IDS 703 Nature Language Process
Final Project

MOVIE REVIEWS SENTIMENT ANALYSIS

December 10, 2023

George Wang
Yanzheng Wu
Yi Chen

1 Abstract

This project delves into sentiment analysis of IMDB movie reviews with the aim of classifying the reviews as positive or negative. Utilizing a dataset of 50,000 real reviews sourced from Kaggle, the study employs comprehensive preprocessing techniques. Two distinct approaches are implemented for classification. The first utilizes a Naive Bayes Classifier, trained on 80% of the dataset and tested on the remaining 20%, showcasing the results through visual interpretations. The second approach involves developing a Neural Network model, incorporating embedding, convolutional, and LSTM layers, and is optimized for performance. This model demonstrates promising accuracy on real data but exhibits a decrease in accuracy when applied to synthetic data. The project culminates with a discussion on the strengths and limitations of both methods, providing an insightful comparison on their efficacy with real and synthetic data, and underscoring the applicability and robustness of generative probabilistic models and discriminative neural networks in sentiment analysis.

2 Introduction

Sentiment analysis, a critical and hot topic in natural language processing, has gained substantial prominence in extracting subjective opinions and useful sentiment information embedded in textual data. Movie reviews, in particular, serve as an exemplary context for the application of sentiment analysis. These reviews inherently reflect the audience's attitudes, preferences, and emotions towards a cinematic experience, and thus often exhibit pronounced polarity. The dynamic nature of movie reviews, laden with expressive language and varying sentiments, renders them a compelling domain for the exploration and application of sentiment analysis techniques. Consequently, this project aims to explore two approaches for sentiment analysis under the context of IMDB movie reviews.

Our analysis is grounded on a dataset containing 50,000 real movie reviews. The data underwent comprehensive preprocessing, including lowercasing, decontraction, HTML and URL removal, punctuation cleaning, and stopword removal, ensuring a clean dataset for model input. Our object is to classify these reviews into two categories (positive or negative) based on their sentiment. We obtain the data from Kaggle [1] and use [2] for the reference of the development of data cleaning and model development.

We first employ a Naive Bayes Classifier for the classification. The approach was trained with a randomly selected 80% of the dataset and tested on the rest, which had visualized results for interpretability. Additionally, a variable-order Markov Chain Text Generator was developed to create 50,000 synthetic reviews. These reviews underwent similar processing and testing, providing insights through word cloud visualizations and model performance

analysis.

The second approach entailed developing a Neural Network model for Natural Language Processing. The architecture included embedding layers, convolutional and LSTM layers, and dense layers, with a focus on optimizing performance through a learning rate of 0.001 and ten training epochs. This model was initially trained and tested on the real data, showing promising accuracy and loss metrics. However, a significant drop in accuracy was observed when the model was applied to synthetic data.

The subsequent sections then go to a comprehensive discussion of the merits and limitations of both approaches. It included an analysis of error sources, and performance metrics, and highlighted the differences in model effectiveness with real and synthetic data. This comparison provided valuable insights into the applicability and robustness of generative probabilistic models and discriminative neural networks in sentiment analysis tasks.

The rest of the report will be structured as follows: we will first describe our data and data-preprocessing process in Section 3. Then we will give a comprehensive overview of the two employed approaches for sentiment analysis in Section 4. Section 5 will discuss and evaluate the results obtained and Section 6 will conclude.

3 data

We use the Large Movie Review Dataset [3] as our dataset, which contains a set of 25,000 highly polar IMDB movie reviews for training, and 25,000 for testing. The reviews are labeled into two categories, "positive" and "negative", based on their sentiment and the counts of positive reviews and negative ones in the dataset are equal.

We first preprocess and clean our data to ensure that our later analyses are conducted on a refined and standardized corpus. Specifically, we lowercase all words and decontracte the text by transforming contracted words like "shouldn't" into their full forms. We also remove the HTML tags and URLs from the textual content of the reviews. Elements with negligible impact on sentiment, including punctuations and meaningless stop words such as "this" and "that", are also removed. The list of stop words are obtained from NLTK stopwords corpus. Notably, we exclude some stop words from the NLTK stopwords corpus, mainly negating words, since they can make a big difference to the sentiment of text.

