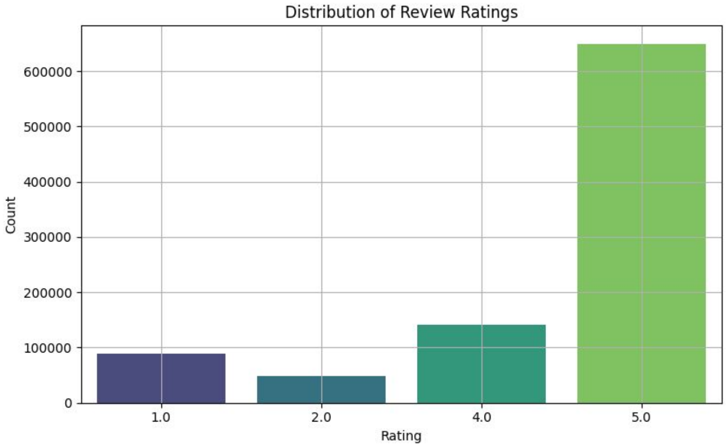
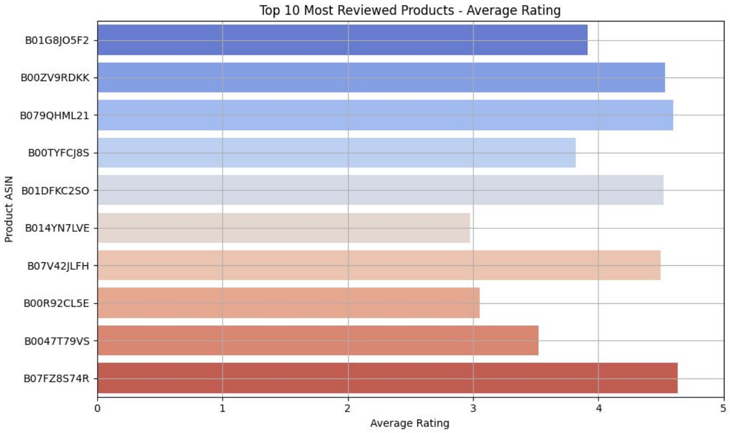
Evaluating NLP Techniques on Amazon Product Reviews

Vineela Goli, Gaurav Kothamachu Harish

Introduction

- Dataset
  - https://jmcauley.ucsd.edu/data/amazon/
  - Electronics Category Reviews
- Preprocessing Techniques
  - Converted all text to lowercase, removed stop words and applied lemmatization during tokenization
  - Sentiment Labeled: all reviews with rating >= 4 as positive, <=2 as negative and neutral otherwise.
- Models
  - Logistic Regression with TF-IDF
  - RNN with Word2Vec and LSTM


Distribution of Review Ratings


Top 10 Most Reviewed Products - Average Rating

Logistic Regression with TF-IDF Architecture

- TF-IDF Vectorization
  - Assigns higher weights to rare but meaningful terms
  - Less influence to common words (i.e. “the”, “is”)
  - Configurable parameters: max_features, ngram_range, use_idf, max_df
- Logistic Regression Classifier
  - Linear model with sigmoid activation to predict probability of positive sentiment
  - Decision threshold of 0.5 -positive or negative labeling
  - Trained with 5-fold stratified cross validation for robust results
- Model Calibration
  - Performed ablation study to determine the effects of various parameters
  - Used k-fold cross validation
  - Studied extreme errors
- Evaluation Metrics
  - Accuracy, F1 score

