Comparison of Generative and Discriminative Language Models for Analyzing Sentiments in Online Product Reviews

Afraa Noureen (an300), Ayush Gupta (ag758), Jiwon Shin (js1149)

Abstract

This report's objective is to provide a comprehensive comparison of sentiment analysis results obtained from various models applied to online product reviews from an e-commerce platform. We employed two distinct models: a generative probabilistic language model and a discriminative neural network, with the goal of identifying the most suitable model for accurately classifying consumer sentiments. For the generative probabilistic language model, we utilized a Multinomial Naive Bayes approach, while for the discriminative neural network, we implemented a Bidirectional LSTM Neural Network. Our primary focus in this comparison will be on assessing the efficacy, accuracy, computational efficiency, and interpretability of these models.

Background

Sentiment analysis, a natural language processing tool, involves the extraction of sentiments, whether positive, neutral, or negative, from textual data. As businesses continually strive to enhance their existing products and services or introduce new offerings to drive revenue and improve the overall customer experience, the accurate analysis of consumer sentiments becomes increasingly critical amid intensifying competition. We are confident that by conducting sentiment analysis on online shopping mall product reviews, utilizing data from platforms such as Amazon and eBay, and extending our model to synthetic datasets, we can provide valuable insights to a diverse range of business stakeholders across various sectors. This analysis has the potential to significantly streamline and economize the resources and time that would otherwise be allocated to customer analysis efforts.

Dataset

We initially collected a dataset from Kaggle, which included online product reviews from Amazon and Ebay, consisting of 360,000 rows. After thorough cleaning, we refined the dataset to contain 92,100 reviews spanning various consumer feedback categories. This dataset is structured with three columns: sentiment (categorized as positive, negative, or neutral), sentiment labels (-1 for negative, 0 for neutral, and 1 for positive), and the original text. To ensure balanced data representation, we distributed an equal number of reviews for each sentiment category during the preprocessing stage. This refined dataset served as the primary training data for our models.

In addition to the Kaggle dataset, we created a separate synthetic dataset exclusively for testing purposes. This synthetic dataset mirrors the structure of the original dataset, with sentiments labeled as -1 (negative), 0 (neutral), and 1 (positive). It was designed to provide an independent evaluation set to assess the performance of our sentiment analysis models. This approach allows for a more objective evaluation of the model's capacity to comprehend and classify sentiments across diverse consumer feedback categories.

This dual-dataset strategy enhances the reliability of our evaluation process, offering a comprehensive assessment of the model's performance in both real-world and synthetic scenarios.

Generative Language Model

In the realm of Natural Language Processing (NLP), generative probabilistic models play a crucial role in various applications, and sentiment analysis is a prime example. Generative probabilistic models are statistical techniques designed to understand the complex probability distribution that underlies text generation. Their primary goal is to identify the emotional tone or sentiment conveyed within sentences or textual data. One prominent example of a generative probabilistic model commonly used for sentiment analysis is the Multinomial Naive Bayes classifier. This classifier analyzes and categorizes text into one of three primary sentiment categories: positive, neutral, or negative, based on word frequencies and conditional probabilities. These models enable the classification of text data into these sentiment categories.

In our specific application, we chose the Multinomial Naive Bayes classifier, a widely used and effective algorithm. To effectively represent our textual data for analysis, we adopted the Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction methods. This choice was guided by several compelling reasons:

- Effective Text Vectorization: To effectively analyze shop reviews and capture sentiment-related insights, we opted for Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) models. These methods were chosen because they excel in converting textual content into numerical vectors, which is essential for identifying sentiment-related words scattered throughout the reviews. BoW simplifies this by representing each review as a collection of its constituent words, while TF-IDF goes a step further by considering word importance within individual reviews relative to the entire dataset. This approach enables us to pinpoint crucial sentiment-related information accurately, enhancing the precision of our sentiment analysis for shop reviews.
- Simplicity and Computational Efficiency: BoW and TF-IDF models offer simplicity and computational efficiency. They streamline the processes of model training and prediction, a

significant advantage when dealing with a large volume of online shop reviews, all while maintaining minimal computational overhead.