Figure 1 shows the word cloud of the total movie reviews, those with negative sentiment and those with positive sentiment respectively. We can see that though many words (e.g., "movie", "film", "one") occur frequently in both positive reviews and negative ones, some variations are highlighted in the prevalence of words like "well" in positive reviews and

Figure 1: The word cloud of IMDB reviews.

"bad" in negative ones.

4 Methodology

The methodology for the sentiment analysis project on IMDB movie reviews encompasses several steps, focusing on data preprocessing, probabilistic modeling, building discriminative neural networks, and evaluation using both real and synthetic data.

4.1 Generative probabilistic model

4.1.1 Initial Analysis on Unprocessed Data

Before delving into data preprocessing, an initial run of the Naive Bayes classifier was conducted on the raw, unprocessed dataset. The purpose of this initial analysis was to establish a baseline for the model's performance, which would later serve as a point of comparison to understand the impact of data cleaning and normalization on the model's effectiveness.

4.1.2 Probabilistic Modeling - Naive Bayes Classifier

The first approach employed a Naive Bayes classifier [4] for sentiment analysis. The model operates on the principle of conditional probability, utilizing Bayes' Theorem. It assumes independence between features (words in this context) and calculates the probability of a document (review) belonging to a certain class (positive or negative sentiment) based

on the frequency of words. Mathematically, the model evaluates the posterior probability $\mathbb{P}(C \mid X)$ for a class C given a feature vector X, and classifies the document based on the highest probability.

$$\mathbb{P}(C \mid X) = \frac{\mathbb{P}(X \mid C) \cdot \mathbb{P}(C)}{\mathbb{P}(X)}$$

The dataset was divided into an 80-20 split for training and testing. A Bag of Words model was used for feature extraction, transforming the text data into a set of numerical features representing the frequency of words in a document. The Naive Bayes model was then trained on the processed data, and predictions were made on the test set.

To evaluate the model's performance, various methods were used:

- Generation of word clouds to visually represent the most frequent words in the entire dataset, as well as in correctly and incorrectly classified reviews.
- Identification of the most indicative words for positive and negative sentiments.
- Calculation of the accuracy and generation of a classification report to assess the model quantitatively.

4.1.3 Generative Modeling - Markov Chain Text Generator

In addition to the Naive Bayes model, a Markov Chain Text Generator [5] was developed to create synthetic movie reviews. This generative model was capable of producing reviews of variable orders, thereby mimicking the style of the original dataset. The Markov Chain operates on the premise of state transition probabilities, predicting the next word based on the current word or phrase, thereby capturing the sequential nature of text. A total of 50,000 synthetic reviews were generated, maintaining a balance between positive and negative sentiments.

$$\mathbb{P}(w_t \mid w_{t-1}, w_{t-2}, ..., w_{t-n})$$

The Markov Chain formula represents the probability of the next word given the previous n words. Here, w_t is the current word, and $w_{t-1}, w_{t-2}, ..., w_{t-n}$ are the preceding n words.

The synthetic data underwent similar testing and evaluation as the real data:

- The Naive Bayes model was retrained and tested on the synthetic dataset.
- Word clouds were generated for the entire synthetic dataset and for the subsets of correctly and incorrectly classified synthetic reviews.

4.1.4 Evaluation and Comparison

The project's methodology was designed to compare the performance of the Naive Bayes classifier on both real and synthetic data. This comparison was done both qualitatively, through visual representations like word clouds, and quantitatively, using metrics such as accuracy and classification reports.

4.1.5 File Management

Throughout the project, key results, such as misclassified reviews and the synthetic dataset, were saved to the Colab virtual machine and downloaded for further analysis and record-keeping.

