Sentiment Analysis Annotation

Raghav Upadhyay
raghavupadhyay@arizona.edu

Abstract

This project explores sentiment analysis of customer reviews for refurbished mobile phones on Amazon. A manually annotated dataset was created by two annotators, achieving a high agreement score (Cohen's Kappa: 0.91). The reviews were categorized as positive or negative, and disagreements were resolved to form a reliable golden label. The data underwent preprocessing, including text cleaning and TF-IDF vectorization, to ensure readiness for training a neural network. The model achieved an accuracy of 80% and demonstrated balanced performance across sentiment classes, as highlighted by key metrics and visualizations. This study underscores the value of sentiment analysis in understanding customer feedback, aiding businesses in improving decision-making and customer satisfaction.

1 Introduction

This project focuses on sentiment analysis of Amazon reviews for refurbished mobile phones, classifying them as positive or negative. Two annotators labeled the reviews, achieving a high agreement score (Cohen's Kappa: 0.91).

The data was cleaned and processed using text cleaning and TF-IDF vectorization, and a neural network model was used for classification. Visualizations of sentiment distribution and annotator agreement provided key insights.

This work demonstrates how clean data and effective models enable accurate sentiment analysis, helping businesses better understand customer feedback and make informed decisions.

2 Dataset Overview

The dataset for this project was manually created by collecting reviews from the Amazon review section of a refurbished phone seller. The reviews capture diverse customer experiences, ranging from product performance and quality to delivery satisfaction. To ensure consistency, the dataset underwent a dual-annotation process:

- **Annotator 1:** The primary annotator labeled reviews as positive or negative, aiming for a balanced sentiment distribution.
- **Annotator 2:** The secondary annotator independently reviewed and labeled the same dataset, introducing variability in perspectives.

Discrepancies between the annotators were resolved to create the **Golden Label**, representing the final agreed-upon sentiment for each review. The dataset comprises a near-equal distribution of positive and negative sentiments, ensuring fairness in model training and evaluation.

3 Methodology

3.1 Annotation Guidelines

The annotation process followed clear rules to ensure consistent and accurate sentiment labeling. Reviews were categorized into two sentiment classes:

1. **Positive:**
    - Assigned to reviews expressing satisfaction, appreciation, or positive experiences.
    - Indicators included:
        - Praise (e.g., "great," "excellent").
        - Satisfaction with product performance, appearance, or service.
        - Positive emotions such as gratitude or delight.
    - Examples:
        - "The phone works perfectly and looks brand new!"
        - "Great value for the price. I'm very happy with the purchase."

2. **Negative:**
    - Assigned to reviews expressing dissatisfaction, frustration, or negative experiences.
    - Indicators included:
        - Criticism or complaints (e.g., “terrible,” “waste of money”).
        - Issues with product quality or delivery.
        - Expressions of frustration or disappointment.
    - Examples:
        - “The phone stopped working after one week. Total waste of money.”
        - “There were scratches on the screen, and it looks used, not refurbished.”

3.2 Annotation Process

The annotation involved the following steps:

1. **Review Content:**
    - Each review was read carefully to understand the sentiment expressed.
2. **Identify Sentiment:**
    - Sentiments were categorized based on the guidelines, looking for key indicators and emotional cues in the text.
3. **Label the Sentiment:**
    - Reviews were labeled as either Positive or Negative.
4. **Document Annotations:**
    - To maintain consistency, annotations were recorded clearly, referring back to the examples and definitions if uncertainty arose.

Two individuals performed annotations:

- **Annotator 1:** Primary annotator who labeled all reviews (Raghav Upadhyay).
- **Annotator 2:** Reviewed the same dataset to provide a secondary opinion (Sriram Manikyala).
- **Golden Label:** Final agreement between the two annotators, ensuring reliability in sentiment classification.

3.3 Data Visualization

The annotated dataset was visualized to analyze and validate the quality of annotations. Table 1 provides an overview of the annotation counts by Annotator 1, Annotator 2, and the Golden Label.

Table 1: Counts of Annotations for Annotators and Golden Label

Annotator	Positive	Negative
Annotator 1	125	125
Annotator 2	126	124
Golden Label	128	122

The following visualizations provide insights into sentiment distributions and annotator agreement:

- **Sentiment Distribution for Annotator 1:**
    - The pie chart shows an even split between Positive and Negative labels, with each accounting for 50% of the dataset.
    - This balance indicates that Annotator 1 ensured equal representation of both sentiment classes during annotation.