Logistic Regression - Ablation Study

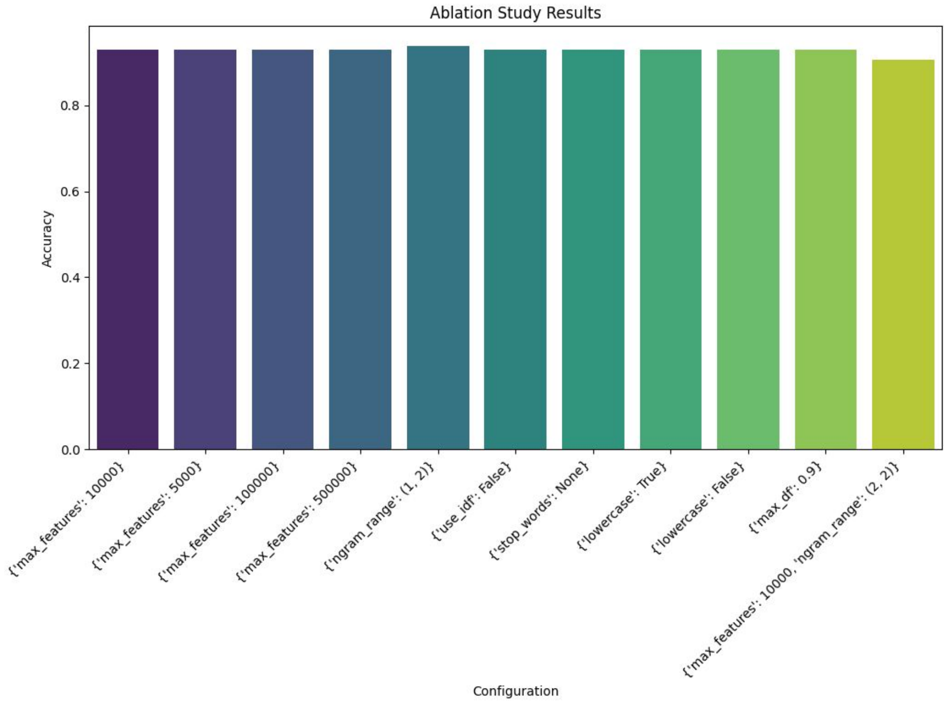
- All configurations performed fairly well (~90-93% accuracy)

- N-gram range (1,2) (unigram along with bigram) performed best of all configurations
  - Bigram may have better assisted in distinguishing phrases like “not good” “wasn’t great” better than unigram to get more contextual and sentimental meaning.

- Using only bigrams performed the worst

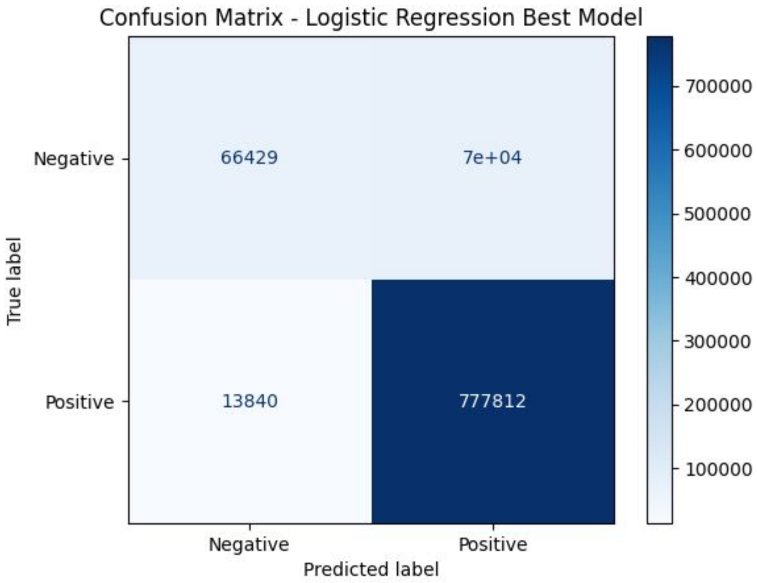
- Casing and change in max features had little impact

- Use_idf: false performed slightly worse
  - Suggesting giving less weight to the common words and more weight to the rare ones was beneficial