4.2 discriminative neural network

We train the discriminative neural network model using Convolutional Neural Networks and Long Short-Term Memory networks. CNNs excel in extracting hierarchical spatial features from data in text analysis. These local features can include common sentiment markers like "amazing" or "terrible", as well as more complex linguistic constructs. On the other hand, LSTMs are adept at handling sequential data and are particularly well-suited for tasks that require an understanding of context over longer sequences. CNNs and LSTMs provide a comprehensive approach to text analysis in sentiment analysis tasks. While CNN layers initially extract salient features and patterns from the text, the LSTM layers interpret these features within the broader context of the entire review. The structure and parameters of the discriminative model is shown in Figure 2.

We have 7 layers in the discriminative neural network model. The first layer in our model is the embedding layer, which is designed to process text input. This layer converts word indices to dense vectors of size 100. Second, the convolutional layer has 50 filters and a kernel size of 5. It's designed to extract local features from the sequence of word embeddings. The ReLU activation function adds non-linearity to the model. The third layer is the max pooling layer, which reduces the dimensionality of the data by pooling over 4-word windows. This step helps to decrease the computation and also to extract more robust features. The fourth layer is the LSTM Layer. This LSTM layer with 64 units processes the sequence data. It captures long-term dependencies in text, aiming to understand the overall context of a review. The fifth layer is the dropout Layer. This layer randomly sets input units to 0 with a frequency of 0.5 at each step during training time, which helps to prevent overfitting. We use the dense layer with 32 units and ReLU activation function to learn non-linear

Layer (type)	Output Shape	Param #
embedding_9 (Embedding)	(None, 120, 100)	1000000
conv1d_9 (Conv1D)	(None, 116, 50)	25050
max_pooling1d_9 (MaxPooling1D)	(None, 29, 50)	0
lstm_9 (LSTM)	(None, 64)	29440
dropout_9 (Dropout)	(None, 64)	0
dense_18 (Dense)	(None, 32)	2080
dense_19 (Dense)	(None, 1)	33

Total params: 1056603 (4.03 MB)
Trainable params: 1056603 (4.03 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 2: The model structure of the discriminative model.

combinations of features. Since this model is used for binary classification, the output layer has a single neuron. The sigmoid activation function squashes the output between 0 and 1.

4.3 Overall Assessment

This comprehensive methodology allowed for a detailed analysis of sentiment analysis models, highlighting their strengths and limitations in handling real and synthetic data, and providing insights into the nature of text data and the effectiveness of different modeling approaches in NLP tasks.

5 Results

5.1 Evaluation of Generative Probabilistic Model on Real Data

The results of the Naive Bayes classifier on the real data indicate a nuanced understanding of sentiment analysis. The presence of words such as "well," "great," "story," and "good" among the most indicative of positive reviews aligns with general expectations. Interestingly, the word "not" also appears prominently in positive reviews, suggesting its role in negations or complex expressions of sentiment.

Words most indicative of positive reviews	well	great	story	would	good	like	one	movie	film	not
Words most indicative of negative reviews	time	good	bad	even	would	like	one	file	movie	not

Table 1: Words most indicative of all reviews.

