Sentiment Analysis of U.S. Airline Reviews
Multinomial Naive Bayes and LSTM Models

Yisong Cheng, Jiawen Gu

Khoury College of Computer Sciences

Northeastern University

San Jose, CA 95113

Cheng.yis@northeastern.edu

Gu.jiawen@northeastern.edu

April 13, 2024

Abstract

This project implements the Naive Bayes model and LSTM model with K-folds cross-validation on the "Twitter Airline Sentiment" dataset from Kaggle. Our Naive Bayes model is able to reach 76.21% accuracy, while the LSTM model can reach 77.19% accuracy. This shows the capability of these two models in sentiment analysis after being trained by the dataset without overfitting at the same time. K-fold cross-validation is also helpful in reducing variance and bias. It is expected that the performance of the two models will improve with more data.

Contents

1. Introduction 3

2. Method 5

2.1 Naive Bayesian Classifier 5

2.2 LSTM (Long Short-Term Memory) Networks 6

2.3 K-fold Cross-Validation 7

3. Results 8

3.1 Original Dataset Distribution 8

3.2 LSTM Model Performance 9

3.3 Multinomial Naïve Bayes Model 11

4. Discussion 13

5. Conclusion 14

6. References 16

1. Introduction

Social media has transformed the way people communicate, offering easily accessible platforms for feedback and comments across various aspects, including airline services. This Natural Language Processing project selects the comments on airlines in the United States from Twitter during February 2015 as a dataset, aiming to analyze the sentiments from text by categorizing it into positive, neutral and negative, which might be invaluable for both airlines to improve customer service and customers to make decisions.

Wide range of methods have been developed for sentimental analysis in the area of NLP, which can be primarily divided into supervised learning and unsupervised learning, with respective strengths and drawbacks.

As for the supervised learning, the models are trained on a labeled dataset where the sentiments (positive, negative and neutral) are already specified, including Linear Regression Models (logistic regression), Naive Bayes Classifier, Support Vector Machines (SVM), Decision Trees, Random Forests, Neural Network and Transformers. As for unsupervised learning, it attempts to derive sentiment from the intrinsic properties of texts through methods like Word Embeddings like Word2Vec, Clustering, latent Dirichlet Allocation (LDA) and Lexicon-Based Methods. While unsupervised learning is expert in discovering hidden patterns and handling unforeseen scenarios, supervised learning tends to display better accuracy and efficiency with ease of evaluation and applicability. The labeled dataset can make up for the weakness of supervised learning such as dependency on labeled data.

This project adopts the Naive Bayes Classifier and LSTM model along with K-fold Cross Validation to conduct sentiment analysis. K-fold Cross Validation, as a robust statistical technique, works well to assess the performance of machine learning models. It mitigates the risk of being dependent on how the data is split, coping with one of the disadvantages of supervised learning.

As for the expected outcome, we expect to accurately classify the sentiments of the tweets and identify patterns in customer feedback. The analysis will involve creating visualizations of sentiment distribution over time and among different airlines. These visualizations will highlight trends in customer perceptions and experiences, offering valuable insights into the effectiveness of current airline service strategies.

2. Method

2.1 Naive Bayesian Classifier

Naive Bayes classifiers is a probabilistic algorithm based on Bayes' Theorem, with the assumption of independence between every two features. It usually stands out for its success in text classification tasks because of its simplicity and speed while handling large datasets. For instance, research has explored using Naive Bayes for sentiment analysis on Twitter data, where the algorithm was applied to classify tweets into sentiment categories such as positive, negative, and neutral [1]. This method benefits from the simplicity and efficiency of Naive Bayes, making it ideal for processing large volumes of data typical of social media platforms (ar5iv). This project utilizes Naive Bayes to classify tweets into positive, negative, and neutral sentiments by calculating the probability that a text belongs to each sentiment category based on the presence of certain words in the tweets.

$$P(class\ c \mid word\ i) = P(word\ i \mid class\ c) \times \frac{P(class\ c)}{P(word\ i)}$$

The probability, that is, the posterior probability here, according to Bayes' theorem, is calculated by multiplying the prior probability of each class and the likelihood of having those certain words in this class. Then each tweet's sentiment is determined according to the highest probability. It excels facing large datasets and high-dimensional features.

2.2 LSTM (Long Short-Term Memory) Networks

Long Short-Term Memory (LSTM) model is one type of recurrent neural network (RNN) specifically designed to address the problem of learning long-term dependencies. While traditional RNNs have difficulties carrying information across many steps of sequence data, leading to problems like vanishing or exploding gradients, LSTM has a cell state to memorize previous inputs, thus capturing long term dependencies in sequence data, that is, text here.