- Handling Sparse Data: Large textual datasets are inherently sparse, with a vast vocabulary of words and documents. Similar to the BoW and TF-IDF-based Multinomial Naive Bayes model, the Multinomial Naive Bayes framework adeptly navigates this sparsity through the skilled use of smoothing techniques, effectively calibrating word probabilities.
- Interpretability: Models based on BoW and TF-IDF enhance interpretability. They provide the ability to quantify the impact of specific words in determining different sentiments. This interpretive capacity reveals the reasoning behind specific predictions, enhancing user understanding and facilitating effective debugging.

Considering these factors, we unequivocally chose BoW and TF-IDF-based Multinomial Naive Bayes models for sentiment analysis. These models consistently demonstrate exceptional performance, computational efficiency, the ability to handle sparse data, and enhanced interpretability. These attributes make them a robust choice for analyzing extensive text datasets, including online shop reviews with well-defined sentiment categories.

To implement our approach, we employed both CountVectorizer and TF-IDF Vectorizer to transform our training and test datasets into numerical representations. We deliberately limited the vocabulary size to 7,000 words, a choice that yielded optimal accuracy for both our training data and synthesized datasets. Additionally, we removed English stopwords when building the models. Furthermore, we systematically explored the ideal smoothing parameter (alpha) by evaluating five discrete values: 0.1, 0.5, 1, 2, and 5. This hyperparameter tuning process was conducted meticulously through Grid Search, identifying the most suitable model and hyperparameter values. Subsequently, our model relied on Bayes' theorem to calculate the conditional probability of each review's association with distinct sentiment classes. Using the principle of independence, the model selected the sentiment class with the highest probability to arrive at the final sentiment prediction.

Discriminative Neural Network

To accommodate the lengthy nature of the reviews assessed (each exceeding 100 words), we aimed to incorporate memory within our model. This was crucial to ensure that our predictions encapsulated both past and future data, offering an overall sentiment output. Instead of opting for a Recurrent Neural Network, renowned for its utilization of feedback loops to process data sequences and retain memory within the model, we decided on a Bidirectional Long Short-Term Memory network (BiLSTM). The use of BiLSTM aimed to address potential information loss in the network by leveraging its ability to retain crucial inputs needed for precise sentiment sequence predictions.

In reviewing some of our training data, we encountered reviews that commenced with a positive sentiment but culminated with strongly negative expressions. To address the occasional randomness in the scattering of sentiment-bearing words, we introduced bidirectional layers in our model, preserving information from both the initiation and conclusion of each review.

The initial step involved defining an embedding layer transforming words into 32-dimensional vectors. Unlike one-hot encoding, the vectors were designed to identify word similarities. The neural network was bidirectional, allowing input flow from both directions to ensure continuity of sentiment across preceding and subsequent words in the sequence.

The subsequent layer after embedding was a bidirectional layer containing 64 cells, followed by another bidirectional layer processing inputs in the opposite direction. To convert the outputs into binary classification labels, two Dense layers were used. The first Dense layer used a ReLU activation function with 16 units, while the final layer (output layer) incorporated a softmax activation function to distinguish among the adjusted three output categories.

To mitigate overfitting, Dropouts were inserted across the model's layers. Dropout randomly eliminated a percentage of neurons during transitions between hidden layers, ensuring the model's generalizability to future data. A layered dropout approach was implemented to prevent the loss of critical features before computing class probabilities.

During the model fitting process to our training data, we specified hyperparameter values for epochs and batch_size. The batch_size was set to 64, accommodating 625 total batches from our training dataset of 40,000 rows. With 10 specified epochs, the model underwent exposure to the entire dataset or all 625 batches ten times. Following each batch, predictions were compared to actual labels, and model parameters were updated using gradient descent for further refinement.

During the model fitting process to the training data, hyperparameter values for epochs and batch size were specified. The batch size was set to 32, accommodating 2303 total batches from the training dataset (for real data). With 100 specified epochs, the model was designed to undergo exposure to the entire dataset or all 2303 batches one hundred times. Early stopping, implemented as a callback with a patience of 15 epochs, was used to prevent the model from reaching the full 100 epochs and to restore the model's best weights when necessary. Following each batch, predictions were compared to actual labels, and model parameters were updated using gradient descent for further refinement. The evaluation on real data demonstrated an accuracy of 70.90%.

