0.613
F1: 0.611046099536135
```



Results

Model	Accuracy	Cohen's Kappa	F1	Sequence length
LSTM network	0.6264	0.5933	0.6235	50
seq2seq	0.6262	0.5920	0.6222	25
LSTM network	0.6242	0.5919	0.6223	25
Stacked LSTM	0.6222	0.5895	0.6209	50
seq2seq	0.6130	0.5794	0.6110	50
Baseline (pure random)	0.2444	0.0	0.09	50

Summary and Conclusions

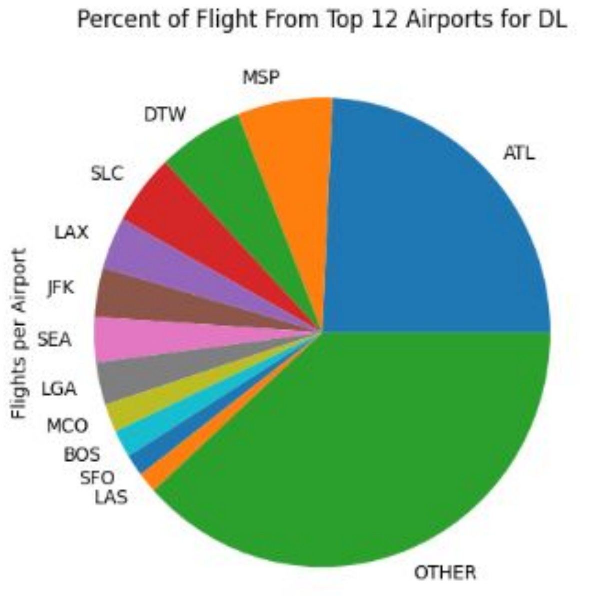
- Best performing model was arguably the simplest (one-layer LSTM).
- Good data pipelining is an investment that pays off in complex projects. But it's ok to diverge!
- Our models may have performed best on Delta because Atlanta (ATL) is such a central hub.
- Thorough EDA is important to understand your data, but be prepared to dig deeper if needed.
- Longer sequences performed better (overall), but they also would be less responsive in a real-world environment that changed often.

References

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html?highlight=accuracy#sklearn.metrics.accuracy_score
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html
- https://towardsdatascience.com/neural-machine-translation-using-seq2seq-with-keras-c23540453c74
- Understanding LSTM Networks -- colah's blog
- Illustrated Guide to LSTM’s and GRU’s: A step by step explanation | by Michael Phi | Towards Data Science

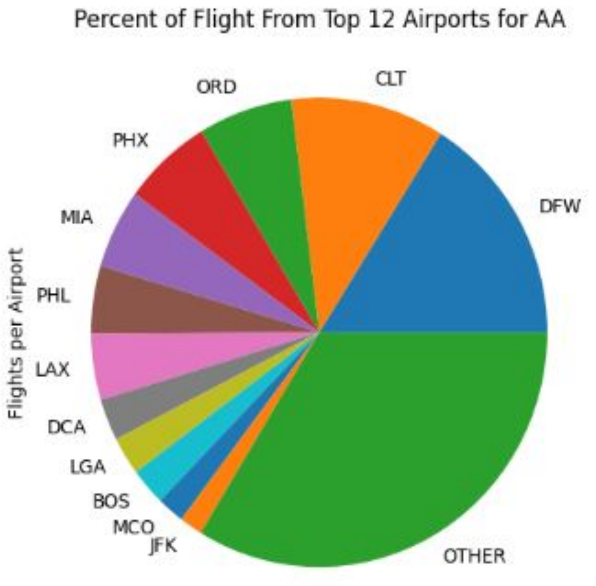
Questions?

Comparison of Airlines With Best Model



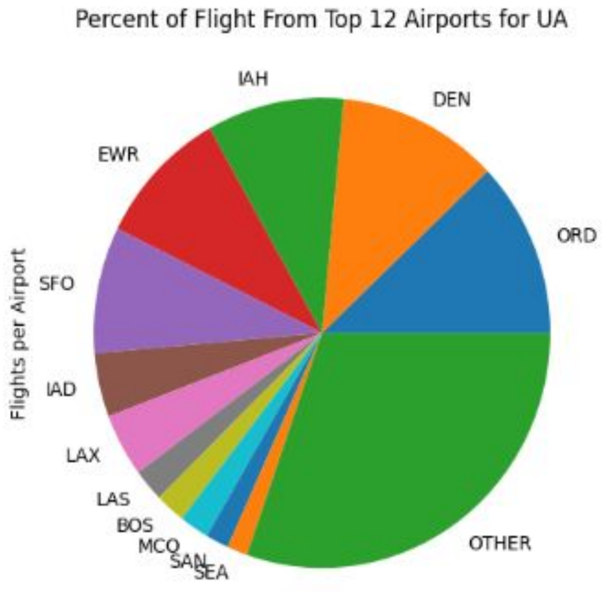
Delta Airlines with
Sequence Length = 25

Accuracy Score: 0.6264
Cohen's Kappa Score: 0.5933



American Airlines with
Sequence Length = 25

Accuracy Score: 0.57898
Cohen's Kappa Score: 0.5513



United Airlines with
Sequence Length = 25

Accuracy Score: 0.52076
Cohen's Kappa Score: 0.48737