Evaluating NLP Techniques on Amazon Product Reviews

Vineela Goli (002936526), Gaurav Kothamachu Harish (002301064)

Abstract— **This project evaluates and compares two Natural Language Processing techniques for sentiment analysis on the Amazon product reviews (electronics category) with Logistic Regression with TF-IDF and Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). The dataset has been preprocessed by text cleaning, tokenization, stopword removal and lemmatization. For the first approach, TF-IDF vectorization transformed the reviews into weighted numerical features, which were classified using Logistic Regression. An ablation study was conducted to evaluate the effect of different configurations, with the best result achieved using both unigrams and bigrams yielding an accuracy of 93.75% and an F1 score of 0.9640. The second approach utilized word embeddings with an LSTM-based RNN to model sequential dependencies and capture deeper contextual patterns in the reviews, achieving an accuracy of 93.91% and an F1 score of 0.9645. Ablation studies revealed that architectural complexity in LSTM models provided marginal gains over simpler configurations, while TF-IDF benefited significantly from n-gram combinations. The comparative analysis demonstrates that while both approaches achieve similar accuracy, they offer distinct trade-offs: Logistic Regression provides computational efficiency and interpretability, while LSTM offers enhanced contextual understanding at higher computational cost.**

Github— https://github.com/khgaurav/CS6120FinalProject

Data— https://northeastern-my.sharepoint.com/:f:/g/personal/kothamachuharish_g_northeastern_edu/EiwO8Q42NbpBjCo4-Wce0fEBZxF3fYXK6SB-ffA628aUJg?e=Utl63y

I. Introduction

The proliferation of e-commerce has fundamentally reshaped global commerce, creating a digital marketplace where consumer feedback, in the form of user-generated reviews, has become a cornerstone of the decision-making process. This shift is underscored by market research indicating that 95% of the consumers read online reviews before committing to a purchase. These reviews contain invaluable insights into consumer preferences, product strengths, and areas for improvement.

Sentiment analysis processes textual opinions to classify them as positive, negative or neutral, transforming unstructured data into actionable insights. The foundational work using SVM[1] on movie reviews achieves 72.8%-82.9% accuracy. The progression from traditional machine learning to deep learning approaches[2] has yielded substantial performance improvements. to current applications on Amazon product review datasets, the field transforms raw, unstructured customer feedback into structured, actionable business intelligence.

Our research addresses the practical question of optimal model selection for Amazon product review sentiment analysis by implementing and comparing representative approaches from both traditional machine learning and deep learning paradigms. Specifically, we investigate:

How do traditional TF-IDF + Logistic Regression approaches compare with modern LSTM + Word2Vec architectures?

How do attention mechanisms, bidirectional processing, and regularization affect LSTM performance?

II. Methodology

A. Data Source and Characteristics

We utilized the Amazon Product Reviews dataset (Electronics category) sampling 1 million reviews spanning 18 years. Such a large scale gives us real-world e-commerce context and natural class distribution reflecting actual customer sentiment patterns. Each review consists of:

- **Review text**: The main textual content written by users
- **Rating**: A numerical score from 1 to 5 stars
- **Product metadata**: Product ID and category information
- **Temporal information**: Review timestamp

We transformed the 5-star rating system into binary sentiment labels following established practices in sentiment analysis research:

- **Positive sentiment (1)**: Reviews with ratings ≥ 4 stars
- **Negative sentiment (0)**: Reviews with ratings ≤ 2 stars
- **Excluded**: Neutral reviews (3 stars) to ensure clear sentiment polarity

Exploratory analysis revealed significant positive bias (>70% positive reviews), with the top 10 most-reviewed products all receiving predominantly positive ratings.

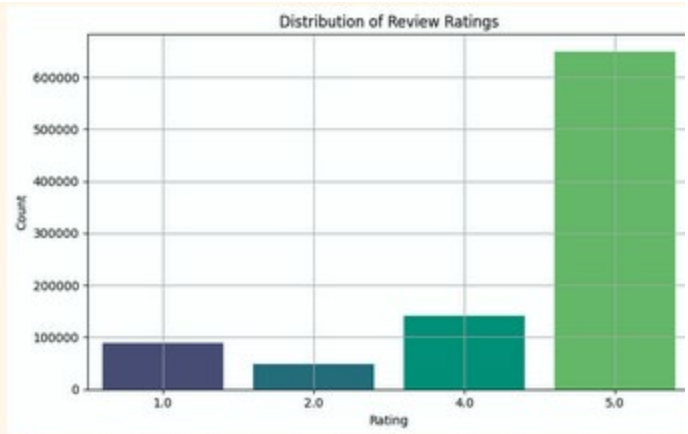

Figure 1: Bar Plot showing the distribution of positive and negative reviews in the Amazon electronics reviews dataset.

