NLP220 HW4

Yuchia Chang
ychan137@ucsc.edu

Abstract

The goal of this project is to experiment various combinations of features and models on article labeling task. The experiment results from consists of five models and four features.

1 Introduction

The report is **Twitter US Airline Sentiment** analysis. We are tasked to perform various methods of data processing on the texts and analyze the results difference. We will introduce the data distribution, data processing methods, and results from the experiments.

1.1 Dataset

The dataset contains tweets related to six major airlines and their sentiment labels. The data contains 15 columns/features, including, airline_sentiment, text, negativereason, airline, and so on. Key features are:

- **airline_sentiment**: Sentiment label (positive, neutral, or negative).
- **text**: The tweet text.

For more details, the dataset can be accessed at https://www.kaggle.com/crowdflower/twitter-airline-sentiment.

2 Part 1: Data Analysis

2.1 Overview of the Dataset

- Total number of samples: 14872.
- Distribution of **airline_sentiment**:
  - Positive: 16.14%.
  - Neutral: 21.17%.
  - Negative: 62.69%.
- Distribution of **negativereason**: Include top reasons with their frequencies.

Table 1 shows the data distribution among the six airlines. American airline has the most data and Virgin America has the least amount of data.

Airline	Tweet Count
American	2759
Delta	2222
Southwest	2420
US Airways	2913
United	3822
Virgin America	504

Table 1: Tweet Counts by Airline

Column **airline sentiment** has three unique values: "positive", "negative", and "neutral," and the most common is "negative". Column **negative reason** has ten unique values, and the most common is "Customer Service Issue." **Table 2** shows the distribution of the two values among the airlines.

Metric	Airline Sentiment	Negative Reason
Unique Count	3	10
Most Frequent	Negative	Customer Service Issue
Virgin America	181	60
United	2633	681
Southwest	1186	391
Delta	955	269
US Airways	2263	811
American	1960	768

Table 2: Summary of Airline Sentiment and Negative Reason Metrics

The longest and shortest tweet lengths are shown in **Table 3**. We observed that United Airlines has the shortest tweet and US Airways the longest. However, there are no significant differences in the tweet length analysis.

The histograms of the tweet length distribution are shown in **figures 1 2 3 4 5 6**

Airline	Shortest	Longest
Virgin America	22	159
United	12	165
Southwest	18	165
Delta	13	167
US Airways	15	186
American	17	167

Table 3: Shortest and Longest Tweet Lengths by Airline

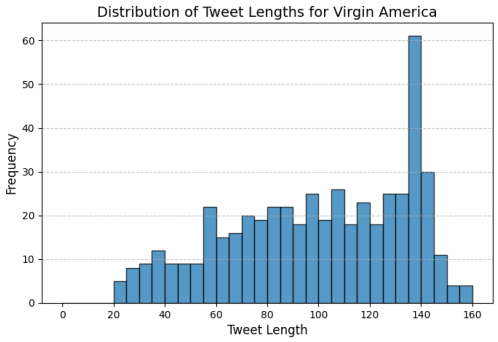

Figure 1: Tweet Length Distribution (Virgin America)

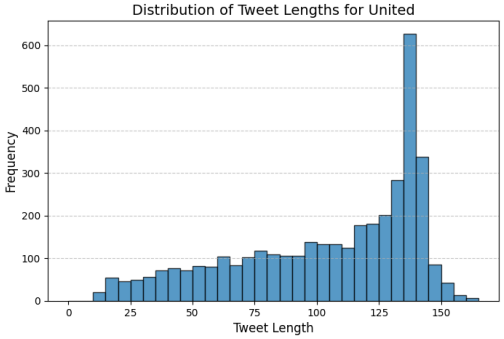

Figure 2: Tweet Length Distribution (United)