Besides the cell state, the architecture of LSTM contains 3 gates: forget gate, input gate, and output gate, determining what information should be deleted from the cell state, what new information should be stored in the cell state and what the next hidden state should be.

$$\tilde{c_t} = tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$$
$$h_t = o_t * tanh(c^t)$$

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$i_t \rightarrow represents\ input\ gate.$
$f_t \rightarrow represents\ forget\ gate.$
$o_t \rightarrow represents\ output\ gate.$
$\sigma \rightarrow represents\ sigmoid\ function.$
$w_x \rightarrow weight\ for\ the\ respective\ gate(x)\ neurons.$
$h_{t-1} \rightarrow output\ of\ the\ previous\ lstm\ block(at\ timestamp\ t-1).$
$x_t \rightarrow input\ at\ current\ timestamp.$
$b_x \rightarrow biases\ for\ the\ respective\ gates(x).$

2.3 K-fold Cross-Validation

To evaluate and validate the predictive models rigorously, we employ the K-fold cross-validation. This technique enhances the reliability of the machine learning models by reducing variance and bias, thus avoiding overfitting or underfitting that can be seen frequently by a single train-test split.

Here we set k as 10, which means the dataset is split equally into 10 sets. For each round (there are 10 (k) rounds in total), 9 (k-1) out of them are set as the training set while the remaining 1 part is used as a testing set to evaluate model performances. This process is repeated 10 (k) times, which means each of the 10 sets will have been used exactly once as the testing set. This provides a good balance between computational efficiency and effectiveness of the model. The results shown by metrics like accuracy, precision, recall, and F1-score, are then averaged over all 10 (k) times to provide an objective performance estimate, in case that some training or testing set can perform abnormally.

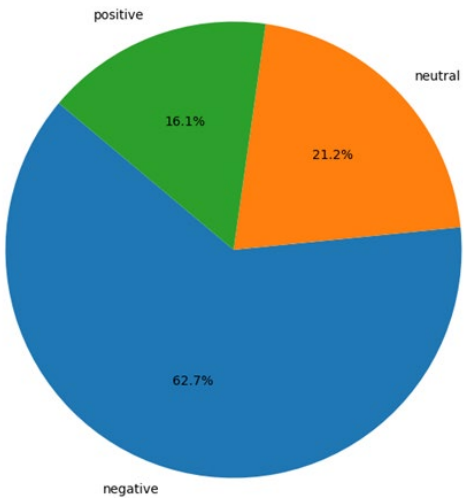
Combining these models and techniques, the study is expected to be both comprehensive and precise to determine sentiments in US airlines tweets during Feb 2015 and to further detect a trend about it if it can be applied to later or prior datasets. It can not only help validate the robustness of these findings, but also explore the nuances of sentiment analysis in a real-world context.

3. Results

In this part, we compare two models used to judge if airline reviews on Twitter are positive, negative, or neutral. We used an LSTM model, which is good for understanding sequences of words, and a Naive Bayes model, which is straightforward but effective. We measured their accuracy, precision, recall, and F1-score to see how well they worked. The results are laid out in the following sections.

3.1 Original Dataset Distribution

The pie chart illustrates the distribution of sentiments in our dataset. Most tweets are negative (62.7%), while neutral and positive tweets account for 21.2% and 16.1% respectively. This indicates a tendency among passengers to share negative experiences over positive or neutral ones on social media.



