Sentiment Analysis on Movie Reviews

HEZBRI Nour
nour.hezbri@ensae.fr

Abstract

Sentiment analysis seeks to automatically detect opinions and emotions in text, making it a key tool for business, politics, social media, and healthcare. Early work Maas et al. [2011a] reported that word embeddings trained with sentiment labels improve classification over generic vectors usage. Building on this idea, we compare modern transformer models to the classic approach of that paper, using the IMDb movie-review dataset. We benchmark static embeddings (GloVe, Word2Vec, BERT, RoBERTa) as well as fine-tuned encoders (BERT, RoBERTa, XLNet, ELECTRA, DistilBERT) paired with simple linear classifiers. Our results demonstrate that contextual embeddings alone outperform the old baseline and that end-to-end fine-tuning further boosts accuracy, highlighting the value of current deep models, especially transformer-based architectures for sentiment tasks. We provide the code to reproduce our experiments in the following repository `https://github.com/nourhez09/NLP_course.git`.

1 Introduction

Sentiment analysis, opinion mining, subjectivity tracking, and many other expressions describe the aim of automatically detecting opinions and emotions expressed in text, which confer a certain polarization to it. This field has many applications in business, politics, social media, and healthcare. Therefore, it has always been an active and highly interest-catching area of research within Natural Language Processing.

Early work in sentiment analysis focused on learning word representations that capture sentiment polarity efficiently. This would be the main building block—as one could expect, if we want to capture sentiment subtleties, the representation of words in vector space should embed that information. Improving word embeddings and representations is therefore essential to improving the sentiment analysis task, among others. In particular, Maas et al. [2011a] introduced a method for "learning word vectors for sentiment analysis," that combines an unsupervised and a supervised approach, demonstrating that embeddings trained with document-level polarity labels could improve sentiment classification over generic word vectors.

In this work, we build upon Maas et al.'s approach, using it as a baseline, as it is quite outdated in the current landscape of methods. We aim to improve sentiment classification accuracy using current state-of-the-art models and architectures.

The rest of this paper is organized as follows. In Sec. 2, we briefly summarize the state-of-the-art strategies and approaches in sentiment analysis. In Sec. 3, we describe the dataset (Sec. 3.1) and the models (Sec. 3.2) we explore for this task. In the experimental section (Section 4), we benchmark several models to evaluate their performance on a standard sentiment classification task using the IMDb dataset Maas et al. [2011b].

2 Related works

Over the last decade, sentiment analysis in NLP has transformed from simple static word-vector models that capture generic co-occurrence patterns to large, context-aware pre-trained transformers that can be prompted to perform sentiment tasks with minimal labeled data. Early efforts learned vectors specifically tuned for sentiment, while subsequent methods improved global structure, subword handling, and context sensitivity. More recently, transfer learning and prompt-based approaches have enabled strong sentiment performance in few-shot and zero-shot settings.

In 2011, Maas et al. [2011a] introduced a hybrid unsupervised–supervised model that learned "sentiment-specific word embeddings" by combining document-level polarity annotations with a neural word-vector framework, demonstrating improved performance on movie-review classification benchmarks. Building on the idea of vectorized semantics, Mikolov et al. [2013] proposed Word2Vec, which learns continuous skip-gram and CBOW embeddings from large corpora, capturing rich semantic and syntactic relations—but not yet sentiment nuances. To inject sentiment information directly into embeddings, Tang et al. [2014] presented Sentiment-Specific Word Embeddings (SSWE), jointly optimizing for context prediction and supervised polarity, thus bringing positively and negatively oriented terms closer or farther apart in the vector space.

Pennington et al. [2014] introduced GloVe, a global co-occurrence factorization approach that produces high-quality static embeddings by combining local and global statistics. GloVe quickly became a standard input for sentiment classifiers.

Static embeddings often fail to reflect how a word's polarity can shift depending on context. To address this, many approaches were developed to leverage contextual information for solving NLP tasks, including sentiment analysis. Particularly with the introduction of the attention mechanism Vaswani et al. [2023], the Transformer architecture revolutionized language modeling and significantly advanced the state of the art.

Devlin et al. [2019] introduced BERT, a deeply bidirectional encoder pre-trained on masked-language and next-sentence prediction objectives. BERT set new records on multiple benchmarks, including sentiment datasets, by learning rich contextual representations at every layer. Building on BERT, Liu et al. [2019] released RoBERTa, which refined pre-training hyperparameters and optimized data usage to push performance further on sentiment and general NLP tasks.