Results

Generative Language Model Results

Interpretation of Real Data Results

In our first experiment, we trained and applied both Bag of Words (BoW) and TF-IDF-based approaches using the Multinomial Naive Bayes model on our real dataset. BoW achieved an accuracy of 65.57%, while TF-IDF achieved a slightly higher accuracy of 66.07%. This suggests that TF-IDF performed marginally better than BoW. Given our objective, which is to minimize the possibility of negative values being misclassified as positive, we also calculated Recall. For BoW, the Recall was 65.66%, and for TF-IDF, it was 66.16%. Therefore, there is not a significant difference in terms of performance between BoW and TF-IDF, as both accuracy and recall metrics are quite close. Furthermore, In the grid search optimization, it was found that the optimal hyperparameter alpha for both Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) was 5, resulting in the highest accuracy. Regarding execution time, BoW took approximately 4.60 seconds to complete its processing, while TF-IDF completed in a similar timeframe, taking approximately 4.59 seconds. In terms of memory usage, BoW consumed around 507 megabytes (MB), while TF-IDF utilized approximately 521 MB. These memory usage values indicate that there was not a significant difference in memory consumption between BoW and TF-IDF, suggesting that both approaches had similar memory requirements.

Upon analyzing misclassified reviews, we observed the following patterns. The largest portion (25.45%) of misclassified reviews were initially neutral but were incorrectly predicted as negative. Additionally, 24% of misclassified reviews were originally negative but were erroneously classified as neutral. Another 19% of misclassified reviews were genuinely neutral but mistakenly labeled as positive. When considering the entire dataset, it became evident that the most challenging class to correctly classify was the neutral category. This is understandable because neutral reviews often contain both positive and negative elements, making it difficult for our model to make accurate predictions. Furthermore, negative reviews in the actual dataset were frequently misclassified, while positive reviews were generally well-predicted by the model.

When we closely examined the specific reviews that the model mispredicted, it became clear that the model struggled in cases where a more in-depth understanding of the context within the reviews was necessary. This challenge was particularly noticeable in reviews that contained elements of ambiguity or made comparisons, demanding a higher level of precision in classification.
For instance, consider the following review: "Not fantastic but not bad. I added veggies and a bit of sugar as it is slightly bitter. I also added some coconut milk at the end." Originally, this review was rated as neutral with a score of 3 out of 5. However, our model incorrectly classified it as positive.

Similarly, reviews like "I enjoyed the low price, but they put too much garlic in this oil" were actually positive reviews with high ratings but were mistakenly categorized as neutral.

We posit that these instances of misclassification can be attributed to two primary factors. Firstly, certain reviews employ language that lacks clear polarity, rendering it challenging to definitively categorize sentiments as positive or negative. It is noteworthy that the sentiment labels were derived from converting numerical ratings into sentiment labels (e.g., designating 3 stars as neutral). In such cases, it's plausible that many individuals may provide either a negative or positive review while still assigning a 3-star rating to the product. This inherent misalignment in the data presents a significant challenge in sentiment classification.

As a response to this challenge, we explored the use of bidirectional Long Short-Term Memory (LSTM) models to enhance our model's performance. Bidirectional LSTM models have the capability to capture the context of several sentences, potentially addressing the gap in accuracy when dealing with real-world nuanced sentiment in reviews.

Interpretation of Synthetic Data Results

We performed additional experiments utilizing synthetic data to evaluate the performance of our Bag of Words (BoW) and TF-IDF-based Multinomial Naive Bayes models. When working with the synthetic dataset, the BoW model achieved an accuracy of 96.62%, while the TF-IDF model exhibited a slightly higher accuracy of 97.16%. Similarly, the recall scores for the BoW and TF-IDF models were 96.6% and 97.14%, respectively, indicating that the TF-IDF model marginally outperforms the BoW-based model.