B. Text Preprocessing

To prepare the Amazon reviews data for our natural language processing models, we applied a series of preprocessing techniques aimed to clean and normalize the raw data, reduce noise and improve model performance. The techniques include converting all text to lowercase, tokenization to split the text into individual words (tokens), removing common English stopwords, and WordNet lemmatization to preserve semantic meaning while reducing vocabulary size from 1M+ to 32k tokens. The cleaned tokens are joined back as full strings into a cleaned_text column ready for use in vectorization to extract term frequency, n-grams and co-occurence of words.

For LSTM implementation: Vocabulary construction included only words appearing $\geqslant 5$ times, with special tokens <PAD> and <UNK> for sequence normalization and out-of-vocabulary handling.

C. Logistic Regression with TF-IDF Architecture

Logistic Regression with TF-IDF is a strong baseline approach for sentiment analysis classification tasks, especially on a large structured dataset such as Amazon product reviews dataset.

TF-IDF Vectorizer: TF-IDF (Term Frequency-Inverse Document Frequency) transforms raw text data into numerical feature vectors by assigning higher weights to terms that are commonly present in a review but rare amongst the entire corpus. TF (Term Frequency) tracks how frequently a word appears in a review. IDF (Inverse Document Frequency) assigns weights such that the more common words have less influence. This helps place importance on the meaning and informative words and downplay the more common, less informative words such as “the”, “is”, etc. Parameters such as max_features, ngram_range, use_idf and max_df allow for tuning the captured detail. An ablation study was conducted using different configurations of these parameters to assess their impact on model performance.

Logistic Regression Classification: Logistic Regression using TF-IDF vectors performs well in modeling the probability that a given review belongs to a positive or negative sentiment class based on the weighted vectors. After vectorization, each review is passed to the logistic regression model which applies the linear function, sigmoid activation function to compute the probability of a positive sentiment between 0 and 1, prediction rule to apply the predicted positive or negative label and a loss function to penalize the wrong predictions. We also applied K-fold cross validation with k=5 folds to ensure the model generalizes well across different data splits. The average accuracy and F1 score across the 5 folds is tracked to determine the model performance and best performing configuration.

D. RNN-LSTM Model Architecture

The primary limitation of the bag-of-words approach is its inability to understand word order, syntax, and context. A review like "This product is not good" would be seen as having similar features to "This product is good, not bad." To address this, the analysis incorporates two sequential deep learning models that process text as an ordered sequence.

Word Embeddings: Words were first converted into dense, low-dimensional vectors using a custom-trained Word2Vec model (Skip-gram architecture, 100-dimensional vectors, context window of 5). Unlike sparse TF-IDF vectors, these embeddings capture semantic relationships, meaning words with similar meanings are represented by similar vectors. Training custom embeddings on the corpus, rather than using generic pre-trained ones, allows the model to learn representations specific to the jargon and context of electronics reviews.

Bidirectional LSTM (Bi-LSTM): The core of the model is a Bi-LSTM layer. This architecture processes each review sequence in two directions—from start-to-end and end-to-start—and concatenates the outputs. This allows the prediction for any word to be informed by both its preceding and succeeding context, which is crucial for resolving ambiguities and understanding negation. The LSTM cells themselves employ a gating mechanism (Forget, Input, and Output gates) to selectively remember or forget information over long sequences, effectively mitigating the vanishing gradient problem that plagues simple RNNs.

Attention Mechanism: A Bahdanau attention layer was applied after the Bi-LSTM layer. This mechanism enables the model to dynamically assign different "attention" weights to each word in the sequence when forming a final representation of the review. It learns to focus on the most sentiment-critical words (e.g., "excellent," "broken") while assigning less importance to neutral, descriptive text, thus mimicking human intuition.

Regularization and Output: A Dropout layer was included to prevent overfitting by randomly setting a fraction of neuron activations to zero during training. The final sentiment prediction is produced by a Dense output layer with a sigmoid activation function. The model was trained using the Adam optimizer and a binary cross-entropy loss function, with a batch size of 64.

III. Evaluation

E. Logistic Regression with TF-IDF

To evaluate the performance of logistic regression with TF-IDF, we implemented a k-fold stratified cross validation approach to ensure that each fold uses a new split of dataset balancing out any skewness and preventing overfitting.

Data Splitting: The stratified KFold method from scikit-learn was used to split the dataset into five folds. In each iteration, four folds were used for training and one fold was used for testing until all the folds are used for testing once.

Vectorization and Training: For each fold, the TF-IDF vectorizer was fitted on the training data and then used to transform both the training and testing data sets. A logistic regression model was trained on the training data vectors.