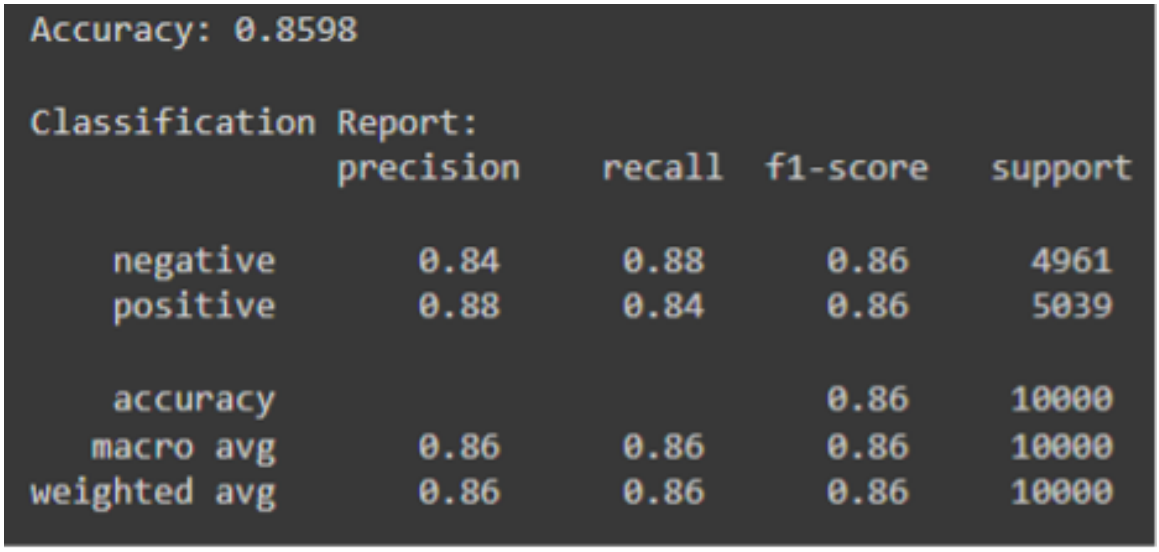

Figure 3: Evaluation of generative probabilistic model on real data.

The negative reviews are characterized by words like "bad," "even," and "time," alongside

"good" and "movie," reflecting a mixed sentiment or the presence of expectations versus reality themes in negative reviews. The accuracy of 85.98% demonstrates the model's robustness in distinguishing between positive and negative sentiments under real-world conditions.

The precision, recall, and F1-scores in the classification report provide a more granular understanding of the model's performance. The balanced scores across both categories suggest that the model is equally adept at identifying both positive and negative sentiments, with slightly higher precision for positive reviews, indicating fewer false positives in this category.

Additionally, the word cloud for both correctly and incorrectly classified reviews (Figure 4) shows that there are lots of same of similar words appearing at the same time, which also reflects the limitation of the model on need for contextual analysis.

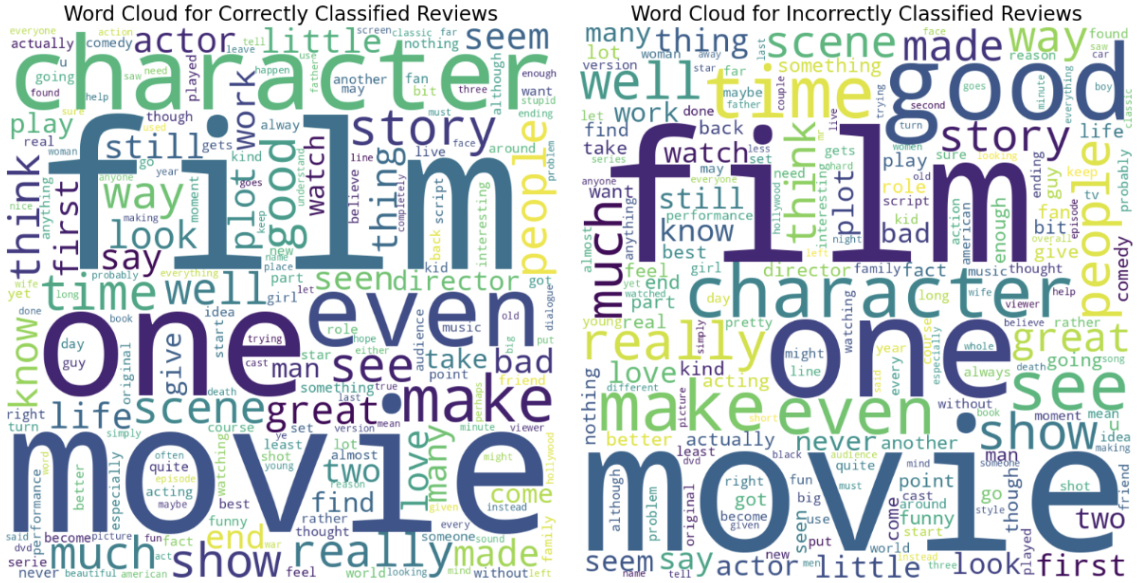

Figure 4: Word cloud for correctly and incorrectly classified real reviews.