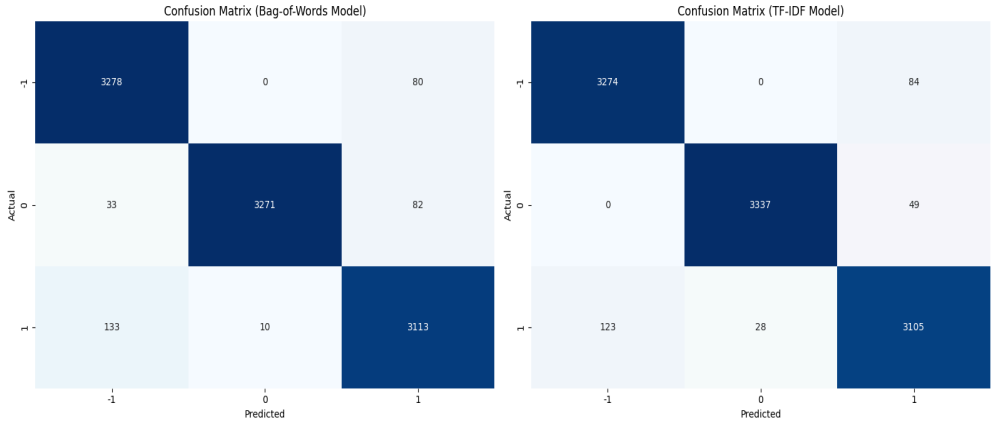

Fig1. Confusion Matrix for the Final Fitted Multinomial Naive Bayes Model using Bag of Words (BoW) and TF-IDF

The confusion matrices of the Bag-of-Words (BoW) and TF-IDF models display striking similarities, but a significant distinction arises in the classification of neutral sentiments. Specifically, while the BoW model accurately identified 3,271 neutral cases, the TF-IDF model surpassed it by correctly classifying 3,337 neutral cases. This superior performance of the TF-IDF model underscores its enhanced classification ability across positive, negative, and neutral values, particularly in the case of neutral sentiments. This distinction holds particular significance as generative probabilistic models prioritize frequency and probability over the nuanced contextual meanings or polysemous interpretations of words.

Regarding computational requirements, both models exhibited nearly identical execution times, approximately 0.72 and 0.71 seconds, respectively. This similarity in execution time could be attributed to the smaller size of our synthetic dataset, containing 49,998 cases compared to our training dataset's 92,100 cases. However, the memory usage for both the BoW and TF-IDF models was higher, totaling 521MB and 531MB, respectively, compared to the training dataset.

To explain the observed increase in accuracy compared to the training dataset, several contributing factors were considered. Firstly, the creation of a synthetic dataset allowed for the introduction of a wider range of sentiments and sentence structures in contrast to our original training dataset. Consequently, our models gained increased versatility and an enhanced capacity to handle a broader spectrum of cases, resulting in a substantial boost in accuracy, from approximately 65-66% to well over 96-97%. Furthermore, the inclusion of the synthetic dataset likely raised the complexity of our models, exposing them to more intricate patterns and nuanced sentiment expressions. This heightened complexity, in turn, empowered our models to more effectively capture and accurately classify sentiments, further amplifying their performance.

Discriminative Neural Network Results

Interpretation of Real Data Results

In our Discriminative Neural Network experiment, we used a Bidirectional Long Short-Term Memory (BiLSTM) architecture to analyze real-world data. The model was trained over 100 epochs, with a batch size of 64, resulting in an accuracy of 70.90% during evaluation on the real dataset. The model architecture comprised an embedding layer transforming words into 32-dimensional vectors, followed by a bidirectional layer with 64 cells, and subsequent dense layers for classification. The total trainable parameters amounted to 371,779, with an overall model size of 1.42 MB. During training, the model demonstrated a steady reduction in loss and an increase in accuracy over epochs. In the evaluation on real data, the model achieved an accuracy of 70.90%. A more detailed performance breakdown reveals a precision of 70.70%, recall of 70.90%, and an F1-score of 70.78%. The

confusion matrix illustrates the model's proficiency in classifying reviews across three sentiment categories: positive, neutral, and negative.