Logistic Regression with TF-IDF Results

- Extremely good at identifying positive reviews
- Moderate performance in identifying negative reviews (~70000 false positives)
- Could be due to the skewed dataset with more positive than negative reviews



Logistic Regression - Extreme Error Analysis

True: 0, Pred: 1, Confidence: 1.00

get reverse description packaging say gave lens friend stuck reverse lot size besides people saying work great sold

True: 0, Pred: 1, Confidence: 1.00

daylight work great nighttime workout rte horrible think night ir light bouncing waterproof plastic glass daylight work outstanding brought unit nighttime usage

True: 0, Pred: 1, Confidence: 1.00

work great

True: 0, Pred: 1, Confidence: 1.00

meh work great left streak chean lot get streak

True: 0, Pred: 1, Confidence: 1.00

work great stay using outlet

True: 0, Pred: 1, Confidence: 1.00

work great darn small

True: 0, Pred: 1, Confidence: 1.00

work great constantly dropping phone connecting general

True: 0, Pred: 1, Confidence: 1.00

product snapback customer unit via fixed tab hinged tab fixed tab thick ca enter notch gps delivered useless ca used deterred shaved thousandth inch tab work great

True: 0, Pred: 1, Confidence: 1.00

month receiving cord started charging intermittently cord anker work great

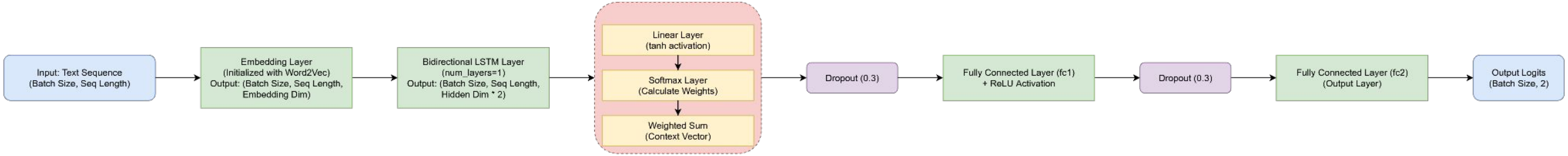
True: 0, Pred: 1, Confidence: 1.00

point mean work great point

RNN-LSTM Model for Sentiment Analysis

- Core Components: Word2Vec embeddings (custom-trained on dataset) + Bidirectional LSTM (processes sequences forward/backward) + Attention mechanism (focuses on key words)

- Why RNN-LSTM? Captures word order and context (e.g., "not great" vs. "great") – addresses TF-IDF limitations in sequential data