Finally, Brown et al. [2020] scaled autoregressive Transformers to 175 billion parameters (GPT-3), demonstrating that prompt-based few-shot or zero-shot learning can tackle sentiment analysis without any model fine-tuning, simply by crafting input templates with examples. This prompt-based paradigm represents the current state of the art, offering rapid domain adaptation and multilingual sentiment capabilities with minimal annotation.

3 Methodology

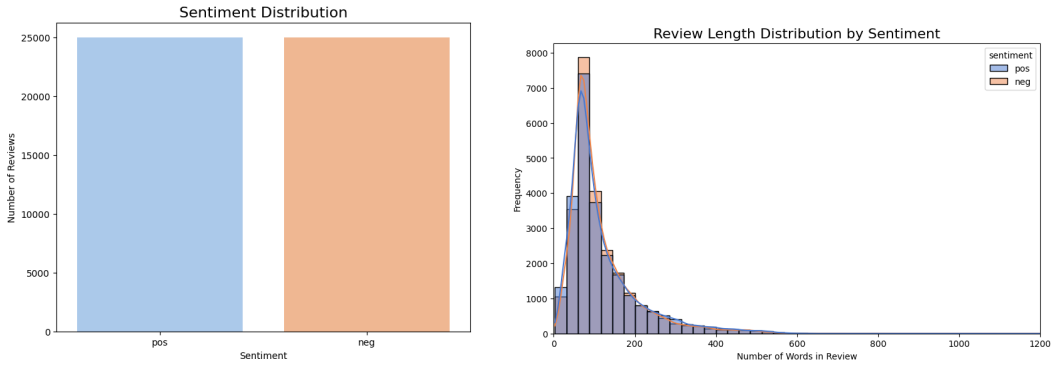
3.1 Data analysis

In this project, we study the dataset introduced in Maas et al. [2011b], which was proposed as a benchmark for sentiment analysis in the seminal work by Maas et al. [2011a]. The dataset contains 50,000 labeled samples, of movie reviews, equally divided into training and test sets, with balanced positive and negative examples (see Fig. 1a). In particular, only reviews with strong sentiment are included, setting a binary classification problem.

The authors also provide additional unlabeled data samples, enabling unsupervised or self-supervised approaches. This makes the dataset a convenient and robust benchmark for evaluating sentiment analysis methods.

We begin by exploring some characteristics of the dataset. To do so, we perform standard preprocessing steps such as removing stop words and HTML tags, lowercasing, and removing punctuation and numbers, in order to better visualize the data content. Note that depending on the model we choose, we may or may not apply the same preprocessing steps during training and evaluation (see in particular Sec. 4).

The reviews in the dataset vary in length, but the length distributions are analogous for both positive (blue curve and histogram) and negative (red curve and histogram) samples. Therefore, we do not expect any bias from review length (see Fig. 1b).



(a) (b)

Figure 1: Some statistics of the dataset

Next, we examine the word clouds of the most frequent words in both positive and negative reviews. As shown in Fig. 2, this visualization is not particularly informative, as many frequently occurring words (e.g., *film*, *movie*, *show*) appear in both classes. This is expected, since the dataset exclusively contains movie reviews.

Figure 2: Reviews point clouds before stripping neutral words

To enhance this observation, we define a neutrality threshold: a word is considered neutral if the difference in its occurrence frequency between positive and negative reviews is below 0.2—a manually tuned hyperparameter. This filtering is illustrated in Fig. 4. After removing such neutral words, the remaining word clouds reveal a clearer sentiment polarization (see Fig. 3).

Figure 3: Reviews point clouds after stripping neutral words

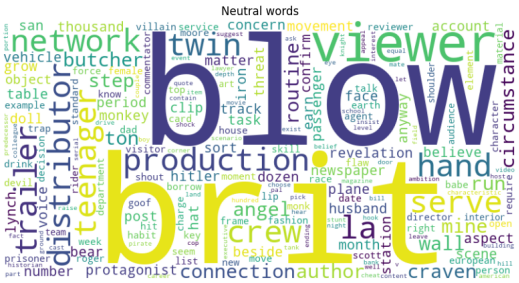


Figure 4: Neutral vocab

Note that in our study, we restrict the analysis to the labeled portion of the dataset. This contrasts with the approach in Maas et al. [2011a], where the authors propose a model that makes use of the unlabeled data (cf. Full+unlabeled+bag of words). Although the improvement reported in that paper was marginal, it is worth noting that, since we employ transformer-based models—which are particularly data-hungry—leveraging additional unlabeled data could potentially enhance training performance.