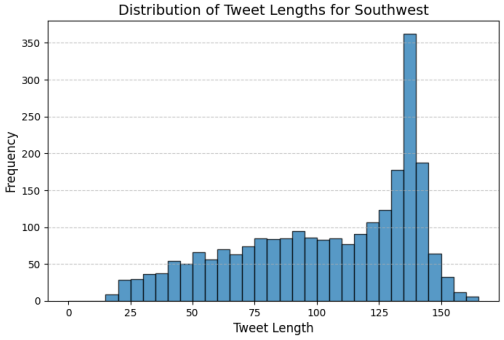

Figure 3: Tweet Length Distribution (Southwest)

The sentiment distribution of all airlines can be visualized in **figure 7**, where we can see that all airlines receive more negative sentiment tweets

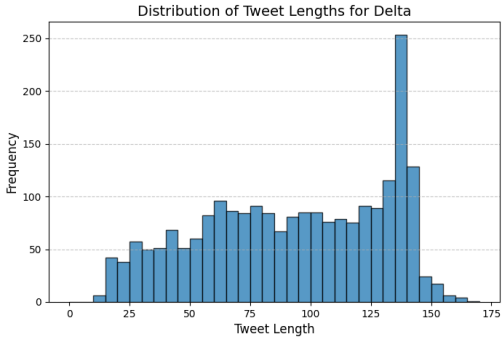

Figure 4: Tweet Length Distribution (Delta)

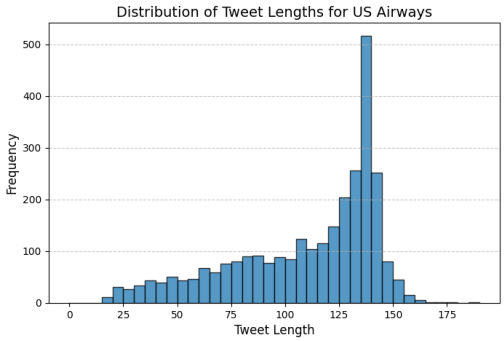

Figure 5: Tweet Length Distribution (US Airways)

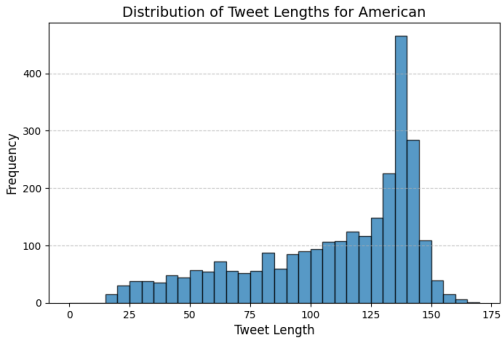

Figure 6: Tweet Length Distribution (American)

than positive and neutral ones. Virgin America receives the smallest percentage of negative sentiment tweets, and US Airways receives the largest percentage of negative sentiment tweets.

2.2 Customized Tokenizer

The tokenizer I built has the features: lowercase, splitting by punctuation characters, removing double white spaces, splitting contractions, parsing URLs or emails, and splitting mentions and hashtags. To evluate my own tokenizer, I compared the outputs of my tokenizer and of `NLTK.word_tokenize` method. I randomly chose

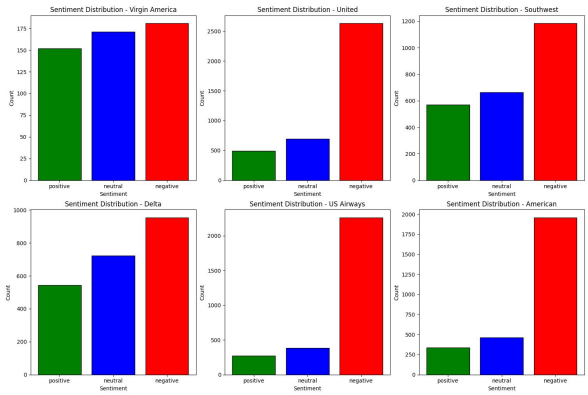

Figure 7: Sentiment Distribution

five text from the dataset, and analyzed their difference. I noticed the most different features are uppercase lowercase handling. NLTK does not convert tokens to lowercase by default. The other difference is NLTK does not split tokens by contractions.

