Comparative Analysis on Traditional and Emerging Text Classification Methods

Connor Morin
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514

Annie Pi
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514

Christina Yi
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514

Jiangyuan Yuan
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514

Jonathan Zhao
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514

Abstract

This project presents a comparative analysis of diverse text processing methodologies applied to document classification and sentiment analysis. We begin by implementing classical machine learning classifiers, including logistic regression, support vector machines, random forests, and decision trees, using foundational text representations such as TF-IDF. To capture richer semantic information, we develop neural network models, including feed-forward architectures and LSTM-based models that utilize word embeddings. In addition, we explore state of the art transformer architectures by fine-tuning a pre-trained BERT model and assess mainstream large language models under both zero-shot and few-shot conditions. Experiments across multiple datasets will enable a systematic evaluation of each approach using rigorous, task-specific metrics. Ultimately, this project aims to quantify the strengths and limitations of each method, establishing a robust foundation for real-world text processing applications.

1 Introduction

In recent years, the exponential growth of textual data across digital platforms has made effective text processing methodologies more essential than ever. This data spans a wide range of domains, including news articles, research papers, customer reviews, and social media. Because of how vast and unstructured the textual data is, organizations have begun to increasingly prioritize real-world natural language processing applications in order to extract meaningful insights. Thus, natural language processing has emerged as a critical area of research at the intersection of machine learning and artificial intelligence.

A multitude of methods have been developed over the years to handle the problem of text processing. These include classical data science approaches that have been in use for decades, as well as cutting-edge innovations in neural networks and transformer-based architectures developed in more recent years. For more traditional methods such as logistic regression, support vector machines (SVM),

Preprint. Under review.

random forests, and decision trees, they require text representations such as bag-of-words and TF-IDF in order to convert the text into numerical vectors for use [1]. For the newer techniques such as neural networks, LSTM-based models and transformer-based architectures, they use word-embedding algorithms for feature extraction [2].

Given such a rapidly developing area of research, there is a growing need to systematically evaluate and compare the performance of both traditional and modern methods for real-world text processing tasks. This research aims to conduct a comprehensive comparative analysis of the diverse text processing methodologies that can be applied to document classification and sentiment analysis. We seek to identify the strengths and limitations of each approach and develop a solid outlook on the most effective strategies for real-world applications. We will be testing logistic regression, SVM, random forest, decision trees, and neural networks, including LSTM RNN models and BERT, as well as LLMs like GPT-4o.

The datasets we will be using are the "IMBD Dataset of 50K Movie Review" from Stanford University [3] and the "20 newsgroups text dataset" from the scikit-learn documentation [4]. The IMBD dataset includes 50,000 movie reviews, split evenly between training and testing samples. This dataset is for binary sentiment classification tasks, with the reviews being marked as positive or negative, making it ideal for benchmarking traditional classifiers like logistic regression as well as modern deep learning models such as LSTMs and transformer-based architectures. The 20 newsgroups dataset includes around 18000 newsgroups posts on 20 topics. Unlike the IMDb dataset, this dataset is for multi-class classification, with classes like atheism, electronics, and med. Using these two datasets will allow us to create a comprehensive analysis of the generalizability and effectiveness of traditional and modern text processing methods.

2 Approach

2.1 Traditional Text Classification

Our traditional pipeline begins by converting raw text into sparse vector representations using term–frequency inverse document frequency (TF–IDF). We apply standard preprocessing (tokenization, lowercasing, stop-word removal) and extract unigram and bigram TF–IDF features, resulting in a high-dimensional, sparse input space. We split the data into two subsets: 80% was for the training subset, while the remaining 20% was reserved as a validation set to evaluate the model's accuracy and generalization performance. These features are then fed into four off-the-shelf classifiers imported from scikit-learn:

- **Logistic Regression** with ℓ_2 regularization, optimized via stochastic gradient descent.
- **Support Vector Machine (SVM)** with a linear kernel, tuned for the soft-margin parameter C.
- **Decision Tree** classifiers, with maximum depth of 20, selected by cross-validation to control overfitting.
- **Random Forests**, an ensemble of 100 decision trees, using bootstrap sampling and feature subsampling to reduce variance.

