Sentiment Analysis on Movie Reviews

https://github.com/mchabrol/NLP-sentiment-analysis

Suzie Grondin
ENSAE Paris
suzie.grondin@ensae.fr

Abstract

Sentiment analysis is a classical task in natural language processing, aimed at classifying textual data as positive or negative. This study compares traditional machine learning techniques, such as TF-IDF with Naïve Bayes, and advanced word embeddings, such as Word2Vec, with state-of-the-art transformer models like RoBERTa for binary sentiment classification of movie reviews. Using the IMDB dataset, we show that while classical models perform well, transformer-based models significantly outperform them, achieving an accuracy of 94.87% compared to 84.06% for TF-IDF.

1 Introduction

Sentiment analysis consists in identifying and classifying subjective opinions expressed in text as positive or negative. It has been widely studied in natural language processing, with early approaches relying on classical machine learning techniques and hand-crafted features.

One of the most influential works in the field was presented by Maas et al. (2011) in their paper *"Learning Word Vectors for Sentiment Analysis"* [3]. Their study introduced a novel method for training word representations that explicitly capture both semantic and sentiment information. Their model, inspired by Latent Dirichlet Allocation (LDA), was designed to capture both semantic and sentiment information by leveraging labeled movie review data. Using a joint optimization framework, they trained word representations that improved sentiment classification. Evaluated on multiple datasets, including the Pang and Lee (2004) corpus and a new 50,000-review IMDB dataset, their method outperformed traditional approaches like TF-IDF, LDA, and Latent Semantic Analysis (LSA). With a peak accuracy of **88.90%**, their sentiment-aware embeddings proved highly effective for polarity classification, demonstrating the benefits of integrating sentiment supervision into word representations.

However, the field has evolved substantially since the original dataset's release, with transformative developments in embeddings (e.g., word2vec, GloVe, FastText) and large language models (e.g., BERT, RoBERTa, GPT-based architectures). These advancements suggest significant potential for further improvement in classification accuracy.

This article explores precisely this possibility: can recent advances in embedding techniques and large-scale pre-trained language models lead to improved accuracy on sentiment analysis tasks for movie reviews? To address this, we propose an experimental comparison between classical embedding methods and contemporary large language models.

2 Problem Presentation

The specific NLP task we address is binary sentiment classification of movie reviews. Given a textual review, the goal is to classify it accurately as either positive or negative. Despite the apparent simplicity of this binary classification, sentiment analysis remains challenging due to linguistic nuances such as sarcasm, irony, and complex sentence structures that affect the interpretation of sentiment.

To evaluate the effectiveness of current models, we use the IMDB movie reviews dataset. This dataset comprises 50,000 reviews, evenly split into positive and negative sentiment categories. Prior research, notably by Maas et al. [3], has established benchmark results against which modern techniques can be assessed.

Performance Metrics and Evaluation The models were evaluated using **accuracy** as the primary performance metric. Accuracy is defined as the proportion of correctly classified instances out of the total number of instances:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{1}$$

This metric is widely used in classification tasks, especially in balanced datasets, because it provides a simple and interpretable measure of overall model performance. Note that we can use it in our case because the dataset is made of 50% of positive and 50% of negatives evaluations.

3 Data Preparation and Visualization

We worked with the IMDB dataset, consisting of movie reviews labeled into two categories: positive and negative. Each review includes a rating from 1 to 10 and a textual comment.

To preprocess the data, we applied a basic cleaning procedure, which mainly involved removing HTML tags (specifically `<br />`) from the reviews and eliminating common words (such as "I," "then," "and," etc.) that do not contribute to prediction. Positive and negative reviews were extracted separately and labeled (positive: 1, negative: 0), then combined into a balanced dataset of 25,000 reviews per class.