Ablation Study: We performed an ablation study to evaluate and find the best tuned model configuration for logistic regression with TF-IDF.

1) Max_features: Trained using max_feature sizes of 5000, 10000, 100000, 500000 to evaluate the effects of vocabulary size in logistic regression with TF-IDF.

2) Ngram_range: Used (1,1), (1,2), (2,2) to evaluate the effect of capturing different contextual patterns from unigram, to combination of unigram and bigram and only bigrams.

3) Use_IDF: True vs false to determine whether the application of inverse document frequency weighting had an effect on the model performance.

4) Stop_words: None to test the impact of not removing any stopwords.

5) Lowercase: True vs false - to check the effects of casing.

6) Max_df: 0.9 to remove the extremely common terms and reduce noise.

F. RNN-LSTM Model

A similar ablation study was conducted for the RNN-LSTM model to justify its final architecture by incrementally adding complexity. An 80-20 train-test split was used for performing these ablation studies. Nine architectural variants tested including hidden dimensions (64, 128, 256), layer depth (1-3 layers), dropout rates (0.0-0.5), bidirectional vs unidirectional, and attention mechanisms.

For each of the ablation configurations, the model was evaluated using the 5-fold cross validation strategy.

Performance Metrics: We used accuracy measures to get a percentage of correct predictions across all data points in a fold, and F1-scrore to get a harmonic mean of precision and recall, providing a balanced measure especially since the dataset has an imbalance towards positive sentiments. The accuracy and f-1 score results are averaged from all the folds to get the final average performance metric for each configuration. These average measures are used to compare and rank each configuration to determine the best model design.

Extreme Error Analysis: We performed extreme error analysis to evaluate where the model fails significantly. While the performance metrics can help understand where the model is performing well, an understanding of where the model fails significantly can help tune the model design better. Extreme error analysis helps identify and understand the largest and most impactful misclassifications

made by the model. This includes an analysis of false positives (where the model predicted a review to be positive while the review is negative) and false negatives (where the model predicted a review to be negative while it was positive). After training the best model determined from the ablation study, predictions were generated from the test folds in the cross validation process. False positives and false negatives were extracted from the prediction results. Confidence scores were used to identify extreme errors such as where the model was highly confident on a prediction but was wrong. The review texts from the extreme errors were manually inspected to identify patterns.

IV. Results

G. Logistic Regression with TF-IDF

The baseline logistic regression model using TF-IDF features performed really well and achieved high accuracy and F-1 scores indicating that it is a well-suited model for the amazon reviews dataset. Using Kfold stratified cross validation, the model maintained consistent performance across all 5 splits. The high performance can be a result of a clear separation between positive and negative sentiment in the reviews and logistic regression with TF-IDF's ability to capture defining and discriminative terms.

Below is a table showing the average accuracy and average f1 score from the ablation study:

The best performance was achieved by the model configuration of bigrams in addition to unigrams (ngram_range: (1,2)) with an average accuracy of 0.9374 and f1 score of 0.9640 indicating that capturing short words as long as short phrases helped capture context and improved the model performance significantly. Changes in max_feature size didn't show much variation in the model performance (~0.0007 variation) suggesting that even a smaller vocabulary can perform as well as a larger one. Disabling idf weighting slightly reduced the model performance suggesting that it was important to downweight the common terms. Case sensitivity did not have any effect on the model performance. Using only bigrams (ngram_range: (2,2)) shows a significant drop in model performance indicating that unigrams/single words still held a lot of meaning in analyzing review sentiment.

TABLE I: Ablation Results for TF-IDF

Configuration	Accuracy	F1 Score
Max_features: 10000	0.9300	0.9597
Max_features: 5000	0.9293	0.9593
Max_features: 100000	0.9298	0.9596
Max_features: 500000	0.9298	0.9596
Ngram_range: (1,2)	0.9374	0.9640
Use_idf: false	0.9287	0.9590
Stop_words: none	0.9298	0.9596
Lowercase: true	0.9298	0.9596
Lowercase: false	0.9298	0.9596
Max_df: 0.9	0.9298	0.9596
Max_features: 10000; ngram_range: (2,2)	0.9063	0.9470

Extreme Error Analysis of the best performing logistic regression model design (ngram_range: (1,2)) revealed that many high-confidence errors were false positives of reviews in which the reviews contained a positive word in a negative context. Manual evaluation showed that a large number of these misclassified reviews included the word "great" within a negative context such as "not great". This indicates that the model struggled to capture the contextual relationship between positive and negative terms within the same expression, leading to incorrect sentiment classification. See Figure 3 for the error analysis results.