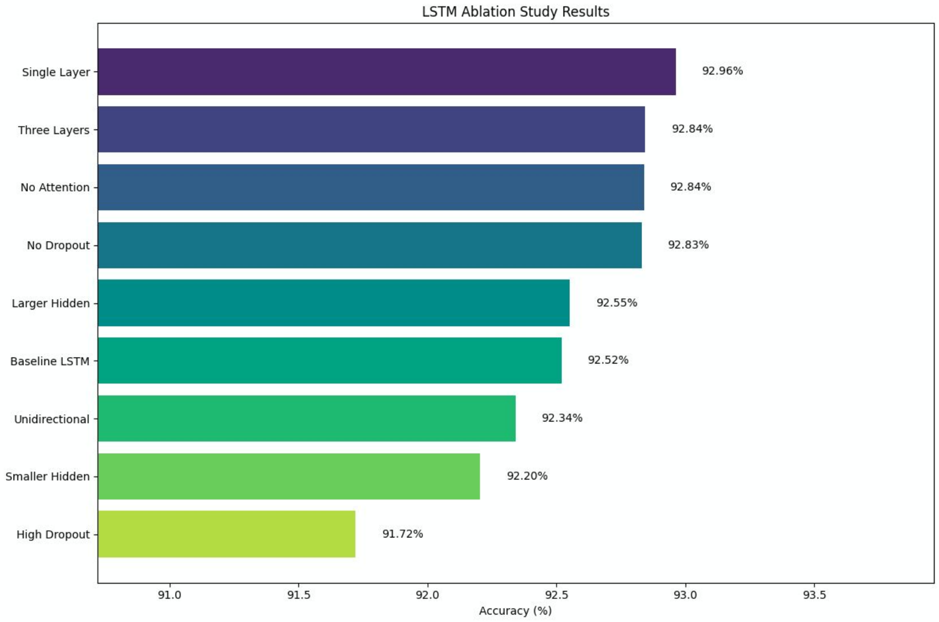


Model Specs: 128 hidden units, 1-3 layers tested, dropout 0.3, trained with Adam optimizer (lr=0.001) for 20 epochs

RNN-LSTM Performance and Ablation Insights

Best Model is the Single-layer Bi-LSTM + Attention with 93.91% accuracy, 0.1577 test loss

Class Performance
Positive (Precision: 0.9581, Recall: 0.9711, F1: 0.9645)
Negative (Precision: 0.8183, Recall: 0.7540, F1: 0.7848)



RNN - Extreme Error Analysis

All top errors are false positives

Model struggles with sarcasm and there is a lot of label noise in data

Example 1:

True: Negative, Predicted: Positive

Confidence: 1.000, Loss: 8.342

Text: well fantastic awesome...

Example 2:

True: Negative, Predicted: Positive

Confidence: 1.000, Loss: 8.047

Text: satisfied nice design work great microsd unexpected surprise actually boot lot tested hdmi output yet recommend...

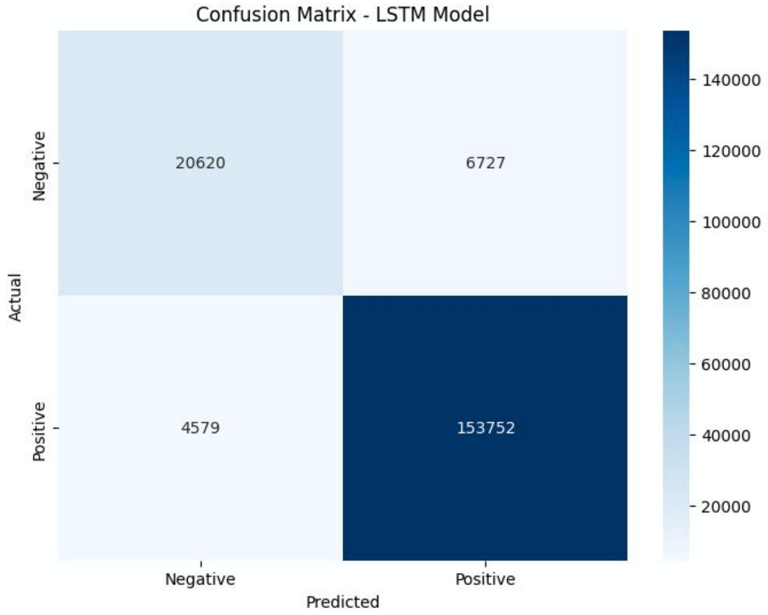
Model Comparison

TF-IDF + Logistic: Fast (93.74% acc), efficient for large data, but misses context (e.g., negation errors)

RNN-LSTM: Similar acc (93.91%), excels at sequences, but computationally heavier (longer training)

When to Choose: TF-IDF for quick analysis; LSTM for nuanced reviews

Overall: Both strong baselines; LSTM edges out on F1 for positives (0.9645 vs. 0.9640)



Conclusion

Both models perform very well on this dataset. The best choice depends on the specific needs of the application, balancing the need for speed and efficiency with the ability to understand complex language.

Future Work

Incorporate Trigrams for TF-IDF: To capture more contextual information.

Clean the Data Further: To handle potentially mislabeled reviews.

Explore More Advanced Models: Such as transformers like BERT, which could lead to even better performance