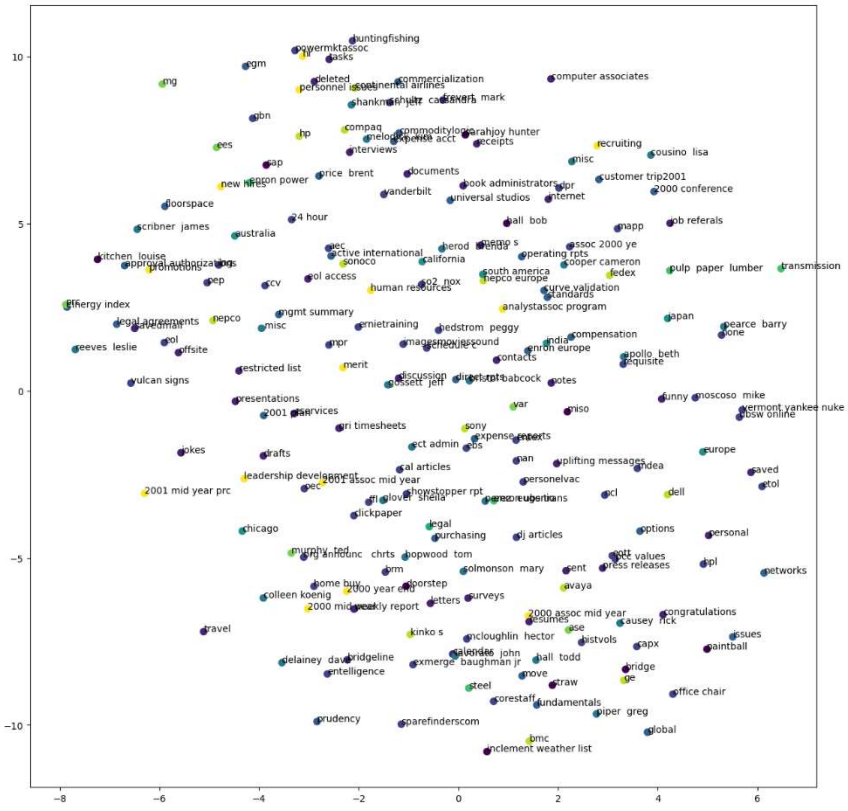
CMPT 310
Milestone 2
Albert Hong

SiftMail

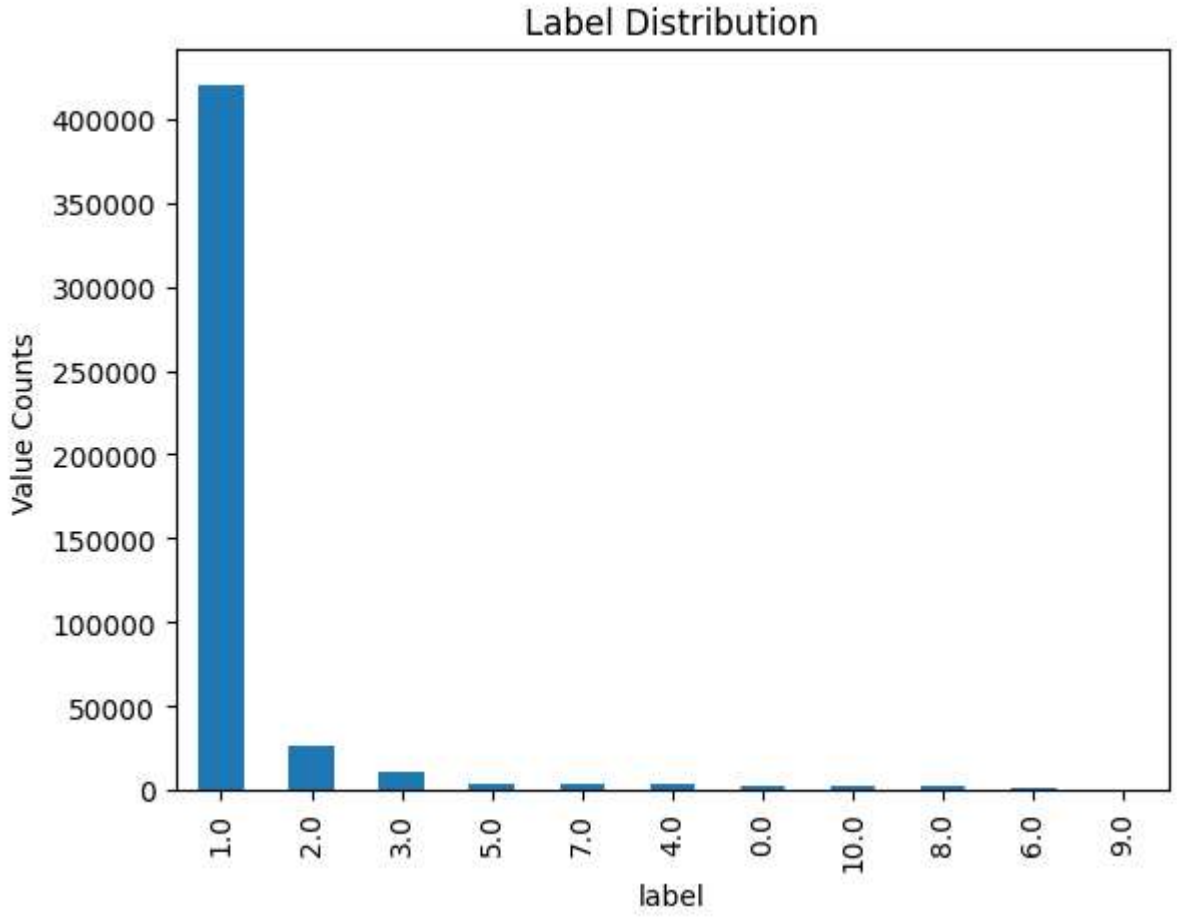
Description: SiftMail is a machine learning project for email classification that uses LSTM neural networks to automatically categorize emails into different categories. It uses TensorFlow/Keras for the LSTM model and NLP preprocessing with tokenization and sequence padding. The AI is tasked with categorizing emails into 11 predefined categories based on its content.

Significant Accomplishments:

Data Collection: We moved on from using the spamham which included prelabeled data (spam or ham) to using a real-world example using enron email dataset. We