H. RNN-LSTM Model

The LSTM ablation study shows the single-layer model performed best at +0.44% over baseline, while three layers and no attention both yielded +0.32%. Removing dropout gave a negligible +0.31%, whereas increasing hidden size slightly hurt performance (-0.03%) and reducing it caused a larger drop (-0.32%). The unidirectional variant fell by -0.18%, and high dropout led to the steepest decline at -0.80%. Overall, shallow architectures with balanced hidden sizes offered the best performance–complexity trade-off.

Table II: Ablation Study for the RNN-LSTM Model

Configuration	Accuracy (%)	Parameters
Single Layer	**92.96**	**3,472,431**

Three Layers	**92.84**	**4,262,959**
No Attention	**92.84**	**3,867,438**
No Dropout	**92.83**	**3,867,695**
Larger Hidden	**92.55**	**5,645,999**
Baseline LSTM	**92.52**	**3,867,695**
Unidirectional	**92.34**	**3,470,255**
Smaller Hidden	**92.20**	**3,396,335**
High Dropout	**91.72**	**3,867,695**

An extreme error analysis of the model's outputs revealed a clear systematic failure pattern: all of the top 10 most confident misclassifications were false positives, with predicted probabilities of 100% and exceptionally high losses (9.984–11.870). These reviews featured highly positive lexical cues such as "well fantastic awesome," "worked great computer good price quick delivery," and "love color price awesome," yet were labeled negative due to accompanying low ratings (1–2 stars). Manual inspection suggests that many of these instances are likely the result of label noise — cases where the star rating contradicts the written sentiment — while others appear to involve sarcasm or a polarity shift that the model failed to capture. This indicates the model's strong bias toward positive sentiment keywords and overconfidence in such predictions, as well as its limited ability to detect contrastive cues or reconcile contradictory sentiment signals. Addressing these issues may require cleaning mislabeled data, incorporating sarcasm/contrastive sentiment examples into training, and improving probability calibration.

V. Discussion

Both Logistic Regression with TF-IDF and the RNN with LSTM model revealed similar outcomes in terms of accuracy and model performance. There was a marginal difference between the two outcomes. Logistic Regression is computationally efficient in both training and inference even with large vocabulary sizes. Model fitting on TF-IDF vectors is fast whereas RNN with LSTM requires significantly more computation due to sequential processing of tokens and the overhead from embedding lookups. Training time is longer in RNN. Logistic regression with TF-IDF captures word importance based on the frequency but it does not encode any semantic meaning or word order beyond the n-grams. On the other hand, RNN with LSTM processes sequences in order allowing it to capture context amongst farther or longer phrases better. We have seen in the logistic regression error analysis that the model struggled to produce correct sentiment prediction when the review included a positive word within a negative context. Such cases can be better evaluated by the use of RNN where longer phrases and sequences of words are used.

Logistic Regression can be preferred on large structured data such as the amazon reviews as it offers a great trade-off between accuracy, speed and interpretability whereas RNN with LSTM could be preferred when the language is much more nuanced and the word order, context play a key role or when the dataset is less structured and contains more noise.

VI. Conclusion

Logistic Regression with TF-IDF performed exceptionally well on the amazon reviews dataset suggesting that the dataset has a clear separation of positive and negative sentiment. The approach provided speed, reliability and a computationally effective way of analyzing and performing sentiment analysis on the amazon reviews dataset. An accuracy of 93% and an F1 score of 0.96 make it a good fit for large scale sentiment analysis on structured datasets of similar nature. However, extreme error analysis revealed that the model struggled when a positive word appeared within a negative context and misclassified the review as a false positive. Future improvements could involve incorporating trigrams to capture richer local context to better understand semantic relationships and improve contextual interpretation.

Our best RNN-LSTM model (Bi-LSTM + attention, single LSTM layer, 128 hidden units, dropout 0.3) reached a final test accuracy of 93.91% with a test loss of 0.1577. Classwise performance shows clear asymmetry that mirrors the dataset imbalance: for the Positive class the model achieved precision 0.9581, recall 0.9711, and F1 0.9645, while for the Negative class it attained precision 0.8183, recall 0.7540, and F1 0.7848. Thus, the model is highly reliable at recognizing positive sentiment and slightly conservative at flagging negatives, missing

some hard negative cases (lower recall) but keeping false alarms relatively controlled (precision > 0.81). Compared to the TF-IDF + Logistic Regression, the LSTM delivers comparable a lot better Positive-class F1. This supports the error inspection: the most confident mistakes are predominantly false positives where strongly positive tokens dominate despite low star labels or contradictory cues.

REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in Proc. EMNLP-2002, 2002, pp. 79-86.

[2] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in Mining Text Data, C. Aggarwal and C. Zhai, Eds. New York, NY, USA: Springer, 2012, pp. 415-463.