Sentiment analysis on IMDB movie reviews

Ilian Ariessen and **Sebas van Sluijsdam**

1 Introduction

Sentiment analysis, also known as opinion mining, plays a large role in understanding and analyzing people's opinions and emotions expressed in texts. With the growth of online reviews and social media content, sentiment analysis has gained attention in various domains, including marketing, customer feedback analysis, and brand reputation management. In this report, there will be a look at sentiment analysis by focusing on the analysis of movie reviews from the Internet Movie Database (IMDB).

The primary objective of this project is to develop and compare three different models for sentiment analysis: Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Recurrent Neural Network (RNN). By exploring and evaluating these models, their effectiveness and performance in predicting sentiment labels (positive or negative) for IMDB movie reviews will be determined.

To do this, a dataset is ued containing a collection of IMDB-sourced movie reviews. This data set includes various reviews that span various genres, languages, and sentiments. Furthermore, the dataset will first be preprocessed and already has the positive or negative labels, enabling us to train and evaluate the models effectively.

This report is structured as follows. First, an overview of related work and existing research is provided in the field of sentiment analysis. it then goes into the details of the IMDB movie review dataset, including its size, characteristics, and the preprocessing steps employed. Next, the methods used for each of the three models: BoW, TF-IDF, and RNN will be explained. The experimental setup, results, and a comprehensive analysis of the results obtained from each model are presented. Finally, a discussion and conclusion of the findings.

2 Related work

Numerous studies have been conducted in the field of sentiment analysis, exploring different methods to extract sentiment information from textual data. The Bag-of-Words (BoW) model has been widely used, representing text as a collection of words to capture sentiment patterns. The Term Frequency-Inverse Document Frequency (TF-IDF) model considers term frequency and relevance across the dataset, highlighting important discriminative terms. Deep learning models, such as Recurrent Neural Networks (RNNs), have gained attention for their ability to capture context and long-term dependencies in sentiment analysis. Other research has focused on feature extraction techniques, sentiment lexicons, and machine learning algorithms like Support Vector Machines (SVM) and Logistic Regression. These previous works provide valuable insights and form the foundation for this project in sentiment analysis.

3 Dataset

Next a section discussing the dataset and the preprocessing of the dataset. The dataset containing movie reviews gathered from IMDB used in this project consists of two columns: the "review" column containing the text of the reviews and the "sentiment" column indicating whether the review is classified as positive or negative. We examined the distribution of sentiment labels and found an even distribution between positive and negative sentiments.

Before proceeding with the classification task, we performed a series of preprocessing steps on the reviews. The following operations were applied:

1. Lowercasing: All text was converted to lowercase to ensure case-insensitive analysis.

2. HTML tag removal: Any HTML tags present in the reviews were removed, as they do not

contribute to the sentiment analysis.

3. Removal of parentheses and their contents: Parentheses and the text enclosed within them were eliminated to focus solely on the review text.

4. Contraction expansion: Contractions such as "n't" or "'ll" were expanded to their full forms (e.g., "can't" to "cannot", "I'll" to "I will") to avoid any potential loss of meaning.

5. Removal of commas between numbers: Commas occurring between numbers were removed to maintain numerical consistency.

6. Removal of possessive forms: Possessive forms such as "'s" were eliminated to simplify the text representation.

7. Replacement of the symbol "&" with the word "and": This replacement was made for clarity and to maintain consistency in the text.

8. Removal of numbers and decimal numbers: Numbers and decimal numbers were removed, as they are not likely to contribute significantly to sentiment analysis.

9. Reduction of repeated periods: Instances of repeated periods were reduced to a single period, ensuring proper sentence structure.

10. Removal of special characters and punctuation marks: Special characters and punctuation marks that do not carry essential semantic information were removed.

11. Stop word removal and lemmatization: Stop words, which are commonly occurring words with limited semantic value, were removed. The remaining words were lemmatized to reduce inflectional forms to their base or dictionary form.

To gain further insights into the reviews, we analyzed the distribution of bigrams, which are consecutive pairs of words. This revealed some interesting patterns that provide clues about the sentiment expressed in the reviews. For instance, bigrams such as "worst movie" and "good movie" clearly indicate the sentiment associated with these phrases.

