IMPACT OF MODEL VARIATIONS AND DATA CLEANING ON SENTIMENT ANALYSIS CLASSIFICATION TASKS

Elias Haddad & Anastasia Chen
University of Rochester
Rochester, NY 14627, USA
ehaddad2@u.rochester.edu, achen99@u.rochester.edu

ABSTRACT

In this paper, we explore the performance of BERT on binary sentiment analysis tasks with regard to two main factors: number of linear layers and use of attention-based probing architectures, as well as common dataset cleaning procedures. Our results suggest that model variants with single-headed probing mechanisms achieve exceptionally high performance due to their ability to effectively amplify sentiment-specific features effectively, outperforming models with solely MLP classification heads by more than 10% on average test accuracy. Furthermore, we find that commonly used data cleaning procedures in NLP can hinder BERT performance, due to the nature of BERT's pre-training procedure.

1 INTRODUCTION

Traditionally, machine learning approaches for sentiment analysis have relied heavily on RNNs due to their ability to capture sequential dependencies in text. Architectures based on RNNs such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have demonstrated efficacy in binary sentiment classification tasks, where the goal is to classify input text into positive or negative sentiment classes (Hochreiter & Schmidhuber, 1997; Cho et al., 2014). However, the sequential nature of RNNs poses a series of computational challenges for pretraining large-scale models on massive datasets. This is particularly due to the difficulty and synchronization overhead required to parallelize these models, which was a motivating factor in creating the transformer.

The introduction of the Transformer architecture marked a paradigm shift in NLP, offering significant advantages over RNN-based methods. Transformers utilize self-attention mechanisms with positional encodings, enabling them to capture long-range sequential dependencies in text while benefiting from computational parallelization. This innovation paved the way for the development of transformers, which quickly gained traction for a variety of NLP tasks, including sentiment analysis.

2 BACKGROUND

2.1 SENTIMENT ANALYSIS

Sentiment Analysis has been a widely studied task in Natural Language Processing (NLP). A number of Machine Learning models based around the Recurrent Neural Network (RNN) architecture have been developed for usage especially on binary sentiment classification tasks. With growing interest in Deep Neural Network (DNN)-based Transformers introduced by Vaswani et al. (2017), these models have become a widely used alternative to RNN-based methods, especially due to their computational parallelization attributes. Furthermore, Self-Supervised Learning(SSL) transformer backbones like the Bidirectional Encoder Representations from Transformers (BERT) Devlin et al. (2019) model began to emerge as promising ways for fine-tuning and probing specific model variations for a wide variety of downstream classification tasks.

2.2 Model Fairness

In the context of model fairness, a fair model is defined as a model that makes unbiased predictions across various groups or categories. Fairness has gained significant attention as it is important for a model to not disproportionately favor or disadvantage a single group. This is especially critical in fields like healthcare, criminal justice, hiring, and politics, where the consequences of biased decisions can be far-reaching. There are several types of fairness, including demographic parity, equalized odds, and equality of opportunity (Barocas et al., 2018).

In the context of classification, model fairness is often assessed by examining the accuracy disparity across different classes. Although model bias is typically linked to unbalanced datasets, it can still emerge even with balanced datasets, such as ImageNet. (Cui et al., 2024). The study suggests that bias in image classification models can arise not only from imbalanced datasets but also from problematic feature representation, particularly in classes that are harder to recognize. Such biases can occur because the features extracted for these classes are less representative or more noisy, leading to discrepancies in performance across class labels. Additionally, the paper investigates how techniques like data augmentation and advanced representation learning can reduce biases and enhance fairness in image classification tasks. Consequently, data distillation could play a role in alleviating unfairness in model outcomes.

Fairness matters in reducing discrimination, inequality and increasing generalizability. Unfair models can amplify existing social biases and propagate further discrimination. In addition to mitigating discrimination, fairness in machine learning is essential for fostering trust and ensuring ethical AI deployment. As AI systems increasingly influence decisions in critical areas such as healthcare, criminal justice, hiring, and finance, biased models can have severe consequences, disproportionately affecting marginalized groups and reinforcing systemic inequalities. Moreover, fair models tend to be more generalizable and robust, performing better across diverse datasets found in real-world environments.

2.3 Motivation for Single-Headed Probing Architectures

Many transformer-based sentiment analysis heavily rely on multi-headed attention mechanisms, which are designed to capture a broad range of syntactic and semantic features concurrently. While such redundancy can be beneficial in general-language understanding tasks, it may not always be necessary—especially for simpler classification tasks, where the goal is to identify a relatively narrow set of discriminative signals indicating positive or negative sentiment. Prior studies have noted that smaller, more specialized architectures can sometimes achieve competitive performance by focusing on task-relevant features rather than modeling every nuanced linguistic aspect of the input (Sanh et al., 2019).

However, the literature does not thoroughly address how reducing attention heads to a single "probing" head might highlight sentiment-critical features more directly. By simplifying the attention mechanism to a single head, we remove the need for the model to distribute its capacity to a high dimensional space, but will also preserve enough complexity for the model to attend precisely to the tokens, phrases, or linguistic structures most indicative of some sentiment. In this way, our attention mechanism could be viewed as a "bottleneck" in the model.

To further illustrate our motivation, we can consider bidirectional self-attention as a function where each token assigns weights to every other token to form a contextually enriched representation of the input. Formally, we can consider an input sequence of token embeddings $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the dimensionality of the embeddings. Self-attention computes attention weights $\mathbf{Attn}$ by projecting $\mathbf{X}$ into query, key, and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$ and then computing:

$$\mathbf{Attn} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d}}\right)\mathbf{V}. \tag{1}$$

This operation allows each token to gather information from any other token, capturing dependencies that are not strictly local. In a single-headed scenario, this mechanism acts as a powerful information funnel: rather than dividing the model's representational capacity across multiple heads, a single head with a minimal set of parameters aligns the entire embedding space into a unified contextual representation. This coherence can emphasize sentiment-relevant words and phrases from a more "global" perspective, mitigating local semantic noise and sharpening predictive ability.