5.2 Evaluation of Generative Probabilistic Model on Synthetic Data

The Naive Bayes classifier's performance on synthetic data shows a higher accuracy of 91.71%, suggesting that the synthetic data, generated by the Markov Chain model, might be more distinct in terms of sentiment indicators or less nuanced compared to real reviews.

Words most indicative of positive reviews	would	story	like	well	also	the	one	movie	film	not
Words most indicative of negative reviews	story	bad	even	like	would	the	one	file	movie	not

Table 2: Words most indicative of all reviews.

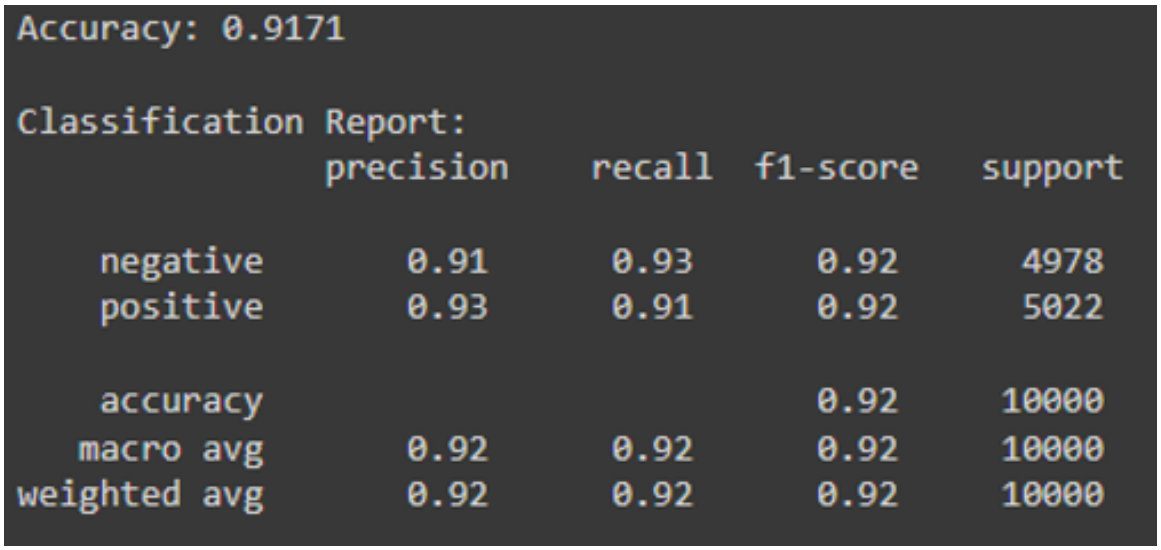

Figure 5: Evaluation of generative probabilistic model on synthetic data.

The most indicative words for positive and negative reviews in synthetic data share some similarities with the real data, such as "story," "like," "movie," "film," and "not." This consistency indicates that the Markov Chain model successfully captured key aspects of the language used in real reviews. However, the presence of "the" as a top word in both categories suggests a potential over-representation of common words in synthetic data.

The higher precision, recall, and F1-scores in the synthetic data indicate that the classifier found it easier to distinguish between sentiments in this dataset. This could be due to more pronounced or less ambiguous language use in the synthetic reviews, a common characteristic of generated text which tends to follow more predictable patterns.

As the figure **??** shows, the same words are frequently found in both correctly and incorrectly

Figure 6: Word cloud for correctly and incorrectly classified synthetic reviews.

classified word clouds. This could be the reason for word ambiguity and model limitation.

5.3 Evaluation of Neural Network Model on Real Data

The training accuracy starts at 83.11% and steadily increases to a final value of 99.58%. This indicates that the model is learning effectively from the training data, fitting it more closely with each epoch. The validation accuracy begins at 87.90%, which is reasonably high, but it does not improve substantially throughout the training process, ending at 85.79%. The model does not generalize as effectively to the validation data as it does to the training data. The training loss decreases from 0.3708 to 0.0163, showing that the model is becoming more confident in its predictions on the training set. The validation loss reduces initially but then shows fluctuations and an upward trend, ending higher than where it started (0.2844 to 0.5720), which has the potential issue of overfitting.

