Sentiment analysis on movie reviews

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

Marion Chabrol[*]
ENSAE Paris
marion.chabrol@ensae.fr

Abstract

This report investigates sentiment analysis on movie reviews using the IMDB dataset. We compare three modeling approaches—TF-IDF with Naïve Bayes, Word2Vec with SVM, and a fine-tuned RoBERTa transformer—to evaluate the impact of increasingly complex representations. Our results show a clear performance improvement with model sophistication, with RoBERTa achieving the best accuracy.

1 Introduction

Problem description Sentiment analysis is a classical task in Natural Language Processing (NLP) that involves determining the opinions expressed in text, typically categorized as positive, negative, or neutral. It is widely used in applications such as customer feedback analysis, brand monitoring, and recommendation systems.

In this report, we focus on binary sentiment classification, a sub-task where the goal is to classify if a movie review is expressing a positive or negative sentiment. This task poses the challenge of understanding negation, implicit sentiment but also sarcasm.

To investigate how recent advances in NLP can help address these challenges, we conduct an empirical study on the IMDB movie reviews dataset. We compare the effectiveness of three different approaches: a traditional TF-IDF + Naïve Bayes pipeline, a Word2Vec + SVM model, and a transformer-based architecture using RoBERTa. Through this comparison, we aim to quantify the improvements brought by contextualized language models.

Performance metrics and evaluation The models were evaluated using accuracy as the primary performance metric. We compute accuracy as follows:

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

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.

2 State of the art

Classical approaches Initial sentiment analysis systems relied heavily on manually engineered features and lexicons. Bag-of-words (BoW) models, often combined with classifiers such as Naïve

[*] The individual contributions to this project are detailed in the Appendix (see Section 6).

Bayes or Support Vector Machines (SVM), formed the baseline methodology for over a decade. These models treat each word independently and fail to capture semantic relationships or word order. TF-IDF (Term Frequency–Inverse Document Frequency) was introduced to weigh words by their relative importance, reducing noise from frequently occurring but sentiment-neutral terms.

Despite their limitations, these methods remain relevant as simple, interpretable baselines. Their ease of implementation and low computational cost make them attractive for many practical applications.

Distributional semantics and word embeddings The advent of word embeddings significantly improved text representation. Instead of sparse, high-dimensional vectors, models like Word2Vec[4] and GloVe[5] map words to dense, low-dimensional vectors that encode semantic similarity. This innovation allowed classifiers to generalize better by capturing latent relationships between terms.

A notable application of these ideas is presented by Maas et al.[3], who proposed 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. This method was evaluated on multiple datasets, including the Pang and Lee (2004) corpus and a new 50,000-review IMDB dataset. Achieved an accuracy of 88.90% on the IMDB dataset, their method outperformed traditional approaches like TF-IDF, LDA, and Latent Semantic Analysis (LSA).

However, a critical limitation of these embeddings is their static nature: each word has a single vector regardless of context. For example, the word *"cold"* could describe weather or a personality trait, and these meanings should ideally be distinguished in sentiment tasks.

Contextualized embeddings and transformer models To overcome the limitations of static embeddings, context-sensitive language models were introduced. BERT (Bidirectional Encoder Representations from Transformers)[1] marked a major step forward by generating context-dependent word vectors through masked language modeling and bidirectional attention mechanisms. BERT captured the left and right contexts of words, enabling a deeper understanding of syntax and semantics.

RoBERTa[2], an enhancement to BERT, optimized the training process by eliminating the next-sentence prediction goal, using dynamic masking and training for longer on a larger corpus. These improvements led to significant gains in accuracy in NLP benchmarks, including sentiment classification.

Today, transformer-based models represent the state of the art in sentiment analysis.

3 Data analysis

The IMDB dataset used in this study consists of 50,000 movie reviews labeled as either positive or negative. The dataset is balanced, containing 25,000 reviews in each category, with each review associated with a rating from 1 to 10.

To prepare the data for modeling, we applied a cleaning pipeline that includes : removal of HTML tags (e.g., `<br />`), lowercasing and elimination of common stopwords. This preprocessing step ensures consistency which is important for classical models that are sensitive to vocabulary variations.

We conducted an initial exploration of the dataset in order to draw some insights.