Hyperparameters for each model are chosen via five-fold cross-validation on the training set. This family of methods establishes a robust baseline by exploiting sparse lexical signals and margin-based decision boundaries, and allows us to quantify the gains afforded by dense contextual embeddings in later sections.

2.2 Deep Learning

Our architecture selections for evaluating the performance of neural networks on text classification include recurrent neural networks (RNNs) and the transformer architecture. Specifically, we chose to evaluate a "long short-term memory" (LSTM) RNN, which is known for being able to capture longer dependencies and patterns in sequential text, a trait that seemed to be ideal for text classification tasks. As for representing the transformer architecture, we chose to evaluate BERT (bidirectional encoder representations from transformers), a widely used encoder-only language model that learns contextual embeddings for words and is pretrained on massive, diverse corpora. We believe these

choices best represent the capabilities of state-of-the-art deep learning on text classification before and after the introduction of the transformer architecture.

For the RNN, we trained a recurrent neural network from scratch and used an LSTM layer, where its outputs were then fed to a dense layer to produce a final output. Dropout and an Adam (Adaptive Moment Estimation) optimizer were also used, and the model was trained for 5 and 10 epochs for each dataset, respectively. On the other hand, for the transformer, we fine-tuned DistilBERT, a smaller, distilled model that still scores within 95% of BERT base. Due to the long fine-tuning time for BERT, we capped the fine-tuning pipeline to just 2 epochs. As for all the other methods tested in this paper, the deep learning models were evaluated on accuracy, F_1, precision, and recall scores.

2.3 Large Language Models

We employ a pretrained large language model (GPT-4o-mini) to perform both zero-shot classification without any gradient-based fine-tuning. In the zero-shot regime, the model receives only an instructional prompt of the form:

> "Classify the sentiment of the following movie review as either 'Positive' or 'Negative': `<text>`"

Inference is conducted via the OpenAI Python client, which yields a deterministic, single-token response. We normalize the returned text to lowercase and match it against "positive" or "negative," assigning "unknown" to any non-matching output. Evaluation is performed on a reproducible sample of 1,000 IMDb reviews, computing accuracy, macro-precision, macro-recall, and macro-F_1 via `scikit-learn`. A similar prompting scheme, with the 20 Newsgroups category set, is used for topic classification. This approach capitalizes on the LLM's extensive unsupervised pretraining, enabling robust classification with minimal task-specific engineering.

3 Results and Discussion

Our empirical evaluation reveals three distinct performance regimes across the seven methods considered. Classical classifiers (logistic regression, SVM, random forest, decision tree) establish strong baselines on sparse TF–IDF representations. Sequence-based neural networks (LSTM) leverage contextual embeddings to capture word-order information. Finally, pretrained transformer models (BERT) and large language models (LLM) deliver state-of-the-art results via large-scale unsupervised pretraining.