The model learns and fits the training data exceedingly well with high training accuracy and low training loss. However, the model does not have good performance on the validation set, indicating the model is overfitting. Moreover, the significant gap between training and validation accuracy, combined with the increasing validation loss, indicates high variance in the model. The model is not robust to new data. One unexpected result is the upward trend and fluctuations in validation loss are concerning. After the fourth epoch, the validation loss

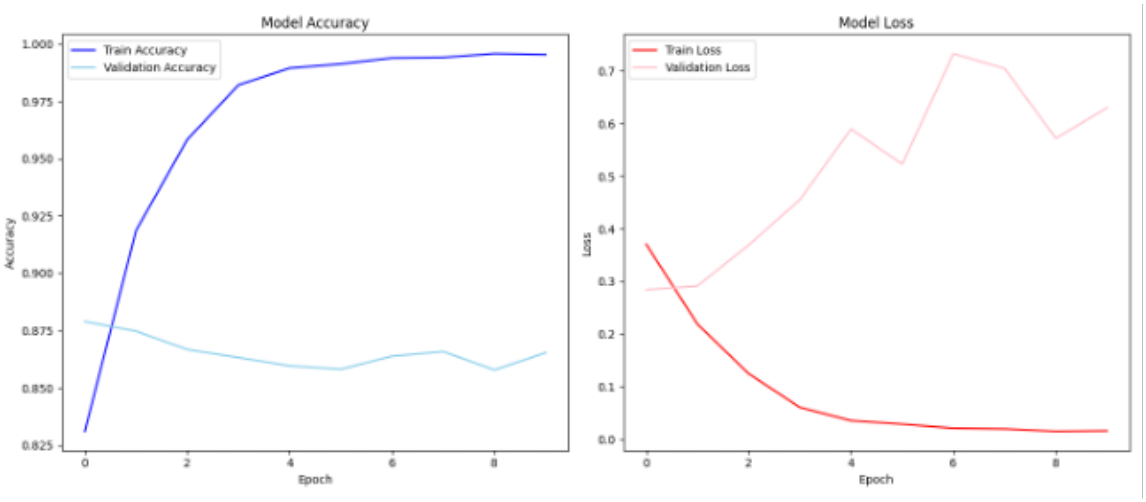

Figure 7: Evaluation of neural network model on real data.

starts to increase, which is not expected in a well-generalizing model. Typically, we expect the validation loss to decrease and stabilize, similar to the training loss. We can optimize the model by early halt training aiming to prevent overfitting. Furthermore, increasing dropout rates might help the model generalize better. Moreover, we can also adjust learning rates, the number of neurons, or layer configurations in the future optimizations.

5.4 Evaluation of Neural Network Model on Synthetic Data

High training accuracy and low training loss indicate that the model has effectively learned the patterns present in the synthetic training data. However, the model performs poorly on the synthetic validation set. The validation loss decreases initially but then increases and fluctuates significantly, indicating that the model's predictions on the validation set are becoming less certain, which could be due to overfitting. Compared to real data, the synthetic data generation process might not have fully captured the nuances and diversity of real-world language used in movie reviews. Synthetic data may lack the idiosyncrasies and edge cases present in natural language, which can lead to a model that performs well on training data but less so on real or validation data. Moreover, the simplicity of Synthetic Data may affect the lower accuracy with synthetic data. If the synthetic data is not complex enough, the model might not learn to handle the complexities of real-world data. Synthetic data might miss out on subtle linguistic cues that are important for sentiment analysis.