First, we created word clouds grouped according to movie rating (from 1 to 10) and sentiment label (positive or negative) to visually assess the frequency distribution of words. Poorly rated reviews (e.g., 1 or 2) included terms such as *bad* and *worst*, while highly rated reviews (e.g., 9 or 10) were dominated by positive terms such as *great*, *love* and *best*. Figure 1 and Figure 2 show these visualizations.

We also examined the distribution of some ambiguous terms. For example, the word *no*, which can signal negation or simply appear in a neutral context, was analyzed for all rating levels. We wanted to see if the occurrence was more frequent in the worst ratings. Figure 3 shows that its appearance is more frequent in poorly rated reviews, but that the word 'no' remains regularly used even in positive reviews. This reinforces the idea that we need models that understand the contextual use of words, rather than relying on isolated frequency counts.

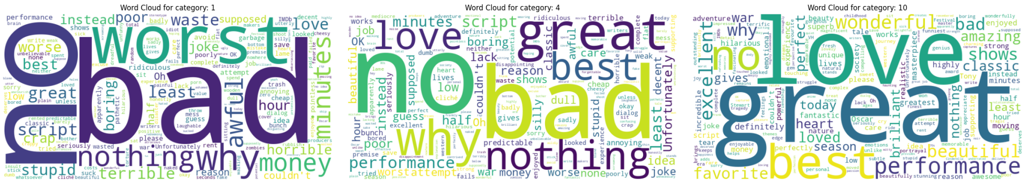

Figure 1: Word clouds for ratings: 1 (left), 5 (middle), and 10 (right).

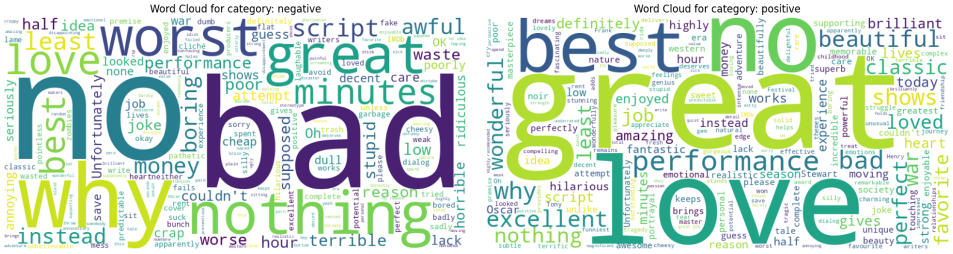

Figure 2: Word clouds for negative (left) and positive (right) reviews.

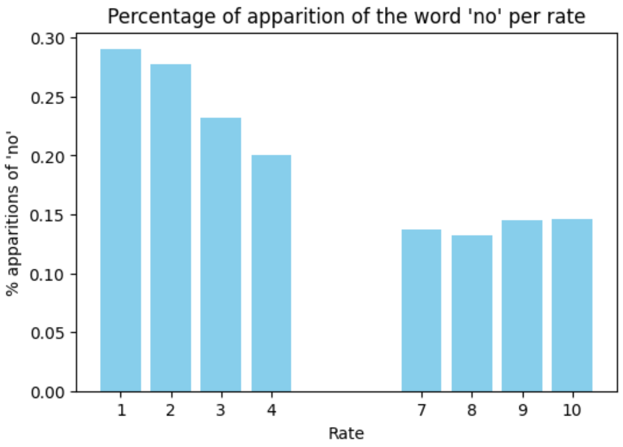

Figure 3: Percentage occurrence of the word *no* by rating.

4 Proposed Models

To evaluate different sentiment classification strategies, we adopt a progressive modeling pipeline—from classical models with linear classifiers to pre-trained transformer-based architectures. Each stage reflects an increase in model complexity and capacity to capture semantic features in text.

4.1 TF-IDF and Naïve Bayes

As a baseline, we use the Term Frequency-Inverse Document Frequency (TF-IDF) representation, which transforms each review into a high-dimensional sparse vector based on word importance across the corpus [6]. We train a Naïve Bayes classifier on these vectors.