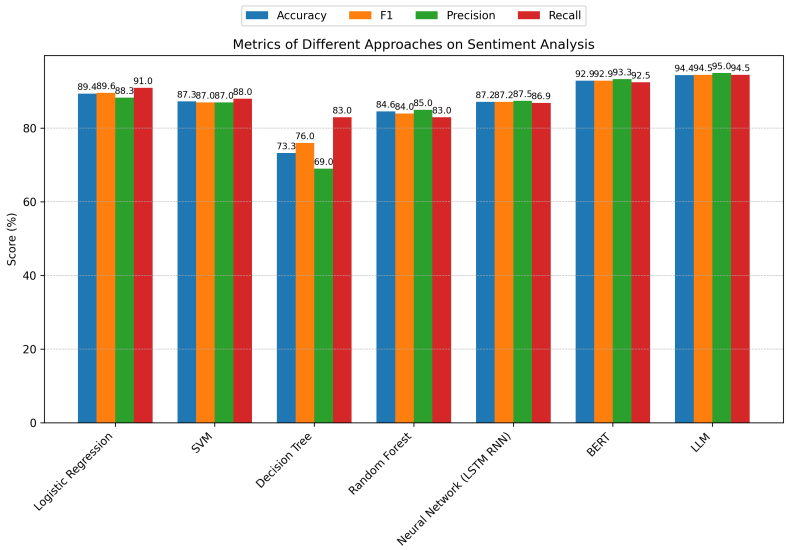

Figure 1: Metrics of Different Approaches on Sentiment Analysis

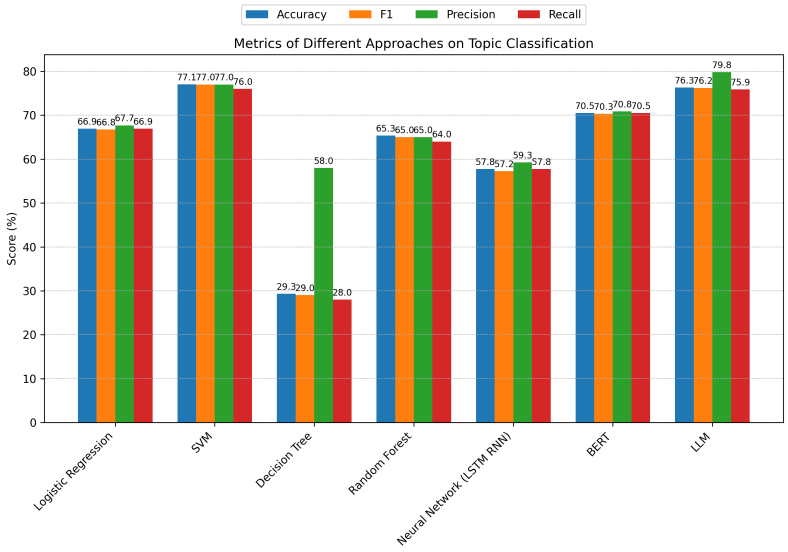

Figure 2: Metrics of Different Approaches on Topic Classification

3.1 Classical Methods

Logistic regression and SVM achieve competitive F_1 scores of 66.8% and 77.0%, respectively, indicating that margin-based decision boundaries generalize effectively in high-dimensional, sparse feature spaces. Random forests perform comparably (F_1 = 65.0%), benefiting from ensemble averaging but still constrained by a bag-of-words view. Decision trees exhibit markedly lower recall (28.0%) despite moderate precision (58.0%), reflecting their tendency to overfit on small, pure leaf regions and thereby miss a large fraction of true positives (F_1 = 29.0%).

3.2 Sequence-Based Neural Network

The LSTM model, which processes word embeddings sequentially, attains an F_1 score of 57.3%. By modeling long-range dependencies, it substantially improves recall relative to decision trees and random forests, but remains limited by its capacity and requirement for task-specific training data, resulting in performance below transformer-based approaches.

3.3 Pretrained Transformers and LLM

Fine-tuned BERT yields balanced performance across metrics (Accuracy = 70.5%, F_1 = 70.3%, Precision = 70.9%, Recall = 70.5%), demonstrating its ability to encode nuanced semantic features. The zero-/few-shot LLM further improves results (Accuracy = 76.3%, F_1 = 76.2%, Precision = 79.8%, Recall = 75.9%), underscoring the value of massive unsupervised pretraining on diverse corpora for downstream classification without extensive fine-tuning.

3.4 Implications and Future Work

These findings confirm that richer contextual representations yield substantial gains: sequential models outperform bag-of-words classifiers, and transformers surpass RNNs by orders of magnitude. However, these approaches require significantly more resources, both in the data needed for training and the computational power required for model optimization and inference. Therefore, classical models remain attractive for low-resource settings due to their efficiency and interpretability.

References

[1] Juluru K, Shih HH, Keshava Murthy KN, Elnajjar P. Bag-of-Words Technique in Natural Language Processing: A Primer for Radiologists. Radiographics. 2021 Sep-Oct;41(5):1420-1426. doi: 10.1148/rg.2021210025.

[2] Wan, Zhongwei. Text Classification: A Perspective of Deep Learning Methods. arXiv, 24 Sept. 2023, https://doi.org/10.48550/arXiv.2309.13761.

[3] Maas, Andrew L., et al. "Learning Word Vectors for Sentiment Analysis." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 142–150. http://www.aclweb.org/anthology/P11-1015.

[4] Scikit-learn Developers. "The 20 Newsgroups Text Dataset." Scikit-learn 0.19 Documentation, https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html. Accessed 24 Apr. 2025.