Sentiment analysis on movie reviews

Mekki Lila
3rd year DSSA
ENSAE Paris
lila.mekki@ensae.fr

Abstract

Sentiment analysis is an paramount task in natural language processing, with the objective of discerning subjective information from textual data. Building upon the work of Maas et al. (2011), which introduced a model combining unsupervised and supervised techniques to learn word vectors with a sentiment component, we use the IMDb movie reviews dataset to replicates their methodology and establish a performance baseline. We later on explore contemporary approaches, including leveraging modern methods for tokenization with pretrained BERT embeddings and fine-tuning neural network classifiers. Our experiments assess the impact of these modern techniques on sentiment classification accuracy.

1 Introduction

The computational study of opinions, sentiments, and emotions expressed in text, known as sentiment analysis, has significant prospects due to its wide-ranging applications in areas such as social media monitoring and customer feedback interpretation. Traditional approaches, including rule-based systems and classical machine learning models, such as support vector machines (SVMs), have laid the groundwork for this field.

The basic approach with word representation is based on embedding words as vectors in a high-dimensional space. However, this approach only captures the semantic relationships between words but does not encapsulate the sentiment that lies behind them. This approach gives powerful results in named entity recognition, or part of speech tagging, but is insufficient with sentiment analysis. That is why, in their seminal work, Maas et al. (2011) proposed a model that combines unsupervised and supervised learning to generate word vectors encapsulating both semantic and sentiment information. Evaluated on a large-scale database of movie reviews, their approach demonstrated notable improvements over existing methods at the time. It consists in learning the sentiment component via a supervised model. Hence, words expressing similar sentiments have similar vector representations. The vectors are thus fine-tuned with both semantic and sentiment information.

This project seeks to replicate the methodology of Maas et al. to establish a reliable performance baseline. Building upon this foundation, we investigate the efficacy of modern techniques, such as pre-trained tokenization using BERT embeddings or also the deployment of neural network classifiers. Our objective is to compare the performance of some modern methods with the original SVM-based model.

This report is structured as follows: section 2 details the dataset and descriptive analysis; section 3 reviews related work; section 4 outlines our proposed model architectures and implementation; section 5 presents our results and evaluations; finally, we conclude in section 6 with discussions and future directions.

2 Data Presentation and Analysis

We use the IMDb movie review dataset introduced by [1], which has become a standard benchmark for binary sentiment classification in natural language processing. The dataset contains 50,000 movie

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

reviews sourced from IMDb, evenly split into 25,000 training and 25,000 test examples. Each review is labeled as either positive or negative, with no neutral class, thus constituting a binary classification task. Originally labeled using a 0–10 star rating, the dataset was mapped into binary labels by the authors using a thresholding approach.

The training and test sets are balanced with equal numbers of positive and negative samples. Additionally, there is no overlap between the two sets in terms of movie titles, ensuring robust evaluation and minimizing the risk of data leakage. Each movie is restricted to a maximum of 30 reviews in the corpus to prevent skew due to prolific titles.