implemented email parsing to extract header (date, subject, etc), message body and metadata. Then we applied NLTK stopwords removal and text normalization. In addition, the original dataset had over 1000 unique folder names, and thus we implemented KNN clustering approach using Word2Vec embedding to reduce the number of labels down to 11 categories.

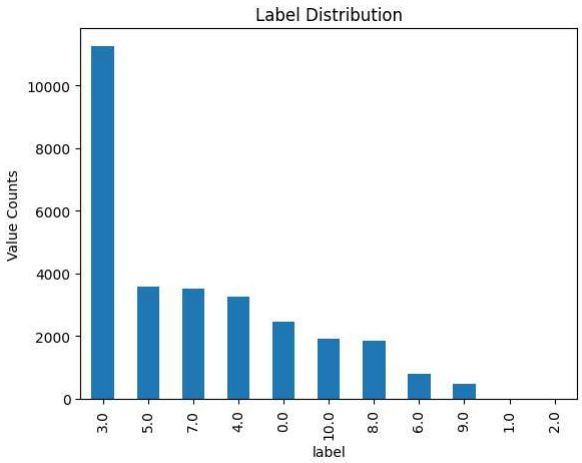


CMPT 310
Milestone 2
Albert Hong



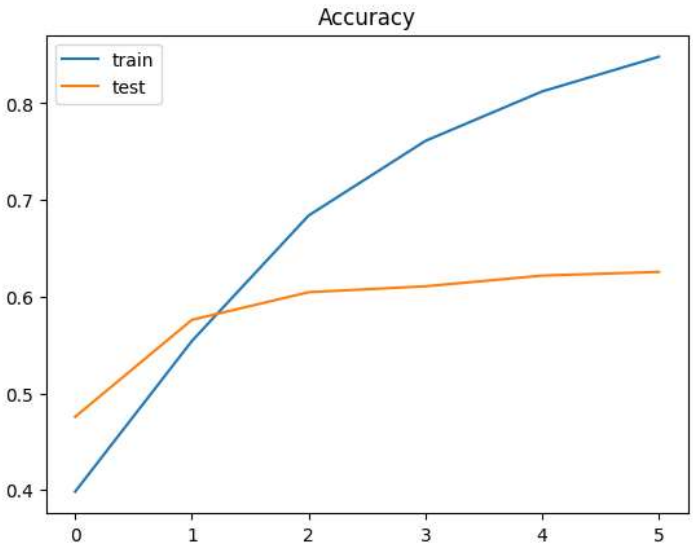
(Label 1 being (sent mail) and Label 2 being emails that we could not classify)

We can see how the label distribution is heavily skewed towards label 1 and 2, so we opted to drop it, which resulted to this.