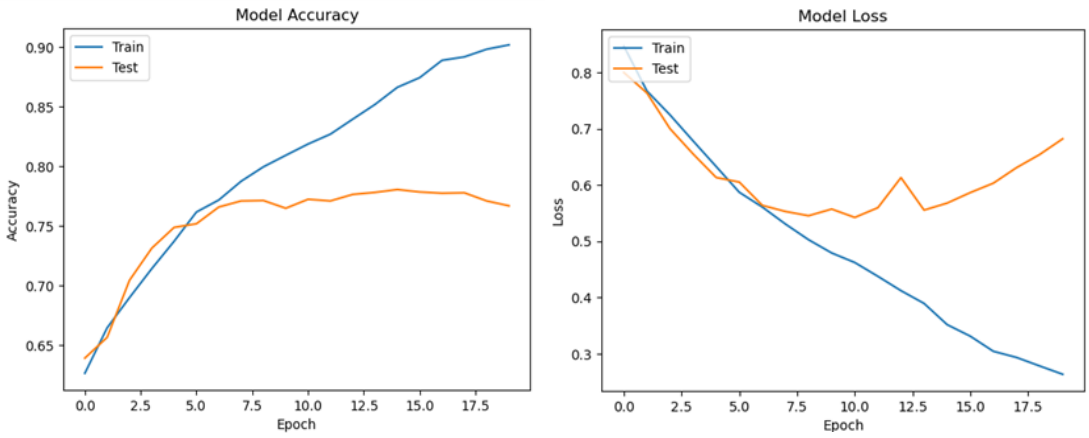
3.2 LSTM Model Performance

The overall accuracy of the LSTM model in classifying the sentiment of airline tweets is about 69.33%. The classification report details the model precision, recall and F1 score for each sentiment category. It performed best in recognizing negative tweets, with a precision of 0.72, a recall of 0.93, and an F1 score of 0.81. Neutral tweets were recognized with lower precision and recall, with an F1 score of 0.40, while positive tweets had a precision of 0.68, a recall of 0.27, and an F1 score of 0.38. These scores indicate that the model was most effective in detecting negative sentiment.

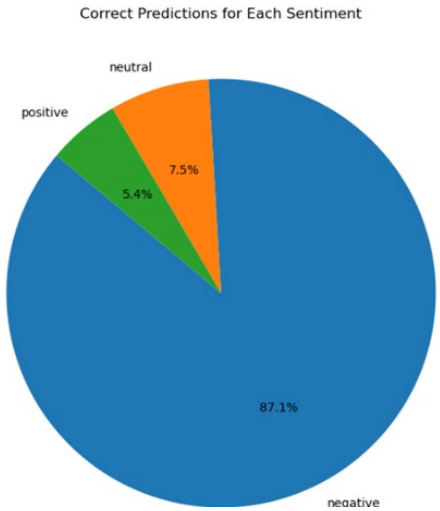
```
Accuracy for LSTM Model: 0.6933060109289617
Classification Report for LSTM Model:
              precision    recall  f1-score   support

    negative       0.72      0.93      0.81      1827
     neutral       0.55      0.32      0.40       603
    positive       0.68      0.27      0.38       498

    accuracy                           0.69      2928
   macro avg       0.65      0.51      0.53      2928
weighted avg       0.67      0.69      0.65      2928
```

The graphs tracking the model's learning history show a steady climb, with training accuracies as high as about 85%, as if the model had answered almost all the questions correctly by the end of the learning session. But when new, surprising questions come up (our test data), it manages to maintain about 75% accuracy, kind of like a good B student. Looking at how the model's uncertainty shrinks over time, we see that it's uncertain to begin with, with plenty of room for improvement. However, by the last round, the

uncertainty during training had been greatly reduced, even if we did face some ups and downs when testing new materials.



This pie-chart below shows the correct predictions for each sentiment among training of the LSTM model. It directly shows that the model is good at predicting negative sentiment more than neutral and positive. There are about 87.1% correct predictions are negative. This might be because the dataset has the most negative reviews (62.7%), so that the model is more familiar with negative sentiment analysis.



The data below shows how the LSTM model did across ten different folds, and we can see it's steady in figuring out the semantic in tweets. The accuracy didn't change much, staying around the 76-77% mark each time. The model was most accurate on the seventh try, with about 79.3%, and a bit less on the first, with around 75.68%. For how sure it was about its guesses, the 'loss' number stayed near 0.65 on average. This data tells us our model reliably understands tweets' sentiments across different data sets.

```
Score for fold 1: loss of 0.6362675428390503; accuracy of 75.68305730819702%
Score for fold 2: loss of 0.708564817905426; accuracy of 77.25409865379333%
Score for fold 3: loss of 0.6238672137260437; accuracy of 77.45901346206665%
Score for fold 4: loss of 0.6246695518493652; accuracy of 77.45901346206665%
Score for fold 5: loss of 0.665093719959259; accuracy of 77.59562730789185%
Score for fold 6: loss of 0.635239839553833; accuracy of 76.63934230804443%
Score for fold 7: loss of 0.6603451371192932; accuracy of 79.30327653884888%
Score for fold 8: loss of 0.620983362197876; accuracy of 78.89344096183777%
Score for fold 9: loss of 0.6819719672203064; accuracy of 76.50273442268372%
Score for fold 10: loss of 0.678119957447052; accuracy of 76.70764923095703%
Average scores for all 10 folds:
> Accuracy: 77.34972536563873 (+- 1.0346497938893113)
> Loss: 0.6535123109817504
```

However, for the future improvements, Bi-LSTM might be a good choice for this case. According to the research by Kian Long Tan, Bi-LSTM model achieved an accuracy of 81.20% on the same dataset we have used [2].

3.3 Multinomial Naïve Bayes Model

We have used K-folds (10 splits) for the MNB model. The classification report for the last folds of our Multinomial Naive Bayes model test tells us that the model is really good at spotting the negative comments in tweets. It got it right 93% of the time. It was pretty good with the positive tweets too, getting them right 61% of the time. When the model

guessed a tweet was negative, it was correct 78% of the time, and for the positive ones, it was right about 73% of the time.