By completing the dataset preprocessing and transforming the sentiment labels, we have prepared the reviews for further analysis and classification.

4 Methodology

In this section, the methodology employed for sentiment analysis on the preprocessed dataset of IMDB movie reviews is outlined. Our goal was to develop accurate models that could effectively classify the sentiment of the reviews as either positive or negative. We utilized three different approaches: Bag-of-Words (BoW), TF-IDF, and Recurrent Neural Network (RNN).

4.1 BoW

To begin, we needed to convert the text data into a format suitable for analysis. We achieved this by using a technique called the Bag-of-Words model. This model represents each document (movie review in our case) as a collection of words, disregarding grammar and word order.

To create the Bag-of-Words representation, we first constructed a vocabulary. The vocabulary is essentially a dictionary that contains all unique words present in the movie reviews. This step involved scanning through the reviews, identifying distinct words, and assigning them unique identifiers or indices.

After building the vocabulary, we applied it to transform the movie reviews into numerical feature vectors. Each review was converted into a vector, where each element represented the frequency or presence of a specific word from the vocabulary. This resulted in a matrix where the rows corresponded to the individual reviews, and the columns represented the words in the vocabulary.

4.2 TF-IDF

Next the TF-IDF model. TF-IDF is a commonly used approach in natural language processing that aims to capture the importance of words in a document by assigning weights to them. It calculates a numerical value for each word based on two factors: term frequency (TF) and inverse document frequency (IDF).

Term Frequency (TF) measures the frequency of a word in a document. It assumes that the more frequent a word is in a document, the more important it is for understanding the content. Words that appear more frequently will have higher TF values.

Inverse Document Frequency (IDF) calculates the rarity of a word across the entire collection of documents. It assigns higher weights to words that appear less frequently in the corpus. Rare words

are considered more informative and carry more significance.

The TF-IDF score is computed by multiplying the TF and IDF values for each word. This results in a representation where words that are both frequent within a document and rare across the corpus receive higher scores.

4.3 Logistic regression

Both BoW and TF-IDF use logistic regression as a classification algorithm The logistic regression involves training a classification model to predict the sentiment of text data. It begins by creating an instance of the logistic regression algorithm and fitting it to the training data, which consists of the transformed features (BOW or TF-IDF) and their corresponding sentiment labels. In essence, logistic regression is a key component in sentiment analysis as it allows for the classification of text data into positive or negative sentiment categories.

4.4 RNN

Finally we implemented a RNN as follows: First, the text data is tokenized into individual words or tokens. This is done to represent the text as a sequence of discrete units that the RNN can process. Next, the vocabulary size is calculated based on the unique tokens in the tokenized text data. The maximum sequence length is determined by finding the length of the longest tokenized sentence.

The RNN model architecture is defined using the Keras Sequential API. It consists of an embedding layer, which maps each word in the input sequence to a dense vector representation. This is followed by a GRU (Gated Recurrent Unit) layer, which captures the sequential dependencies in the data. Finally, a dense layer with a sigmoid activation function is added to produce the binary sentiment prediction.

The text data is then preprocessed using the Tokenizer class, which converts the text into sequences of numerical indices based on the learned vocabulary. Padding is applied to ensure all sequences have the same length.

The model is compiled with the binary cross-entropy loss function and the Adam optimizer. It is then trained on the training data, with validation data used for evaluating the model's performance during training.

After training, the model makes predictions on the training, validation, and test data. The predicted probabilities are thresholded at 0.5 to obtain the binary sentiment predictions.

5 Testing & Results

Now let's go over the different results.

5.1 Testing

To summarise the performance of all different models we make a classification, report which summarizes the performance of a classification model on the test dataset. The report includes several metrics such as precision, recall, and F1-score for each class (0 negative and 1 positive in this case), as well as the support, which represents the number of samples in each class. The precision metric measures the proportion of correctly predicted positive (or negative) instances out of the total predicted positive (or negative) instances. Recall, also known as sensitivity, measures the proportion of correctly predicted positive (or negative) instances out of the total actual positive (or negative) instances.

The F1-score is the harmonic mean of precision and recall, providing a single value that balances both metrics.