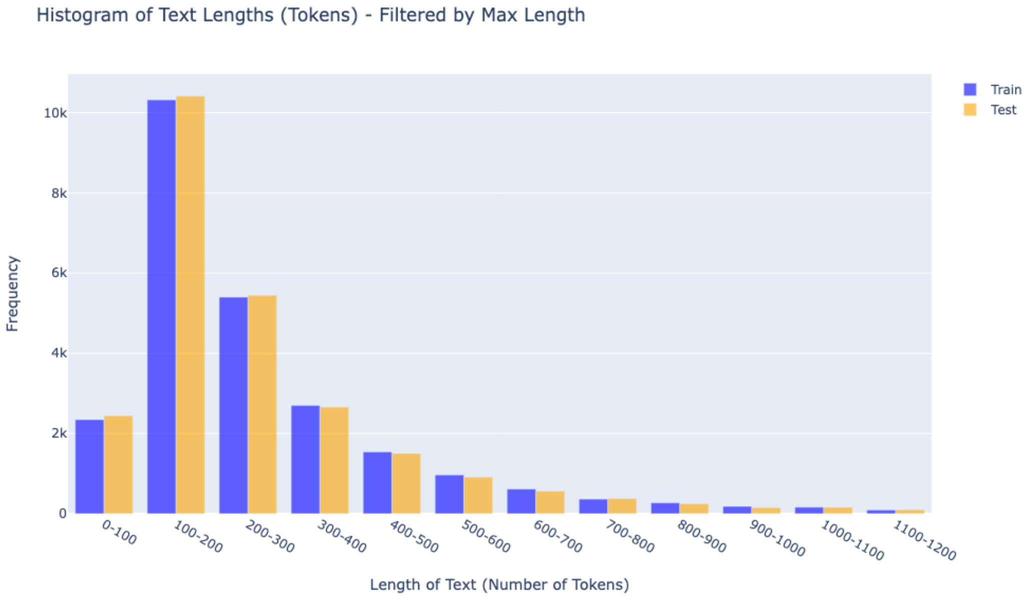


Figure 1: Distribution of review lengths (in tokens) in the IMDb dataset.

Figure 1 illustrates the distribution of review lengths in the dataset. The wide range of token counts across reviews has implications for model design, particularly in choosing padding strategies and sequence length truncation. Notably, the distributions are similar across training and test sets, which supports fair evaluation.

We further investigate whether sentiment correlates with review length. For instance, one might hypothesize that negative reviews are more concise, while positive reviews may elaborate on favorable aspects.

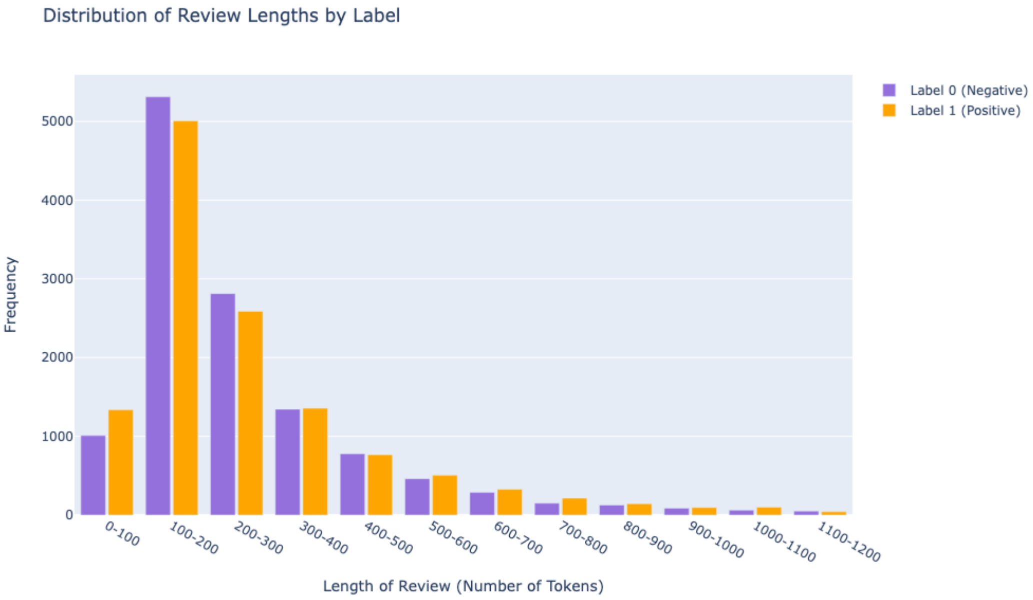


Figure 2: Distribution of review lengths (in tokens) by sentiment label.