In terms of computational considerations, the model fitting process on the real dataset took approximately 320 seconds per epoch, indicating a computationally intensive task. The absence of GPU utilization might have contributed to the extended training time. However, the CPU-optimized TensorFlow binary showed efficient utilization of available CPU.

However, the model faced difficulties when dealing with uncommon or made-up words, as seen in instances like "Ryy" or "Kopp." This challenge is common in both machine and human sentiment analysis. Additionally, the model struggled with sequences of words that alternated between positive and negative probabilities, such as in the phrase "Excellently bad!" Moreover, statements with ambiguous context, like "It is really a budget phone," added complexity, a challenge that is also recognized in human classification due to implicit biases.

Despite these challenges, the model demonstrated strengths in understanding complex sentences that included Internet Slang. It effectively identified key markers of sentiment, even accurately classifying grammatically incorrect sentences with abbreviations like 'w' for 'with.' While we acknowledge the model's limitations, its efficiency, speed, and reasonably accurate performance are notable. This emphasizes its potential as a valuable tool in sentiment analysis. The exploration of bidirectional LSTM models is geared towards mitigating these challenges and potentially bolstering the model's capacity to capture nuanced sentiments in real-world reviews.

Interpretation of Synthetic Data Results

In our examination of synthetic data, our discriminative neural network showed notable performance improvements. Over the course of 27 epochs, the model achieved a compelling accuracy of 98.00% on the synthetic dataset. The precision, recall, and F1-score metrics underscored the model's robustness, as evidenced in the confusion matrix.

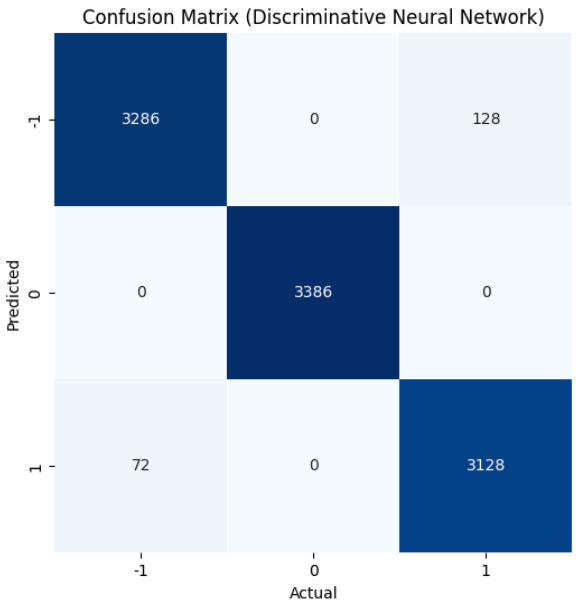

Fig2. Confusion Matrix for the Discriminative Neural Network

Although the numbers look promising, it's crucial to approach the findings with caution when considering real-world applications. The synthetic dataset has inherent characteristics, such as possible randomness and noise, which might create challenges for the model to adapt to unseen situations.

In terms of computational requirements, the execution times for each epoch were consistent, averaging approximately 165 seconds. The model showed practical efficiency, particularly given the similarity in execution times for both the Bag-of-Words (BoW) and TF-IDF models. The total trainable parameters amount to 371,779, occupying 1.42 megabytes.

For a comprehensive analysis, we recommend a detailed examination of potential biases introduced during the synthetic data synthesis process. It will provide valuable insights into the model's adaptability to real-world conditions. Despite potential disparities in synthetic data quality, the accuracy levels of the DNN model (98.00%) compare relatively favorably to those achieved on the training dataset (BoW 96.62%, TF-IDF 97.16%).

In summary, our deep learning model performs well on synthetic data, focusing particularly on computational efficiency and memory usage. A detailed understanding, along with continuous exploration into the characteristics of synthetic data, is essential for deploying the model confidently in various real-world applications.