5.2 Results

First the result of the BoW: Next the result of the

```
Training score: 0.9966968867059547
Validation score: 0.8778542662083648
Testing score: 0.9727862901383451
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     19815
           1       0.97      0.97      0.97     19908

    accuracy                           0.97     39723
   macro avg       0.97      0.97      0.97     39723
weighted avg       0.97      0.97      0.97     39723
```

TF-IDF:
Finally the RNN, This does not have exact results

```
Training score: 0.9338284055627014
Validation score: 0.8906389285063948
Testing score: 0.9259520105595895
              precision    recall  f1-score   support

           0       0.93      0.92      0.93     19815
           1       0.92      0.94      0.93     19908

    accuracy                           0.93     39723
   macro avg       0.93      0.93      0.93     39723
weighted avg       0.93      0.93      0.93     39723
```

because of the random initialization of weights and biases. So these are values the results fluctuate around:

```
Training score: 0.9833078405039445
Validation score: 0.8867371008605168
Test score: 0.9526959787276394
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     19815
           1       0.96      0.94      0.95     19908

    accuracy                           0.95     39723
   macro avg       0.95      0.95      0.95     39723
weighted avg       0.95      0.95      0.95     39723
```

6 Discussion

In terms of training scores, the BoW model achieves a score of 0.997, indicating excellent performance in learning from the training data. The TF-IDF model obtains a slightly lower score of 0.934, while the RNN model achieves a score of 0.983. This suggests that the BoW and RNN models perform better in capturing the patterns and relationships in the training data.

Moving to the validation scores, the BoW model achieves a score of 0.878, indicating good generalization capability. The TF-IDF model obtains a slightly higher score of 0.891, while the RNN model achieves a similar score of 0.887. This implies that the TF-IDF and RNN models perform slightly better in generalizing to unseen data during the validation phase.

For the testing scores, the BoW model achieves an impressive score of 0.973, indicating high accuracy in sentiment classification on the test dataset. The TF-IDF model obtains a score of 0.926, while the RNN model achieves a score of 0.953. This suggests that the RNN model performs slightly better than the other two models in predicting sentiment labels on the test data.

Analyzing the precision, recall, and F1-scores for both positive and negative sentiments, the BoW and TF-IDF models exhibit similar performance, with scores around 0.93 to 0.97. The RNN model also achieves comparable scores. These results suggest that all three models are capable of effectively classifying sentiments.

In summary, while the BoW model achieves the highest accuracy on the testing data, the TF-IDF and RNN models offer competitive performance in terms of precision, recall, and F1-scores. The choice of the most suitable model would depend on specific requirements, such as the dataset characteristics, computational resources, and the trade-off between accuracy and computational complexity.

all models seem to struggle with a particular review where they misclassify the sentiment because they fail to recognize the context of words being used. Despite the reviewer stating that they think the movie is good, the models focus on phrases like "movie bad" without considering the negation and overall sentiment expressed. This results in incorrect predictions of negative sentiments.

In summary, both the RNN and TF-IDF models face challenges in understanding the context of words used in comparisons or as recommendations. They may misclassify sentiments when the reviewer expresses conflicting opinions or when comparing the movie to other installments. The BoW model also struggles with recognizing the context of words, especially after extensive text cleaning. These limitations indicate that the models have difficulty comprehending the subtleties of language and the reviewer's intent, leading to incorrect predictions of sentiment in certain cases.

7 Conclusion

all models face challenges in understanding the context of words, leading to misclassification of sentiments in certain cases. They struggle with recognizing negation, conflicting opinions, and comparisons to other movies or installments. The BoW model also faces difficulties in recognizing context after extensive text cleaning.

These limitations indicate the models' difficulty in comprehending language subtleties and the reviewer's intent. Improvements are necessary to enhance their performance in sentiment analysis, such as incorporating semantic understanding and domain-specific knowledge.

While the BoW model demonstrates the highest accuracy, the choice of the most suitable model depends on specific requirements and dataset characteristics In summary, this study highlights the challenges faced by sentiment analysis models and emphasizes the importance of continued research and advancements to improve their ability to accurately interpret sentiments and understand the complexities of human language.