To better understand the lexical distribution within the reviews, we generated word clouds grouped by movie rating (from 1 to 10) and by sentiment class (positive or negative). Figure 1 shows the notable differences in vocabulary across ratings. Lower ratings (e.g., rating 1) prominently feature negative terms such as *bad*, *worst*, and *boring*, whereas higher ratings (e.g., rating 10) highlight positive terms such as *great*, *love*, and *best*. Figure 2 clearly illustrates the contrast in vocabulary between overall positive and negative reviews, emphasizing words indicative of sentiment.

Additionally, we conducted a focused analysis on the word *no*, often ambiguous in interpretation, revealing significant frequency across both review types (Figure 3), with a slightly higher presence in very negative reviews. This highlights the need for more contextual analysis to improve model accuracy.

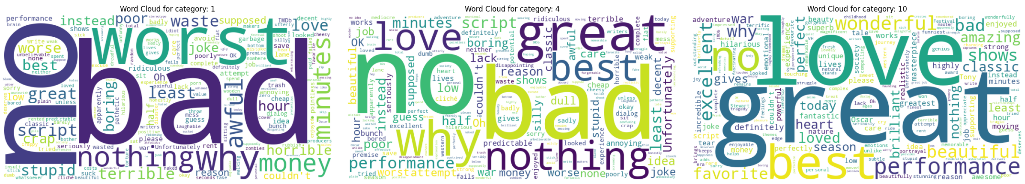

Figure 1: Word clouds for ratings: 1 (left), 5 (middle), and 10 (right).

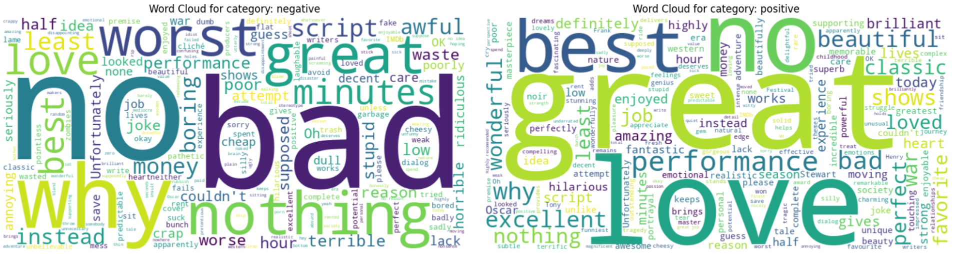

Figure 2: Word clouds for negative (left) and positive (right) reviews.

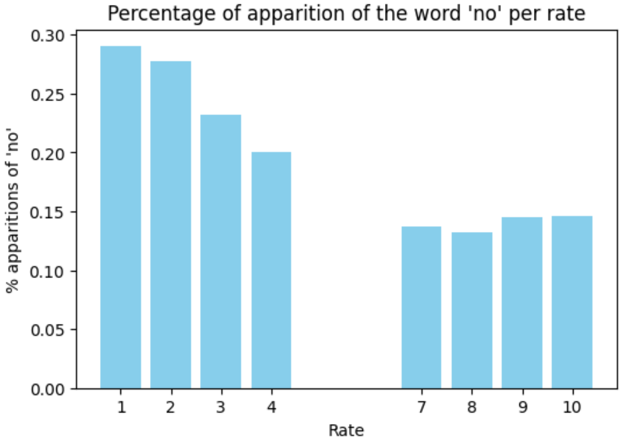

Figure 3: Percentage occurrence of the word *no* by rating.

These preliminary analyses facilitate a qualitative understanding of the corpus, providing a solid foundation for subsequent experiments with advanced embedding and language models.

4 Chosen models

To assess the effectiveness of different sentiment analysis techniques on movie reviews, we adopted a progressive approach, starting from a simple baseline and advancing towards more sophisticated deep learning models. This methodological progression enables us to evaluate the impact of increasingly complex embedding techniques and architectures on classification accuracy.

4.1 Baseline: TF-IDF and Traditional Machine Learning

Our initial approach relies on a classical text vectorization technique: Term Frequency-Inverse Document Frequency (TF-IDF) [5]. TF-IDF represents textual data in a sparse, high-dimensional space, capturing the relative importance of words within a document while mitigating the influence of overly common terms.