Limitations

Generative Language Model

When employing a BoW and TF-IDF-based Multinomial Naive Bayes model for sentiment analysis in shop reviews, several limitations became apparent:

- Loss of Contextual Information: Both BoW and TF-IDF disregard the order and context of words, which is particularly problematic in sentiment analysis. In our shop review dataset, where the average review consists of 4-8 sentences, the meaning of a sentence often hinges on the arrangement of words and their relationships. This loss of context can impact the accuracy of sentiment classification.
- Inability to Handle Negations, Modifiers, and Polysemy: Multinomial Naive Bayes models, when coupled with BoW or TF-IDF, tend to struggle with effectively capturing negations (e.g., 'not good') and modifiers (e.g., 'very good'). Additionally, these models treat each word as an independent feature, making them less adept at handling polysemous words (words with multiple meanings), which can introduce ambiguity into sentiment analysis.
- Out-of-Vocabulary Words and Fixed Vocabulary: BoW and TF-IDF heavily rely on predefined fixed vocabularies. This reliance increases the likelihood of missing sentiment-carrying words or phrases that are not included in the predefined vocabulary. Furthermore, when words not present in the vocabulary are encountered, these methods simply ignore them. Consequently, our model's accuracy may suffer if testing or synthetic datasets contain words absent from the training dataset's vocabulary.

These limitations highlight the necessity of adopting context-aware and adaptive approaches to sentiment analysis, in conjunction with Multinomial Naive Bayes or generative probabilistic models. This need is particularly pronounced in contexts such as shop reviews, where precision in sentiment classification relies heavily on considering the specific context and wording of the reviews.

Discriminative Neural Network

Despite the success using our DNN model in sentiment analysis, it is important to acknowledge certain limitations, especially in the context of our specific implementation:

- Computational Resource Demands: Our DNN model requires substantial computational resources throughout both the training and inference phases. The considerable number of parameters and the necessity for multiple epochs might lead to extended training durations, potentially affecting scalability, especially in environments with limited computing resources.

- Reliance on Training Data Quality: The effectiveness of our DNN is significantly influenced by the quality and representativeness of the training dataset utilized. If our dataset lacks diversity, shows biases, or fails to accurately mirror the real-world scenarios we intend to address, the DNN's ability to generalize proficiently may be compromised.
- Sensitivity to Noisy Data: Our DNN's susceptibility to inaccuracies becomes apparent when confronted with noisy or inadequately preprocessed data. In practical situations where data may encompass inconsistencies, absent values, or errors, the model's robustness becomes susceptible to compromise.
- Need for Hardware Acceleration: Attaining optimal performance with our DNN often hinges on the accessibility of specialized hardware, such as GPUs. In the absence of proper hardware acceleration, the model's inference speed may fall short, impacting its real-time applicability in practical scenarios.

Recognizing these constraints is essential in making well-informed decisions regarding the implementation and ongoing enhancement of our dedicated DNN model within the realm of practical, real-world applications, particularly in the context of product reviews.

Conclusion

In our thorough examination of sentiment analysis models for online product reviews, we compared a Generative Language Model (Multinomial Naive Bayes) and a Discriminative Neural Network (Bidirectional LSTM). We analyzed real-world data from platforms like Amazon and eBay, along with synthetic datasets for rigorous testing.

The Generative Language Model, using Bag of Words (BoW) and TF-IDF, achieved accuracy rates of around 65-66% on real world data and ~96% on synthetic data. However, limitations, such as contextual information loss and struggles with negations and modifiers, highlight the need for context-aware approaches in product review sentiment analysis.

On the other hand, the Discriminative Neural Network, with a Bidirectional LSTM, showed superior accuracy at ~71% on real-world data and an impressive 98% on synthetic data. Despite computational demands and challenges with uncommon words, its efficiency and robust performance suggest its potential in sentiment analysis. However, acknowledging limitations, we advise a cautious approach in real-world applications, emphasizing understanding biases in synthetic data.

Each model had distinct strengths and weaknesses. The Generative Language Model excelled in interpretability and simplicity, while the Discriminative Neural Network showed relatively higher accuracy and adaptability. Balancing computational efficiency, interpretability, and real-world

performance is crucial in choosing an optimal model. Navigating model trade-offs, understanding limitations is vital. The Generative Language Model's struggles with contextual nuances and the Discriminative Neural Network's resource-intensive nature emphasize the need for a nuanced approach in sentiment analysis model selection.