As shown in Figure 2, very short reviews (under 100 tokens) are more likely to be positive. In the mid-length range (100–300 tokens), negative reviews are slightly more prevalent. For longer reviews, the length distribution is similar between positive and negative sentiments.

Each review is written in English and ranges from a few dozen words to several hundred. Although the original dataset includes an additional 50,000 unlabeled reviews for unsupervised learning, our work focuses exclusively on the labeled portion.

To illustrate the diversity of the dataset, we present examples of the shortest and longest reviews categorized by sentiment.

Shortest Reviews

Negative:

> *“I wouldn’t rent this one even on dollar rental night....”*
> **Length:** 11 tokens

Positive:

> *“Adrian Pasdar is excellent in this film. He makes a fascinating woman....”*
> **Length:** 14 tokens

Longest Reviews

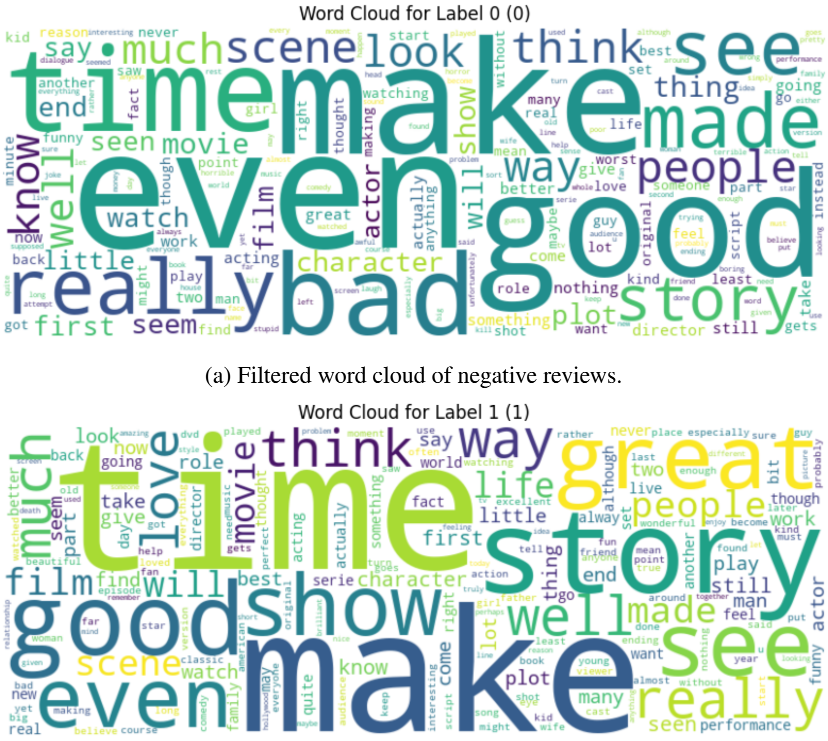
Negative:

> *“Some have praised _Atlantis:_The_Lost_Empire_ as a Disney adventure for adults. I don’t think so–at least not for thinking adults....”*
> **Length:** 1698 tokens (excerpt)

Positive:

> *“Match 1: Tag Team Table Match Bubba Ray and Spike Dudley vs Eddie Guerrero and Chris Benoit... (excerpt)”*
> **Length:** 2707 tokens (excerpt)

To gain additional insights into lexical patterns, we construct word clouds for each sentiment class. Prior to visualization, common stop words (e.g., “the”, “and”, “is”) and domain-specific neutral words (e.g., “movie”, “film”) are removed to better highlight sentiment-relevant vocabulary.


Word Cloud for Label 0 (0)

(a) Filtered word cloud of negative reviews.

Word Cloud for Label 1 (1)

(b) Filtered word cloud of positive reviews.

Figure 3: Comparison of word clouds for each sentiment class in the IMDb dataset.

While certain sentiment-bearing terms such as “bad” are more prominent in negative reviews, some expected polarity markers (e.g., “good”) appear in both classes, possibly due to negation or comparative context (e.g., “not good”, “was good but. . . ”).