To classify sentiment, we trained a Support Vector Machine (SVM) on the TF-IDF representations of the movie reviews. This baseline provides a strong reference point by leveraging well-established statistical features without requiring pre-trained embeddings.

However, despite its interpretability and computational efficiency, TF-IDF suffers from several limitations. Notably, it does not capture semantic relationships between words, treating each term independently. Additionally, it remains sensitive to vocabulary variations, limiting its generalization capacity.

We conducted experiments by varying three key hyperparameters:

- **Maximum number of features** (`max_features`): The vocabulary size used for TF-IDF vectorization, tested with values of 1000, 2000, and 5000.
- **Use of inverse document frequency weighting** (`use_idf`): A boolean parameter determining whether to apply IDF scaling.
- **Smoothing parameter for Naïve Bayes** (`alpha`): The Laplace smoothing parameter, tested with values of 0.1, 1.0, and 10.0.

The best hyperparameter configuration was determined based on validation accuracy.

4.2 Word Embeddings: Word2Vec and SVM

To overcome the limitations of TF-IDF, we explored word embeddings, specifically Word2Vec, which maps words into continuous vector spaces where semantic similarities are preserved. Unlike TF-IDF, Word2Vec captures word relationships by leveraging co-occurrence patterns within large corpora [4].

We trained a Word2Vec model on our dataset to obtain dense word representations, then aggregated word vectors to obtain review-level embeddings. This was achieved by averaging the embeddings of all words in a given review, creating a fixed-size representation.

These embeddings were then used as input features for an SVM classifier. While this approach significantly improves upon TF-IDF by incorporating semantic relationships, it still has notable drawbacks:

- **Loss of Context:** Word2Vec generates context-independent embeddings, meaning that words with multiple meanings (e.g., "bad" in different contexts) are assigned a single representation.
- **Insensitivity to Irony and Negation:** Since word embeddings are averaged at the review level, crucial nuances such as irony, negation, or sentiment shifts within a sentence may be lost.

4.3 Contextual Embeddings: RoBERTa and Neural Networks

To overcome the limitations of traditional models like TF-IDF and static word embeddings such as Word2Vec, we explored transformer-based models that generate contextualized word embeddings. These embeddings adapt based on the context in which a word appears, making them far more effective for tasks like sentiment analysis, where the meaning of words often depends on surrounding phrases.

For instance, take the sentence *"The bank will not approve the loan because it is too risky."* A traditional model like Word2Vec might struggle to determine whether the word "bank" refers to a financial institution or a riverbank, since it assigns the same embedding to a word regardless of context. Transformer-based models identifies that "bank" likely refers to a financial institution due to the presence of words like "approve" and "loan."

BERT [1] is built on the Transformer encoder architecture and is pre-trained on large text corpora using two objectives: Masked Language Modeling (MLM), where certain tokens in the input are hidden and the model learns to predict them, and Next Sentence Prediction (NSP), where the model predicts whether two sentences appear consecutively in the original text. The bidirectional design of BERT allows it to capture the context on the left and right of a word simultaneously, providing deep semantic understanding.

While BERT was a major breakthrough in natural language processing, it has some limitations. The NSP task was later shown to be less beneficial than originally thought, and the static masking of tokens during training reduced the diversity of learning signals.

To address these issues, we used RoBERTa (Robustly Optimized BERT Pretraining Approach) [2], a more advanced variant of BERT introduced by Facebook AI. RoBERTa retains the same underlying architecture as BERT but introduces several key improvements:

- It removes the Next Sentence Prediction task entirely.
- It uses dynamic masking, changing which tokens are masked at each training epoch to improve generalization.

- It is trained on significantly larger datasets for longer durations, using larger batch sizes and higher learning rates.

These modifications make RoBERTa more robust and better suited for downstream tasks like sentiment classification.