3.2 Models

3.2.1 Model from the paper Maas et al. [2011a]

To enrich semantic representations with sentiment information, the proposed approach extends a basic semantic model (unsupervised) by incorporating a sentiment-aware component (supervised). The steps of the overall representation model are as follows:

Capturing Semantic Similarities A document is viewed as a continuous mixture over latent "semantic directions" θ, so that words whose embeddings align with θ become more probable. We assume

$$p(d) = \int p(\theta) \prod_{i=1}^{N} p(w_i \mid \theta)\, d\theta$$

and model each word via a log-linear energy

$$E(w;\theta) = -\,\theta^\top \phi_w - b_w,$$

turning it into a probability by

$$p(w \mid \theta) = \frac{\exp(\theta^\top \phi_w + b_w)}{\sum_{w'} \exp(\theta^\top \phi_{w'} + b_{w'})}.$$

Capturing Word Sentiment To bias embeddings toward sentiment, each word vector ϕ_w also feeds a logistic classifier:

$$p(s = 1 \mid w) = \sigma\big(\psi^\top \phi_w + b_c\big),$$

and we maximize over labeled documents

$$\sum_k \sum_i \log p\big(s_k \mid w_i\big).$$

Joint Learning and Optimization We combine the unsupervised semantic objective and the supervised sentiment objective with regularization on the embedding matrix R and the document weights $\{\hat{\theta}_k\}$:

$$\nu \|R\|_F^2 + \sum_k \lambda \|\hat{\theta}_k\|_2^2 + \sum_{k,i} \log p(w_i \mid \hat{\theta}_k) + \sum_k \frac{1}{|S_k|} \sum_i \log p(s_k \mid w_i).$$

We then alternate between fixing $\{\hat{\theta}_k\}$ and updating (R, b, ψ, b_c) and recomputing each MAP [1] estimate $\hat{\theta}_k$, repeating until convergence to learn the representation. on top of which an SVM classifier is then trained to solve the classification task.

[1] stands for Maximum a Posteriori

3.2.2 Proposed Improvements

Although adding a sentiment-aware component is conceptually appealing in the paper, only marginal numerical improvements when it is included are noticeable, by carefully looking at the numbers reported in Table 2 of Maas et al. [2011a].

Semantic only: 87.30% of accuracy versus 87.44% for the Full.

Semantic +bag of words: 88.28% versus 88.33% for Full + bag of words

To improve upon this, we intuitively need to work more on the word representation of the corpus, which would require an encoder that not only captures rich, fine-grained semantic meaning but also reflects subtle sentiment differences. Transformer-based encoders are a natural choice: BERT provides a strong base; but we test also other refined encoders thereof.

Hence, we propose in the next section, to conduct the following investigation study.

1. **Static embeddings + classifier.** We freeze the encoder, extract embeddings for the reviews, and train only a linear SVM classifier on top.
2. **Fine-tuned encoder + classification head.** We initialize a linear head , attach it to the encoder, and fine-tune all weights on our downstream task.

Training and evaluation Since our task maps a long sequence of reviews to a single binary label (many-to-one), we train with binary cross-entropy loss. We report average loss and Top-1 accuracy to compare models fairly. The main challenge is ensuring the encoder captures both semantic similarity and sentiment nuance.

Encoders under test

- **BERT-uncased**: the original bidirectional encoder. Devlin et al. [2019]
- **RoBERTa**: an improved BERT with longer pretraining and dynamic masking. Liu et al. [2019]
- **XLNet**: a permutation-based model that generalizes autoencoding and autoregression.Yang et al. [2020]
- **ELECTRA**: a sample-efficient model trained via a replaced-token detection task.Clark et al. [2020]

4 Numerical experiments

4.1 Setup

All experiments were conducted on a GPU laptop RTX 4060. The pretrained models used in our evaluation are sourced from the Hugging Face Transformers library. Fine-tuning is performed following standard practices, and the specific hyperparameter settings used in our experiments are detailed in Table 1. The full code required to reproduce these experiments is available in the accompanying GitHub repository `https://github.com/nourhez09/NLP_course.git`.