CMPT 310
Milestone 2
Albert Hong

Model Design and Training + Visualizations: For this project we experimented with multiple approaches. From the spamham dataset we found out that decision trees and random forest were the best choice of models. We then used an LSTM Neural Network as our final model. Where the input is a text sequence, embedding layers with pre-trained Word2Vec vectors, LSTM layers for sequence processing and a dense output layer for 11 class classification. The performance of this model had an 87% accuracy during training, and a validation accuracy of 62%. We are still working to improve the parameters of our model.



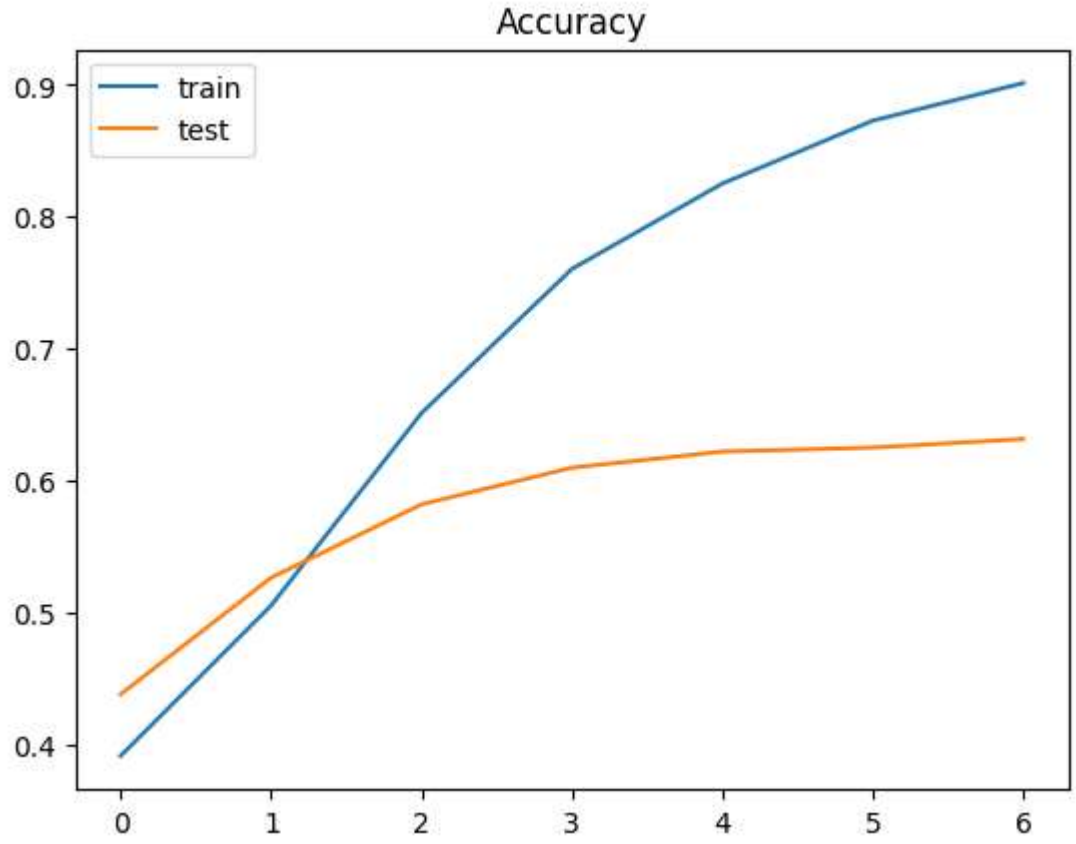
Proof of Accomplishments

```
new_complaint = ['Please transfer the total amount to the clients bank account']
seq = tokenizer.texts_to_sequences(new_complaint)
padded = pad_sequences(seq, maxlen=max_length)
pred = model.predict(padded)
labels = ['Weather/Natural','Sent Mail','Random/NA','Financial/Logistics','Related to Other People',
          'Places','Legal','Buisness','2-Letter/Random','Other Firms','HR/Recruiting/MBA']
print(pred, labels[np.argmax(pred)])
```

```
1/1 [==============================] - 0s 39ms/step
[[0.25816587 0.00338857 0.00290285 0.19505274 0.03923525 0.07588409
  0.02688199 0.3290266  0.03238907 0.01407197 0.02300098]] Buisness
```