In preprocessing, we apply standard text cleaning procedures including lowercasing, HTML tag removal, and tokenization. For classical approaches, we use a custom tokenizer; for transformer-based models such as BERT, we adopt pretrained tokenization schemes. This enables a comparative study of traditional and contextualized embedding methods.

3 Related work

The model we replicated and modified draws inspiration from several influential works in sentiment analysis and representation learning.

A pivotal contribution is the work by 1, who introduced a large-scale IMDb movie reviews dataset and proposed a novel method for learning word vectors that are sensitive to sentiment polarity. Their approach combined an unsupervised log-bilinear model for word embedding training with supervised classifiers such as SVMs to perform sentiment classification at the document level.

Unlike traditional word embeddings, which rely solely on local context, 1 incorporated document-level sentiment supervision during training, enabling their embeddings to capture sentiment-specific semantics. Their model treated document vectors as latent variables, estimated using variational inference, and showed that these representations outperformed earlier methods like Latent Semantic Analysis (LSA; 2) and Latent Dirichlet Allocation (LDA; 3), which, while effective for capturing thematic structure, struggled with polarity distinctions.

Subsequent developments in representation learning, including word2vec [4] and GloVe [5], introduced dense, static word embeddings that became standard in NLP tasks. These were often used in conjunction with neural network models—such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs)—to improve sentiment classification performance. However, these static embeddings do not adapt to context, limiting their effectiveness in capturing polysemy and sentiment shifts.

The introduction of transformer-based models, notably BERT [6], marked a significant shift in sentiment analysis. BERT leverages self-attention mechanisms to produce contextualized word representations, which dynamically adjust based on surrounding text. Fine-tuning such models on sentiment classification tasks has led to state-of-the-art results, outperforming traditional models by a significant margin.

In this project, we replicate the sentiment-aware embedding method proposed by [1] as a baseline and extend the analysis using modern NLP techniques. Specifically, we evaluate the effectiveness of contextual embeddings via pre-trained transformer models, explore simple neural architectures for fine-tuning, and assess the potential of attention-based mechanisms for further improvement. This comparison highlights how sentiment analysis has progressed from probabilistic latent models to large-scale, end-to-end deep learning systems.

4 Implementation and models

Our implementation follows a progression from traditional word representations to modern deep learning-based embeddings. The aim is to explore how different representations influence the performance of sentiment classification models on the IMDB dataset.

4.1 Tokenization and Vocabulary Construction

We began by implementing a custom tokenizer inspired by the method used by Maas et al. (2011), which involves lowercase, keeping the punctuation (since it is helpful for sentiment information), and basic whitespace tokenization. From this, a vocabulary of 5000 tokens was constructed based on term frequency (but skipping the 50 most frequent tokens), allowing for further experiments with Bag-of-Words and embedding-based approaches.

4.2 Word Representations

To compare different semantic representations of text, we experimented with:

Word2Vec : We trained a Word2Vec model with a CBOW (Continuous Bag of Word) setting which is the default and implies predicting the target instead of the context (which is the skip-gram approach). We used the unsupervised part of the dataset to train the model, setting the vector size to 50 (to have words embedded in a 50-dimensional space) with a contextual window of 5 tokens and a minimum frequency of 5 times per word. These settings were similar to those introduced in Maas et al. (2011).

LSA (Latent Semantic Analysis): This is the second word representation model implemented. We apply singular value decomposition to a term-document weighted and cosine normalized count matrix to uncover latent topics. This model explicitly learns semantic word vectors thanks to the co-occurrence matrix. We used 100 components for the SVD and normalized with the L_2 norm.