4.2 Results

Our experiments are organised as follows:

our primary baseline is the original "Full + Bag of Words" model from the paper, evaluated on the same labeled dataset,for the sake of fairness although the paper reports a slight accuracy improvement, when unlabeled data is also added. Another baseline, are the classical static embedding methods—GloVe and Word2Vec vectors (each frozen and followed by an SVM classifier).

Next, we test transformer-based encoders by freezing BERT and RoBERTa embeddings and training an SVM on top.

Finally, we perform fine-tuning experiments in which each encoder (BERT, RoBERTa, XLNet, ELECTRA, DistilBERT) is paired with a linear classification head and all weights are updated on our

Hyperparameter	IMDb Dataset
Learning rate	1e-5
Batch size	8
AdamW	Default params
Epochs	2
Sequence length	512
Embedding size	768

Table 1: Unified hyperparameters for all models on IMDb dataset.

downstream task. All experiments use only labeled data of the dataste IMDb. The results are reported in 2.

	Model	Top-1 Test Accuracy(%)	Average Test Loss
Baselines	`Full + Bag of Words from the paper`	88.33	–
	`GloVe + SVM classifier`	75.46	0.5581
	`Word2Vec + SVM classifier`	72.93	0.6009
Static embeddings	`BERT embedding(frozen) +SVM classifier`	85.18	0.3695
	`RoBERTa embedding(frozen) +SVM classifier`	89.32	0.2597
Fine-tuned LLMs	`BERT encoder + classification head`	93.304	0.1862
	`RoBERTa encoder + Linear classification head`	95.440	0.1256
	`ELECTRA encoder + Linear classification head`	94.720	0.1599
	`Xlnet encoder + Linear classification head`	94.164	0.1736
	`DistilBERT for sentiment analysis (fine-tuned)`	93.244	0.1820
	`DistilBERT for sentiment analysis(huggingface) (zero-shot)`	89.07	0.4123
	`RoBERTa for sentiment analysis(huggingface) (zero-shot)`	95.58	0.2571

Table 2: Performance comparison of models on sentiment classification on IMDb dataset.

4.3 Results analysis

From the results reported in Tab. 2, overall, our models substantially outperform the original "Full + Bag of Words" baseline as well as standard static embeddings. The two strongest performers are:

- **Zero-shot RoBERTa**: Pre-trained and fine-tuned on a large multi-dataset sentiment corpus (15 datasets), this model achieves the highest Top-1 accuracy without any additional task-specific training.
- **Fine-tuned RoBERTa**: When we attach and train our own linear classification head, RoBERTa remains the best fine-tuned encoder, narrowly below its zero-shot, already fine-tuned counterpart.

Further observations on Table 2:

- *Baselines:* GloVe and Word2Vec baselines yield 75.5% and 72.9% accuracy, respectively, highlighting the limited capacity of uncontextualized embeddings.
- *Static embeddings:* BERT (85.2%) and RoBERTa (89.3%) significantly improve on classical vectors, demonstrating the value of deep contextual representations even without fine-tuning.
- *Fine-tuning gains:* All fine-tuned encoders exceed 93% accuracy.
- DistilBERT (fine-tuned on the smaller SST-2 corpus) is competitive with the baseline but lags behind the larger transformer variants—likely because its fine-tuning data is less extensive than IMDb.

We evaluate all models using Top-1 accuracy and average cross-entropy test loss. The IMDb dataset is well balanced, and our classification reports confirm overall near-uniform performance across positive and negative classes.

5 Conclusion

In this work, we have revisited the sentiment analysis problem in NLP, where starting from the results of Maas et al. [2011a], we compared them against a variety of embedding-based approaches on the IMDb dataset, to further improve them. Static embeddings (BERT, RoBERTa) paired with a simple classifier already outperform the original baseline, demonstrating the power of contextual representations. Fine-tuning modern transformer encoders further boosts performance, confirming that end-to-end adaptation is highly effective for sentiment tasks.

To build on these findings, for future work to further improve the results we have on this benchmark, we could

- Develop a semi-supervised learning strategy to exploit the large pool of unlabeled reviews available.
- Extend fine-tuning to more epochs and conduct a systematic hyperparameter search for optimal settings.
- Address computational constraints—so far we could train for only two epochs.

References

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020. URL `https://arxiv.org/abs/2003.10555`.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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*, pages 142–150, Portland, Oregon, USA, 2011a. Association for Computational Linguistics.

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*, pages 142–150, Portland, Oregon, USA, June 2011b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1015`.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

D. Tang, B. Qin, and T. Liu. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020. URL `https://arxiv.org/abs/1906.08237`.