RoBERTa proved especially effective on reviews with complex structures, irony, or negation. For example, in the sentence *"I didn't expect to enjoy this movie, but I absolutely loved it"*, earlier models might misclassify it due to the negation. RoBERTa, thanks to its deep contextual understanding, correctly identifies the positive sentiment.

For our final sentiment classification model, we used the pre-trained `roberta-base` architecture via the `RobertaForSequenceClassification` class from Hugging Face's Transformers library. This model is designed for sequence classification tasks and was adapted to perform binary sentiment prediction.

5 Results

We evaluated three different models for the sentiment classification task described above: a simple TF-IDF + Naïve Bayes baseline, a Word2Vec + SVC model, and a fine-tuned RoBERTa transformer.

For the TF-IDF + Naïve Bayes approach, we tested several parameter combinations and found that the best results were achieved using a maximum of 5,000 features, enabling IDF weighting, and setting the smoothing parameter `alpha` to 1.0. This configuration provided a good trade-off between accuracy and generalization.

As for the RoBERTa model, we also ran multiple experiments to tune its training settings. The final configuration used a learning rate of 1×10^{-5} and trained the model for 3 epochs. To simulate a larger effective batch size and improve training stability, we applied gradient accumulation over 4 steps. We used the AdamW optimizer, which is well-suited for transformer-based models, along with a linear learning rate scheduler without warm-up. The batch size was set to 8.

The performance of all three models is summarized in the table below:

Model	**Train Accuracy (%)**	**Test Accuracy (%)**
TF-IDF + Naïve Bayes	86.50	84.06
Word2Vec + SVC	87.74	86.73
RoBERTa (fine-tuned)	97.76	94.87

Table 1: Train and test accuracy of the three models

6 Conclusion

In this study, we explored the evolution of sentiment analysis models by comparing classical machine learning techniques with modern deep learning approaches. Beginning with traditional TF-IDF representations and Naïve Bayes classifiers, we established a strong baseline for sentiment classification on the IMDB dataset. We then demonstrated that word embeddings such as Word2Vec provide richer feature representations, yielding improved classification performance. Finally, by leveraging contextualized embeddings from transformer-based models like BERT, we achieved state-of-the-art accuracy, highlighting the importance of context-aware representations in sentiment analysis.

Our results indicate a clear trend: as embedding techniques and model architectures become more sophisticated, sentiment classification accuracy improves. The transition from static word vectors to contextual embeddings allows for better handling of linguistic nuances such as sarcasm, negation, and polysemy. While BERT significantly outperformed previous models, it comes with computational costs that may not always be feasible for real-world applications. Future work could focus on optimizing transformer models for efficiency or exploring hybrid approaches that balance performance with computational feasibility. Ultimately, our findings reinforce the value of recent advances in NLP for sentiment analysis and open avenues for further research in explainability and domain adaptation.

References

[1] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018). URL: `https://arxiv.org/abs/1810.04805`.

[2] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692* (2019). URL: `https://arxiv.org/abs/1907.11692`.

[3] Andrew Maas et al. "Learning word vectors for sentiment analysis". In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 2011, pp. 142–150.

[4] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[5] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5 (1988), pp. 513–523.

A Appendix / supplemental material

Contributions

This project was conducted collaboratively by Suzie Grondin and Marion Chabrol. The choice of models to test (TF-IDF, Word2Vec, and RoBERTa) was made jointly after discussion. About code architecture, Marion organized the overall code and repository structure.

Both authors participated in the data analysis. Suzie mainly worked on the wordcloud visualizations, while Marion conducted the focused analysis of specific ambiguous words.

They also both contributed to the implementation of the TF-IDF + Naïve Bayes model: Suzie implemented the main classification pipeline, while Marion contributed to code cleaning and the selection of hyperparameters. For the other models, Marion was primarily responsible for the implementation of the Word2Vec + SVM model, while Suzie focused on fine-tuning the RoBERTa transformer. Nonetheless, each of them contributed to both parts, reviewing and refining the work collaboratively.