We tuned several hyperparameters and chose the best configuration based on validation accuracy:

- **max_features**: the maximum vocabulary size (tested with 1000, 2000, and 5000 words).
- **use_idf**: a boolean that specifies whether to apply inverse document frequency scaling (when enabled, frequent terms across all documents are downweighted).

- **alpha**: the Laplace smoothing parameter in Naïve Bayes (used to avoid zero probabilities for unseen words).

The best results were achieved with: a maximum of 5,000 features, enabling IDF weighting, and setting the smoothing parameter `alpha` to 1.0.

Although computationally efficient and interpretable, TF-IDF models fail to capture word semantics or contextual nuances, as said earlier. Nevertheless, it is a good baseline for the rest of the report.

4.2 Word2Vec embeddings and SVM

To enhance semantic representation, we replace sparse TF-IDF vectors with dense Word2Vec embeddings [4]. A Word2Vec model is trained on the IMDB corpus, and each review is encoded by averaging the vectors of the words composing the sentence.

These aggregated embeddings are then passed to a Support Vector Machine (SVM) for classification. This approach improves generalization by capturing word similarity but remains context-independent.

The main limitations of this approach are as follows: firstly, the same vector is used regardless of usage (a word with multiple meanings therefore has a single representation) and, secondly, averaging all vectors in the end eliminates word order and we may lose sensitivity to negation or irony.

4.3 RoBERTa: transformer-based classification

To address the shortcomings of prior models, we fine-tune a pre-trained transformer model, RoBERTa [2], using the Hugging Face `RobertaForSequenceClassification` implementation.

Unlike static embeddings, RoBERTa generates contextualized representations by encoding the full sentence structure using attention mechanisms. It builds on BERT [1] but eliminates the Next Sentence Prediction objective, incorporates dynamic token masking, and trains on a significantly larger dataset.

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 learning stability, we applied gradient accumulation over 4 epochs. We used the AdamW optimizer, which is well suited to transformer-based models, and a linear learning rate planner without warm-up. The batch size was set at 8.

This model is especially suited for capturing subtle linguistic phenomena, such as sentiment inversion due to negation (e.g., *"The plot was dull at first, yet the ending was unexpectedly powerful"*).

5 Results

We evaluated the three proposed models—TF-IDF + Naïve Bayes, Word2Vec + SVM, and fine-tuned RoBERTa—on the binary sentiment classification task using the IMDB dataset. Their performances are summarized in Table 1.

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

Table 1: Train and test accuracy of the three models

The results show a clear improvement in classification accuracy as we move from classical models to more sophisticated ones. The TF-IDF + Naïve Bayes model, while simple and efficient, performs the weakest, which aligns with its inability to understand contextual or semantic nuances in the reviews. Nevertheless, it remains a solid baseline, offering competitive performance for its simplicity.

The Word2Vec + SVM model shows a modest gain over TF-IDF, highlighting the value of semantic similarity encoded in word embeddings. However, the fact that it still treats word meaning independently of context limits its potential. For example, negation or contrastive conjunctions ("but", "although") are not appropriately handled, as word averaging discards syntactic structure and word order.

RoBERTa significantly outperforms the other models, achieving nearly 95% test accuracy. This model's superiority is attributable to its deep contextual understanding, enabling it to handle complex sentiment structures such as irony. Its performance demonstrates that pre-trained transformer models not only excel in expressiveness but are also capable of fine-tuning effectively on domain-specific sentiment tasks.

6 Conclusion

In this report, we explored and compared three different approaches to sentiment classification on the IMDB movie reviews dataset, each representing a different stage in the evolution of natural language processing methods: classical feature-based models, word embeddings, and contextual transformer-based architectures.

Our findings show a clear trajectory of improvement as model complexity increases. The TF-IDF + Naïve Bayes baseline offered reasonable performance for a simple solution but struggled with semantic nuances. Word2Vec embeddings improved upon this by capturing word similarity, though they remained limited by their inability to model context. Ultimately, RoBERTa achieved the best results, benefiting from its ability to understand language in context and model complex linguistic structures.

This experiment confirms the superiority of modern transformer architectures in sentiment analysis tasks, especially when dealing with nuanced and diverse datasets. However, they are also more resource-intensive.

Future work could explore lighter transformer variants (e.g., DistilBERT), or techniques for interpretability.

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] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

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

Appendix

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.