3 Part 2: Data Cleaning

3.1 Cleaning Steps

The following cleaning actions were applied to preprocess the tweets:

1. Remove mentions (e.g., @username).
2. Remove currency.
3. Remove emails.
4. Remove emojis.
5. Replace HTML escaped characters (e.g., `<` $\rightarrow$ "<").
6. Normalize punctuation (e.g., "!!!!" $\rightarrow$ "!").
7. Normalize dates and times (e.g., "2/24" $\rightarrow$ <DATE>).
8. Remove URLs.
9. Remove hashtags while keeping the text.
10. Remove non-alphanumeric characters.
11. Replace multiple spaces with a single space.
12. Perform lemmatization.

3.2 Challenges and Assumptions

We chose the features in the data cleaning steps because we anticipate the texts can generate noise and does not provide useful information if unhandled.

- Mentions were removed as they do not contribute semantic information to the sentiment analysis task.
- Currency symbols and amounts were removed as they are irrelevant unless analyzing financial data.
- Email addresses were removed as they do not provide useful information for sentiment classification.
- Emojis were removed as their contribution to textual sentiment was considered negligible for this task.
- HTML escaped characters were replaced to make the text human-readable and semantically meaningful.
- Excessive punctuation was normalized to ensure consistency and reduce unnecessary variability.
- Dates and times were normalized to placeholders to reduce variability in the dataset.
- URLs were removed since they typically redirect to external resources with no impact on sentiment.
- Hashtags were retained as words while removing the # symbol to preserve meaningful content.
- Non-alphanumeric characters were removed to simplify text and reduce noise.
- Multiple spaces were normalized to ensure consistent tokenization and cleaner text formatting.
- Lemmatization was applied to reduce words to their base forms for consistency and vocabulary reduction.

We applied each feature to the dataset individually and removed duplicate rows to see how affective our features are. The **Table 4** shows most features can reduce around 200 tweets in dataset.

Feature	Affected Rows
1	266
2	188
3	188
4	188
5	188
6	195
7	188
8	213
9	188
10	223
11	190
12	190
All	**382**

Table 4: Affected Rows per Feature

4 Part 3: Model Training and Evaluation

For the sentiment analysis experiment, we select `SGDClassifier` from scikit-learn for the analysis. With the model hyperparameter fixed to the assigned values, we can eliminate the performance difference due to hyperparameter setting and focus on text processing strategies. The dataset is split into train and test set with ration 9:1. After the split, we use TFIDF vectorizer for token embedding, lowering the influence of frequently occurred terms.

4.1 Feature Order

From the twelve features listed in the previous sections. We selected ten orders for the text processing techniques to evaluate whether the orders can be a factor on model performance. **Table 5** shows the orders we tested.

#	Feature Order
1	1, 8, 4, 9, 3, 2, 7, 5, 6, 10, 11, 12
2	1, 8, 4, 3, 9, 2, 7, 10, 6, 5, 11, 12
3	7, 6, 5, 8, 1, 4, 3, 9, 2, 10, 11, 12
4	9, 6, 1, 8, 4, 3, 2, 7, 5, 10, 11, 12
5	10, 11, 1, 8, 4, 3, 9, 2, 7, 6, 5, 12
6	8, 1, 3, 9, 7, 6, 5, 10, 11, 4, 12, 2
7	1, 8, 4, 9, 10, 11, 6, 7, 5, 3, 2, 12
8	1, 6, 8, 7, 4, 5, 3, 10, 11, 2, 9, 12
9	6, 7, 5, 10, 11, 12, 4, 2, 3, 9, 1, 8
10	1, 8, 4, 3, 2, 9, 10, 7, 6, 5, 11, 12

Table 5: The numbers in feature order column are features listed in Part2 in original order. Texts are processed with the order from left to right.

We performed cross validation accuracy on train set with the ten feature orders and no features 20 times, and the result shown in **Table 6** suggests the accuracy score are in range from 0.79 to 0.82, indicating the range of accuracy score for actual test result. Some features have higher maximum accuracy or lower minimum accuracy, but regardless what order we choose, the accuracy seems to be stable in the range. However, without implementing any text processing does show the worst accuracy in average.