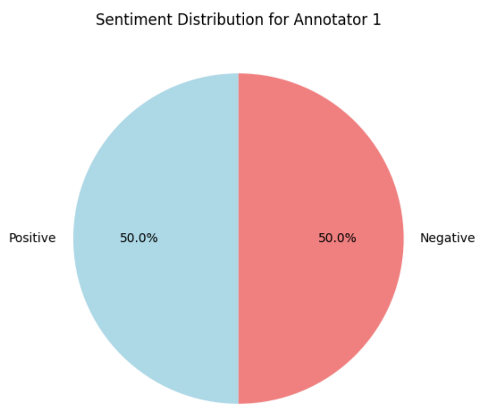

Figure 1: Sentiment Distribution for Annotator 1.

- **Sentiment Distribution for Annotator 2:**
    - The pie chart reflects Annotator 2’s annotations, with 53% Negative labels and 47% Positive labels.
    - This slight imbalance shows a minor difference in sentiment perception between Annotator 1 and Annotator 2.

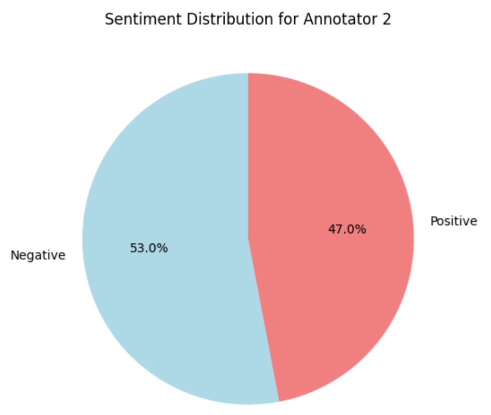

Figure 2: Sentiment Distribution for Annotator 2.

- **Agreement Between Annotator 1 and Annotator 2:**
  - This pie chart highlights that the two annotators agreed on 94% of the labels, with only 6% disagreement.
  - The high agreement percentage indicates strong consistency between Annotators 1 and 2, ensuring reliable annotations for the dataset.

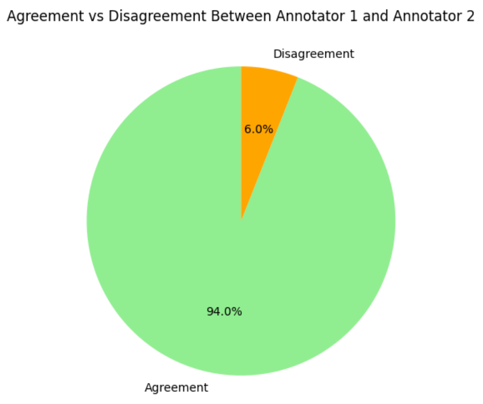

Figure 3: Agreement and Disagreement Between Annotator 1 and Annotator 2.

- **Sentiment Distribution for Golden Label:**
  - The pie chart represents the final agreed-upon labels (Golden Label), derived from resolving disagreements between the two annotators.
  - he split shows 50.5% Positive and 49.5% Negative labels, maintaining a balanced dataset for sentiment classification.

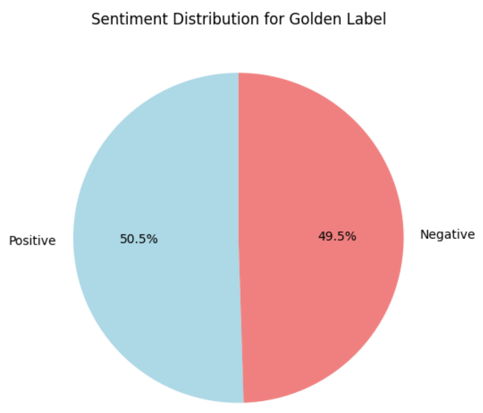

Figure 4: Sentiment Distribution for the Golden Label.

3.4 Data Pre-Processing

Data pre-processing was a critical step in preparing the annotated dataset for training the sentiment analysis model. The steps included:

- **Data Cleaning:**
  - Removed unnecessary characters, extra spaces, and punctuation to standardize the text.
  - Converted all text to lowercase to ensure consistency during model training.
- **Tokenization:**
  - Split the reviews into individual words (tokens) for easier analysis and processing.
- **Stopword Removal:**
  - Common words like "and," "the," and "is," which do not contribute to sentiment classification, were removed to reduce noise.
- **Vectorization:**
  - The cleaned text data was transformed into numerical representations using the **TF-IDF (Term Frequency-Inverse Document Frequency)** technique. This step helped capture the importance of words in each review relative to the entire dataset.
- **Dataset Splitting:**
  - The dataset was split into **training (80%)** and **testing (20%)** sets to train and evaluate the model's performance effectively.

This pre-processing pipeline ensured the dataset was clean, standardized, and ready for machine learning, ultimately contributing to the model's robust performance.

