Introduction to AI and its Applications in Business

Project Report

Sentiment Analysis of tweets towards US airlines

Dhruv Goyal (dg3278) **Sonal Sharma (ss13449)**

For this project, Deep Learning models have been used to build a classification model using PyTorch library to predict the sentiment of consumers towards US airlines based on their reviews expressed in the form of tweets. We have created baseline models like Logistic Regression using sklearn and PyTorch and have compared them with state of the art Deep Learning methods like CNN, RNN and LSTMs.

1. Loading the Data

a. It is fetched in Google Collab directly using the google drive link of tweets.csv file.

2. Preprocessing Data

a. *Raw* tweets have been cleaned using python regex library.
b. The index for every word has been fetched from the provided dictionary.
c. Train/test split of data - 8:2
d. Schema of processed data (stored in df_seq)

```
df_seq.columns
```

```
Index(['sentiment', '1_word', '2_word', '3_word', '4_word', '5_word', '6_word',
       '7_word', '8_word', '9_word', '10_word', '11_word', '12_word',
       '13_word', '14_word', '15_word', '16_word', '17_word', '18_word',
       '19_word', '20_word', '21_word', '22_word', '23_word', '24_word',
       '25_word', '26_word', '27_word', '28_word', '29_word', '30_word'],
      dtype='object')
```

3. Exploratory Data Analysis

a. On exploring we found that there is a class imbalance where ~62% of data belongs to one class only. Based on this accuracy is not the appropriate score for evaluation, metrics like f1-score should be considered, but given the project guidelines we will go with accuracy.

	-1	0	1
Count	9058.000000	3080.000000	2355.000000
Percentage	0.624991	0.212516	0.162492

4. Models - We have used 5 different models to find the sentiments for the tweets.

Baseline models-

a. Logistic Regression using Scikit learn
b. Logistic Regression using PyTorch

Deep learning models-

a. FCN using PyTorch

This is the basic DL based model where we have implemented 2 Layered Neural Nets with ReLU non-linearity and added a softmax layer for final classification. The results are not as satisfying as expected since the model is possibly not able to understand the context of different tweets and is failing to classify the sentiment.

Hyperparameters: n_hidden = 100, epochs = 30
Model: 2 Fully connected layers with ReLu

b. CNN using PyTorch

This is the basic DL based model where we have implemented 1 Layered Neural Nets with ReLU nonlinearity with 1 dimension convolution and 1d max pooling. The results are moderately satisfying (at least more than FCN). The model is able to understand the context of different tweets moderately better.

Hyperparameters: n_iters = 3000, epochs = 50, batch_size = 100
Model: 1 Fully connected layers with 1D convolution with ReLu (lr_rate = 0.01)

c. RNN using PyTorch

Since this is a problem of Natural Language Processing for predicting the sentiment of the users based on the tweets about Airlines, Recurrents models are expected to perform better then CNNs, FCNs and other conventional methods. So we finally went ahead with RNN to check its performance which had satisfactory results.
Hyperparameters: n_hidden = 512, Epochs = 50
Model: Embedding layer, RNN Layer, Layer with Softmax.

d. LSTM using PyTorch

i. LSTMs(Long-Short term memory) is a sophisticated version of RNNs, which are architectured by combining several RNN models in such a way that they help in solving vanishing gradient problem prevalent in RNN, where plain vanilla RNNs cannot remember the context for a longer term. LSTMs are best suited for this sort of problems where we have to analyse the text and do some classification and they have been observed to give the best accuracy in our experiments.
ii. We have used dropout for LSTM which helps to overcome overfitting by randomly switching off some connections.
iii. We experimented with bi-directional LSTM, which trains two LSTMs on the input sequence and helps in preserving information from both past and present.

Hyperparameters: Embedding_dim: 100, Hidden_dim : 256, Dropout: 0.5, Bidirectional: True, Epochs: 50, N_layers: 2
Model: Embedding Layer, LSTM Layer, Dropout, Linear Decoder, Softmax for classification

5. Performance results

Python library	Model Name	Accuracy	Running time
Scikit learn	Logistic Regression	63.5%	0 sec
PyTorch	Logistic Regression	62.3%	4 sec
PyTorch	FC2- Layer	62.1%	8 Sec
PyTorch	CNN	64.3%	21 sec
PyTorch	RNN	65.5%	28 Sec
PyTorch	LSTM	70.95%	685 Sec

6. Conclusion

We were given a problem to find the sentiment of the users towards the airlines based on their tweets. This was a classification problem where we had to analyse the textual data in the form tweets and predict the sentiments which can be positive, neutral or negative. We have tried several models based on scikit-learn like Logistic Regression, then we went ahead with Deep Learning based baseline methods like FCNs and other state of the art methods like CNNs and RNNs, from our experiments we observed that RNN based recurrent methods performed well for our dataset, since this is a problem of analysing text sequence to perform a classification these methods are expected to perform better than CNNs or other DL based methods which are better suited for Computer Vision related tasks, RNNs can remember the context since they are recurrent and LSTMs based on RNN have performed well in our experiments since they have the capability to remember the context even longer.

We believe we have received satisfactory results in this analysis and can conclude that RNNs and specifically LSTMs are more suitable for these tasks since we achieved upto 70% accuracy on the test set.

The code is available at this link
https://colab.research.google.com/drive/1HfephJYXodROG7U42qAM0YLGSFDxQUCO?authuser=2#scrollTo=1JdKCArUJg-p