#	High Accuracy	Low Accuracy
6	0.8207	0.7905
8	0.8207	0.7905
2	0.8184	0.7903
4	0.8184	0.7905
7	0.8184	0.7913
1	0.8176	0.7905
3	0.8176	0.7905
9	0.8173	0.7913
10	0.8161	0.7897
5	0.8157	0.7926
No Feature	0.814	0.7892

Table 6: Cross Validation Accuracy Score

When we ran the ten feature orders with test set, we see order #7 has the highest accuracy score shown in **Table 7**. The score is also within the range that we observed from cross validation accuray. For the limited space in the report, we want to present more detailed evaluation metrics only on the feature order #7.

#	Feature Order
1	0.8163
2	0.8142
3	0.8170
4	0.8163
5	0.8001
6	0.8170
7	0.8177
8	0.8170
9	0.7994
10	0.8170

Table 7: Feature Orders Accuracy Score

4.2 Result

Table 8 is the classification report for each sentiment class. We can see negative sentiment has

significant higher f1-score than neutral and positive sentiments. A possible reason is negative sentiment dataset simply outnumber the other two classes, yielding higher scores in the metric. A interesting finding is even though neutral and positive tweets have similar dataset size, it is much more difficult to correctly predict neutral sentiment than to positive. Our reasoning for the phenomenon is that positive sentiment tweets usually contain some strong indicators, such as "amazing", "worth it!", or "best".

Label	Precision	Recall	F1-Score
Negative	0.8365	0.9552	0.8919
Neutral	0.7243	0.4702	0.5702
Positive	0.8051	0.6978	0.7476
Accuracy	0.8177	0.8177	0.8177
Macro Avg	0.7887	0.7077	0.7366
Weighted Avg	0.8091	0.8177	0.8049

Table 8: Feature Classification Report Metrics

From the confusion matrix (**Figure 8**) clearly shows negative sentiment dominance in label classes, supporting our reasoning for its high f1-score.

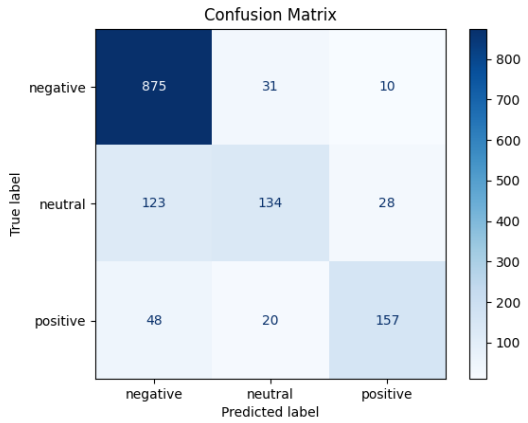

Figure 8: Confusion Matrix

We also checked classification reports for other feature orders, and we confirm that order #7 has the highest macro average f1-score than others. However, if we look at the classification report (**Figure 9**) with none of the twelve features selected, we can see the macro average f1-score is actually higher. The matrix shows lower score for negative recall but significantly higher score for neutral recall. Other scores are in similar levels. The result suggests our features possibly eliminate neutral tokens such as non-alphanumeric tokens or date times, which in the meantime, increases negative precision and recall scores even more.

Label	Precision	Recall	F1-Score
Negative	0.8371	0.9346	0.8832
Neutral	0.7235	0.5304	0.6121
Positive	0.7941	0.6983	0.7431
Accuracy	0.8140	0.8140	0.8140
Macro Avg	0.7849	0.7211	0.7461
Weighted Avg	0.8069	0.8140	0.8052

Table 9: No Feature Classification Report Metrics

5 Conclusion

Through the experiment, we show different text processing strategies sometimes have unexpected impact to mode predictions. A strategy choice may depend on a particular goal. If the task in binary classification problem, eliminating neutral terms can be a good strategy, and vice versa. Sentiment analysis with SGD classifier is also a good setup to evaluate our strategy, since the model has fast training time to quickly verify the strategy.