3.5 Model Training

The sentiment classification model was trained using a neural network built with the `Sequential` API from TensorFlow. The training process involved the following steps:

- **Model Architecture:**
  - The neural network was constructed with the following layers:
    - **Input Layer:** A dense layer with 128 neurons, using ReLU activation, to process the TF-IDF features.
    - **Hidden Layer:** A dense layer with 64 neurons, also using ReLU activation, for feature transformation and representation learning.
    - **Dropout Regularization:** Dropout layers with a rate of 0.3 were applied after each dense layer to reduce overfitting.
    - **Output Layer:** A dense layer with 1 neuron and sigmoid activation to perform binary sentiment classification.
- **Training Configuration:**
  - The model was compiled with the **binary cross-entropy loss function**, which is well-suited for binary classification tasks.
  - The **Adam optimizer** was used for efficient optimization of the network's weights.
  - The model was trained over **20 epochs** with a batch size of **32**.
- **Training Process:**
  - The dataset was divided into **80% training** and **20% testing** subsets to facilitate training and evaluation.
  - During training, both **training accuracy** and **validation accuracy** were monitored to ensure the model was learning effectively.
  - The training process included dropout regularization to minimize overfitting and improve generalization.

The use of the `Sequential` API allowed for a straightforward implementation of the neural network, leveraging dropout regularization and efficient optimization to build a robust sentiment classification model.

4 Results

4.1 Annotator Agreement

- Annotator 1 and Annotator 2 demonstrated a **94% agreement rate**, with only **6% disagreement**.

- The **Cohen's Kappa score** was calculated to be **0.88**, indicating an almost perfect level of inter-annotator reliability.

4.2 Evaluation Metrics

The Evaluation Metrics are summarized in the table below:

Table 2: Evaluation Metrics

Metric	Score
Accuracy	80%
Precision	0.85
Recall	0.81
F1 Score	0.80

- **Accuracy:** The model achieved an accuracy of **80%**, indicating that the majority of predictions were correct.

- **Precision:** The macro-average precision was **0.85**, demonstrating the model's reliability in predicting both positive and negative sentiments.

- **Recall:** The macro-average recall of **0.81** indicates the model's ability to identify actual positives and negatives effectively.

- **F1 Score:** The macro-average F1 Score of **0.80** reflects a balance between precision and recall, highlighting the model's robustness in sentiment classification.

4.3 Training and Validation Metrics

The performance of the model during training was analyzed by plotting the loss and accuracy over epochs for both the training and validation datasets. The figures below illustrate the trends observed:

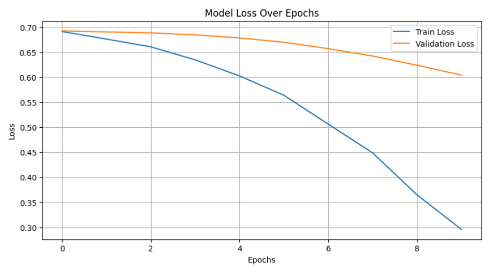

Figure 5: Model Loss Over Epochs.

- **Training Loss:** The training loss decreased steadily across epochs, indicating that the model was learning effectively.
- **Validation Loss:** The validation loss also showed a downward trend, suggesting that the model generalized well to unseen data without significant overfitting.

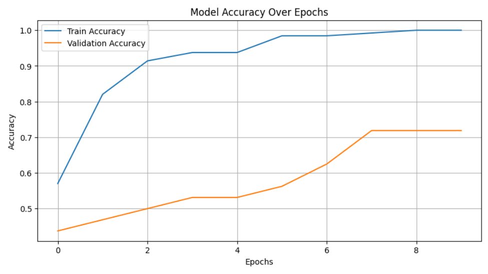

Figure 6: Model Accuracy Over Epochs.

- **Training Accuracy:** The training accuracy improved rapidly and approached 100%, demonstrating that the model effectively captured patterns in the training data.
- **Validation Accuracy:** The validation accuracy steadily increased, reaching around 80%, confirming the model's ability to classify unseen data accurately.

These visualizations highlight the model's consistent improvement during training, with validation trends indicating good generalization and minimal risk of overfitting.

5 Conclusion

This project demonstrated effective sentiment analysis on Amazon reviews for refurbished phones using a manually annotated dataset. High annotation consistency (Cohen's Kappa: 0.91) and balanced sentiment distribution ensured robust results.

A neural network model achieved 80% accuracy and an F1 score of 0.80, supported by comprehensive preprocessing and visualization. This study highlights the value of sentiment analysis in understanding customer feedback and offers a foundation for future improvements in classification performance.

6 References

1. Amazon. *Apple iPhone 15, 128GB, Blue - Amazon.com*. Retrieved from: Amazon.
2. Scikit-learn. *Cohen's Kappa Implementation*. Retrieved from: Scikit-learn Documentation.
3. Grammarly. *Online Writing Assistance Tool*. Retrieved from: Grammarly.