LDA (Latent Dirichlet Allocation): This is a probabilistic model that assumes each review is a mixture of latent topics. For each topic, the model learns a conditional distribution $p(w|T)$ for the probability that word w occurs in T. From this, we can obtain a 50-dimensional vector representation of words. In the original paper, they use the custom vocabulary extracted before but we used the scikit-learn LDA model, `CountVecotrizer` and finally normalized the word/topic distribution (that is, the number of times a word is generated by a topic) in order to get probabilities distributions.

BERT: This is here that our approach derives from that proposed in Maas et al. To compare the traditional word representations to more efficient one, we used a pre-trained BERT. BERT is a pre-trained transformer-based language model that understands the context of a word based on its surroundings. The B in BERT means Bidirectional. Instead of reading text left-to-right or right-to-left like previous models, BERT reads both directions at once, giving it a much deeper understanding of language. BERT relies on self attention mechanisms, to understand relationships between words in a sentence.

We believe that BERT could give better results than the previous word representations presented in the article of Maas et al., since it uses a self-attention mechanism to allow each token in a sentence to dynamically attend to all other tokens. More specifically, it is based on :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V \tag{1}$$

Here, Q, K and V represent the query, key, and value matrices respectively, and d_k is the dimension of the key vectors. This operation enables the model to weigh the importance of different words when encoding meaning. BERT further extends this mechanism via Multi-Head Attention.

This part of the model does not need labeled data and represents the semantic information of the model. However, it does not capture sentiment information. Indeed, it only captures semantic similarities and words appearing in similar context. It does not have a link with the polarity of the review. To add this sentiment part, we need to fine-tune the embeddings. This is the task of the supervised classification part.

4.3 Classification Models

For sentiment classification, we trained logistic regression models with the previous word embeddings based on different models. We also tried an approach based on neural networks to see the possible increase of performance that backpropagation and hidden layers could bring.

To reproduce the sentiment component of the original work by Maas et al. (2011), we implemented both a logistic regression classifier and a simple neural network. Logistic regression serves as a strong linear baseline, widely used in sentiment analysis due to its interpretability and effectiveness when paired with well-separated features like word embeddings. It models the probability of a review belonging belonging to the positive or negative class, making it well-suited for binary sentiment tasks and compatible with the original paper's methodology.

In addition to logistic regression, we experimented with a feedforward neural network to evaluate whether a non-linear model could better capture subtle semantic relationships within the embedded feature space. Neural networks have the capacity to model complex interactions between word features that may not be linearly separable, especially when using contextualized embeddings such as BERT. By comparing both models, we aim to assess the benefit of richer representations and non-linear modeling in integrating sentiment information.

We used the binary cross-entropy loss, also known as log loss, defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] \tag{2}$$

where N is the number of samples in the dataset, y_i is the true label for sample i and $\hat{y}_i$ is the predicted probability for the positive class for sample i. This loss function is the well-suited for binary classification.

Ultimately, we tried to fine-tune the word vectors with backpropagation of the error for the sentiment classifier through the embeddings. This way, embeddings are updated not only based on co-occurrence but also on their effectiveness in predicting sentiment. To do so, we considered a simple feed-forward neural network with one hidden layer and observed the results.

5 Results

5.1 Semantic part

We experimented the results of our various word representations on several query words and we extract the five most similar terms each time.

Table 1: Top 5 most similar words for selected query terms across different embedding models.

	Word2Vec	**LSA**	**LDA**	**BERT**
sadness	warmth sorrow loneliness despair frustration	happiness cry emotions hearts loneliness	capturing complexity human humanity society	sad anguish grief regret compassion
witty	well-written quirky sentimental snappy clever	wit timing comedic quirky charm	lovable romantic stellar charming pleasant	comedies sarcastic stimulating downfall unconventional
dull	tedious boring predictable unoriginal pointless	boring tedious pointless there's nothing	clichéd cheap mediocre horribly pointless	painful depressed shiny smoky dismay
romantic	performance portrayal role actor cast	einstein romance meg screwball comedies	lovable pleasant likable charming nicely	sentimental sensual romance sexual flirting

The results from 1 display that overall the results are quite satisfying, especially for Word2Vec among the "simple" word representations. Between Word2Vec, LSA, LDA, Word2Vec seems to be the only one that does not give some incoherent results such as "there's" for LSA or even "human" for LDA. On the whole, BERT gives the best results, with closer words in meaning and capturing a bit of the essence of the word. It has more profoundness and is due to the fact that BERT is pre-trained. It is thus satisfying for later predictions.