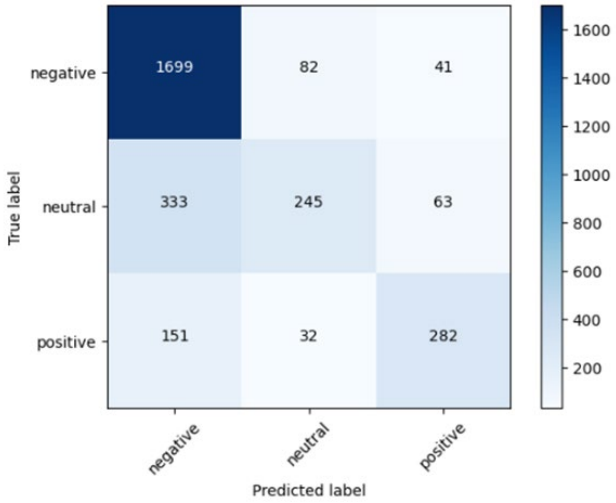
Fold 5

	precision	recall	f1-score	support
negative	0.78	0.93	0.85	1822
neutral	0.68	0.38	0.49	641
positive	0.73	0.61	0.66	465
accuracy			0.76	2928
macro avg	0.73	0.64	0.67	2928
weighted avg	0.75	0.76	0.74	2928

On average, the MNB model predicts the right semantic in the tweets 76% of the time. The numbers also say that the balance between the model's right guesses and being sure about those guesses (precision and F1 score) stayed pretty much the same, around 75%.

```
Average Accuracy: 0.7665300546448088
Average Precision: 0.7546764074206354
Average Recall: 0.7665300546448088
Average F1 Score: 0.7497355696495958
```

The confusion matrix, which is like a detailed scorecard, helps us see exactly where the model made the right calls and where it mixed things up. It shows that a lot of tweets that were actually neutral got labeled as negative. This happened 333 times. And some positive tweets were also mistaken for negative ones 151 times. But overall, this model did a solid job, especially when it came to the negative and positive tweets.



4. Discussion

Our study looked closely at two ways of determining whether tweets about airlines were positive, negative, or neutral: an LSTM network and a more straightforward Multinomial Naive Bayes classifier. We found that both methods did a pretty good job. The LSTM was particularly sharp when it came to catching the grumbles and gripes in tweets, but that could be because there were more of those kinds of tweets to learn from.

Even though it is more straightforward, the Naive Bayes held its own well, no matter how we mixed up the tweets it was looking at, proving it's a sturdy option for sorting tweets. Positive and neutral tweets were trickier for both methods, which makes sense since there weren't as many of those in the mix.

We did not just say they were good because we thought so; we looked at the numbers—how accurate they were, how often they got it right when they said a tweet was positive and negative, and how frequently they nailed it. The LSTM stayed reliable, hovering around a 76-77% right-answer rate, which is decent. Moreover, while it could do with a

better handle on the positive tweets, we did not see it stumbling too much when tested, so we are confident it is learning the right lessons.

Our methods are learning as much as they can from what tweets they have seen. There's always more to learn, so next up, we're thinking about throwing a wider net—more tweets, different kinds, to see if we can get these numbers even higher.

5. Conclusion

Digging into how the LSTM and Naive Bayes models figure out what people are feeling in their tweets about airlines has been quite eye-opening. The LSTM was on point in spotting the negative tweets, which could be handy for businesses trying to pick up on less-than-pleased customers. There is room to get better at recognizing when people are just okay or happy with their flights and getting that right could round out the picture for customer service folks. The Naive Bayes method stood out for its solid performance, no matter how we tossed the data around, showing that it is a straightforward and dependable choice for sifting through feedback. Looking down the road, we are thinking of mixing in a more even spread of tweets to keep any biases in check and to throw in more detail that could make our sentiment-spotting sharper. Taking this beyond our current project, these tuned-up models might be just the thing for businesses to keep an ear to the ground and quickly pick up on customer vibes, helping them jump on the positive, negative, and neutral in a snap. Plus, this is not just about airlines—there is a

chance here to tweak our approach to help in all sorts of customer service spots, making sense of what customers say in real-time.

References

[1] V. A. Kharde and S. A. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques," in International Journal of Computer Applications, vol. 139, no. 11, pp. 5-15, April 2016. https://doi.org/10.48550/arXiv.1601.06971

[2] K.L. Tan, C. P. Lee, K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," in Appl. Sci. 2023, 13(7), 4550 https://doi.org/10.3390/app13074550