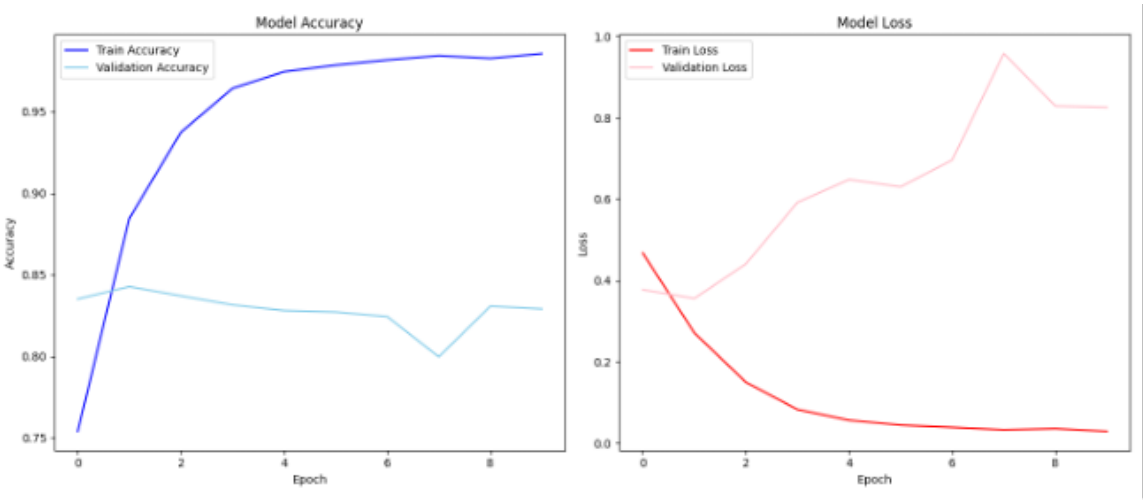

Figure 8: Evaluation of neural network model on synthetic data.

6 Discussion and Conclusion

The Naive Bayes classifier performed well on real data, achieving an accuracy of 85.98%. This demonstrates its reliability and effectiveness in handling real-world sentiment analysis tasks. The model effectively identified key words with strong sentiment indications, such as "great", "bad", "good", indicating its capability in understanding and differentiating positive and negative sentiments. However, the model struggled with reviews expressing complex sentiments, likely due to the Naive Bayes assumption of feature independence, which fails to capture the context and complexity of language. Certain words, like "not", appeared frequently in both positive and negative reviews, potentially causing confusion in sentences with negations or double negatives.

Higher accuracy (91.71%) on synthetic data suggests that the Naive Bayes classifier is more adept at handling the more regular, predictable language patterns found in synthetic reviews. Accurate identification of sentiment-indicative words in synthetic data shows that the model is effective in recognizing sentiment tendencies in text generated by the Markov Chain. We speculated that the high accuracy might be partially due to the simplified language structure of synthetic data, rather than the model's inherent strength. Synthetic data may lack the complexity and diversity found in real data.

CNN & LSTM models, while offering superior accuracy on real data for complex tasks like sentiment analysis due to their ability to capture nuanced sequential patterns, suffer from longer training times and require significant computational resources, with GPUs

being preferred for optimal performance. Their complexity also leads to low interpretability, making them less ideal in situations where understanding model decisions is critical. Moreover, low correctness for synthetic data reveals that the CNN & LSTM model might not be able to learn effectively from the synthetic data due to its complexity and reliance on large, diverse datasets. The performance on synthetic data is poor indicating that the model requires rich, varied datasets that better capture the complexities of natural language. Conversely, while simpler models like Naive Bayes may offer greater interpretability and require less computational power, making them suitable for rapid development cycles and resource-constrained environments, they may not match the performance of CNN & LSTM models on tasks involving complex data patterns.

	Naive Bayes		CNN & LSTM	
Dataset	real	synthetic	real	synthetic
Correctness	85.98%	91.71%	87%	51%
Time (sec)	93.728	54.036	5	4
Computational Requirements	CPU		CPU or GPU (preferred)	
Interpretability	higher		lower	

Table 3: Comparison between the two models.

References

[1] L. N, "Imdb dataset of 50k movie reviews." [Online]. Available: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/data

[2] ——, "Sentiment analysis of imdb movie reviews." [Online]. Available: https://www.kaggle.com/code/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews/notebook

[3] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[4] IBM, "What are naive bayes classifiers?" [Online]. Available: https://www.ibm.com/topics/naivebayes

[5] G. Pernicano, "Text generation with markov chains: An introduction to using

markovify." [Online]. Available: https://towardsdatascience.com/text-generation-with-markov-chains-an-introduction-to-using-markovify-742e6680dc33