5.2 Sentiment classification

For sentiment classification, we used logistic regression with `max_iter=1000` as maximal number of iterations and a bias term for regularization as preconised in the article. For Word2Vec, LSA, and LDA, we used the complete train and test dataset. However, for BERT, we had to take a subset of the dataset (with a train size of 1000 reviews and a test size of 500 reviews) because otherwise it took way too long to train and crashed systematically even when using batches.

The results of accuracy with binary cross-entropy loss given in table 2 are thus all comparable except for BERT where the training and test were made on restricted datasets. However, what is very promising is that with fewer samples, BERT reaches the same levels of accuracy than the other models.

Table 2: Classification accuracy of sentiment and semantics on the IMDB database

Model	**Accuracy (%)**
Word2Vec	81.9
LSA	84.28
LDA	80.74
BERT	82.2

Hence, the models yield high accuracy after the fine-tuning operated. BERT is almost as high as the others with fewer samples and LSA reaches a high level of accuracy thanks to the underlying co-occurrence matrix.

6 Conclusion and future works

In this project, we explored various approaches to sentiment analysis using the IMDb movie review dataset. By replicating and extending the method proposed by Maas et al. (2011), we utilized sentiment-aware word vectors and tested them on a binary classification task. Through experiments using traditional word embeddings like Word2Vec, LSA, LDA, and modern techniques like BERT, we compared their performance for sentiment classification using models such as Logistic Regression. Our results highlighted the effectiveness of contextualized embeddings, particularly BERT, which outperformed older models such as Word2Vec and LSA in terms of accuracy and robustness.

However, with more time and resources, we would have liked to explore more advanced neural network architectures, such as Convolutional Neural Networks (CNNs) and recurrent neural networks (RNNs). These models have shown remarkable success in sentiment analysis tasks due to their ability to capture complex patterns and contextual information from sequential data, like text. Additionally, CNNs, in particular, have been widely used for sentence-level classification tasks, as they can identify local patterns in the text, such as n-gram features, that are relevant for sentiment classification.

Another idea for further research would be to test the fine-tuned embeddings on the query word we used at the beginning to see if progress has been made.

In addition, the original IMDb dataset includes continuous sentiment labels ranging from 1 to 10, which we did not explore in this work. The binary classification approach (positive/negative) was a simpler way to evaluate the models, but a regression-based approach using continuous sentiment labels would allow us to predict the exact sentiment intensity of a review, providing a more nuanced understanding of sentiment. This would be particularly interesting for evaluating how well the models can capture varying degrees of sentiment rather than just a binary outcome.

Finally, while our approach focused on static sentiment classification, further research could look into integrating dynamic models that take into account the evolving context of sentiment over time or across different domains (e.g., film genres). Sentiment analysis can vary greatly depending on the domain, and future work could explore transfer learning or domain adaptation techniques to improve performance across different types of text.

In conclusion, this project demonstrated the potential of modern NLP techniques, including transformer-based models like BERT, for sentiment analysis. Despite the challenges posed by the inherent complexity of natural language, the results provide valuable insights into how word vectors, sentiment supervision, and classification models can be combined to improve sentiment classification tasks. With more time and computational resources, the approaches explored here could be expanded and refined to achieve even better results.

References

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, 2011.

[2] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[3] David M. Blei, Andrew Y. Ng, and John D. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*, 2013.

[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 12:1532–1543, 2014.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2019.