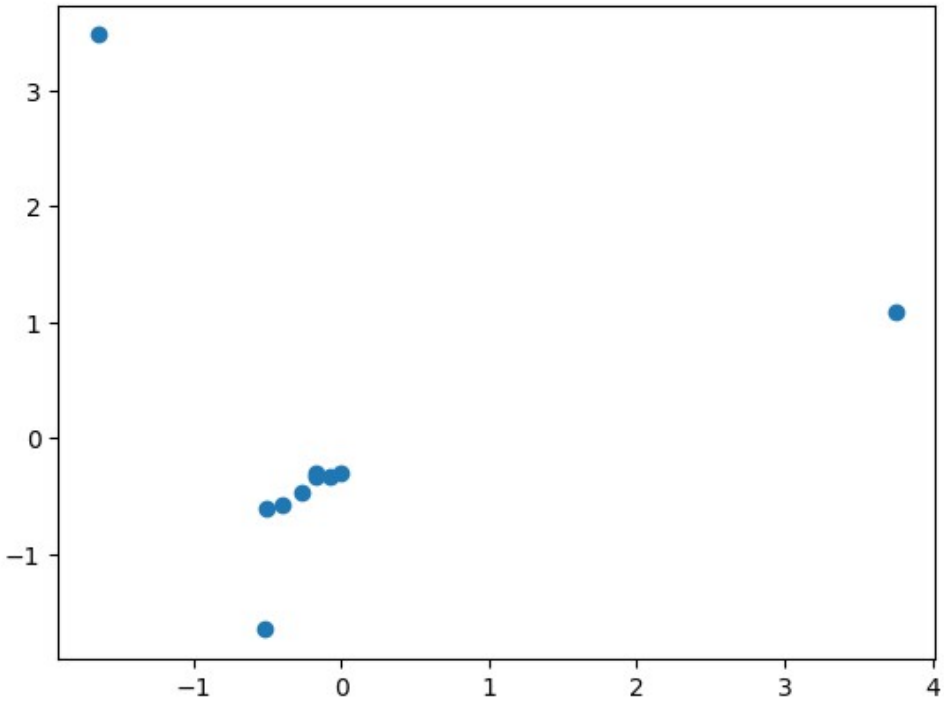
Shows us testing our model with a new complaint and the model outputting its classification.

CMPT 310
Milestone 2
Albert Hong



Our model accuracy during testing and training. Showcasing that our model does work and is “reliable”.

CMPT 310
Milestone 2
Albert Hong



This is a plot for our KNN label clustering, which shows our final label categories. Reducing the labels from 1000+ down to 11.

Challenges or Roadblocks: One of the hardest part was learning how to prepare the dataset into something we can use and train. As mentioned, the original dataset contained more than 1000 unique folder names, which would be impossible to use for a classification problem, and so we had to use a KNN clustering approach using Word2Vec embeddings to reduce the labels down to 11. In addition, the raw enron emails contained inconsistent formatting where there were missing headers, and privacy sanitized contents that required extensive preprocessing. There were also problems were emails had invalid or placeholder email addresses which also required to be handled.

As shown above, the final LSTM model showed a significant performance gap between training accuracy and validation accuracy, which indicates overfitting. We believe that the model may be too complex for the available data which will require additional regularization or other data augmentation techniques.

CMPT 310
Milestone 2
Albert Hong

Changes from Original Plan: Originally, we planned the ability to import user email directly into our model for classification, however due to the nature of the project scope, we reduced it down to using to categorizing a new singular email. This was needed as we want to focus on usability over features.