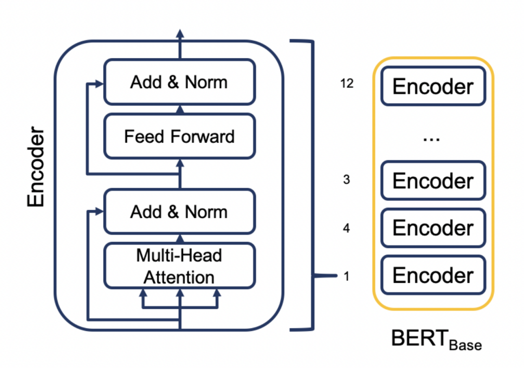

Figure 1: Bert Base architecture

3 Methods

3.1 Data Preprocessing Methods

In this experiment, we used an IMDB movie review dataset obtained from Kaggle, consisting of 50,000 labeled text samples. We tested our models on a raw version and in a cleaned version of the dataset. For the cleaned version, the dataset was subjected to a comprehensive preprocessing pipeline to normalize and optimize the text data for machine learning analysis.

We applied multiple preprocessing transformations to enhance the raw text data's quality and reduce noise. We carefully selected and implemented each technique to improve the potential performance of subsequent machine learning models:

1. **Case Normalization:** All text was converted to lowercase to ensure consistency and prevent the model from treating different letter cases as distinct features. This step helps mitigate potential bias arising from capitalization variations.
2. **Punctuation Removal:** Punctuation marks were systematically removed from the text. This process eliminates potential noise and reduces the dimensionality of the feature space, focusing the model's attention on the textual content rather than syntactical markers.
3. **Stop Word Removal:** Commonly occurring words with minimal semantic value (e.g., "a", "an", "the", "for") were eliminated. Stop words typically do not contribute significant meaning to the analysis and can introduce unnecessary complexity to the model.
4. **Stemming:** Words were reduced to their root or base form using a stemming algorithm. This technique condenses morphologically related words to a common stem, potentially reducing the vocabulary size and capturing core semantic meaning. For instance, "running", "runs", and "ran" would be transformed to the stem "run".
5. **Lemmatization:** Similar to stemming, lemmatization converts words to their base or dictionary form, but with a more linguistically sophisticated approach. Unlike stemming, lemmatization considers the context and part of speech to derive the most appropriate base form, ensuring more accurate word representation.
6. **Number and Symbol Removal:** Numerical digits and symbolic characters were systematically removed from the text. This step helps focus the analysis on textual content and reduces potential noise introduced by non-textual elements.
7. **HTML Tag Removal:** Any HTML or markup tags embedded within the text were stripped, ensuring that only pure textual content remains for analysis.

8. **Spacing Normalization:** Unnecessary whitespaces, including multiple consecutive spaces, leading and trailing spaces, and irregular spacing, were standardized. This ensures consistent text representation and prevents potential parsing issues.

3.1.1 Preprocessing Implementation

The preprocessing pipeline was implemented using Python's Natural Language Toolkit (NLTK) and standard text processing libraries. Each transformation was applied sequentially to ensure comprehensive data cleaning while maintaining the integrity of the original textual information.

3.1.2 Text Preprocessing Demonstration

To illustrate the preprocessing pipeline, we present a representative example of text transformation. The following excerpt demonstrates the comprehensive cleaning process applied to movie review data:

Original, Uncleaned Text	**Preprocessed Text**
`A wonderful little production. <br /><br /> The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece. <br /><br /> The actors are extremely Ill chosen- Michael Sheen not only "has got all the polari" but he has all the voices down pat too! You can truly see the seamless editing guided by the references to Williams\' diary entries, not only is it Ill worth the watching but it is a terrificly written and performed piece...`	`wonderful little production filming technique unassuming oldtimebbc fashion gives comforting sometimes discomforting sense realism entire piece actors extremely ill chosen michael sheen got polari voices pat truly see seamless editing guided references williams diary entries ill worth watching terrificly written performed piece...`

Table 1: Text Preprocessing Comparison

3.2 Model Architectures

We tested these 5 distinct architectures on both the clean and uncleaned IMDB dataset, and thus trained a total of 10 models. For each model, we kept the backbone intact and attached a series of MLP and attention mechanism combinations to the final encoder layer, allowing gradient to flow in these attached mechanisms while stopping all gradient in the backbone to both improve model generalization while also allowing for feasible compute time and resources.

The first series of architectures illustrated in Figure 2 consists of the BERT backbone and either 1, 2, or 3 layered MLP, each with hidden dimension 768 (BERT hidden dimension), and output of 2 for the sentiment classification. Furthermore, each layer is followed by ReLU activation and dropout with $p = 0.1$.

The second series of architectures illustrated in Figure 3 consists of a single-headed self-attention mechanism described above which was directly attached to the last backbone layer of output shape $\mathbb{R}^{64\times308\times768}$ along with the attention head of the same hidden dimension of 768. In this way, the attention head acts as a bottleneck for ensuring all contextual information from the sequence is aggregated efficiently prior to classification. The attention head was also followed both by layer normalization and dropout with $p = 0.1$. Finally, the attention mechanism was proceeded by either a 1 or 2 layer MLP, each with hidden dimension 768 (BERT hidden dimension), and output of 2.

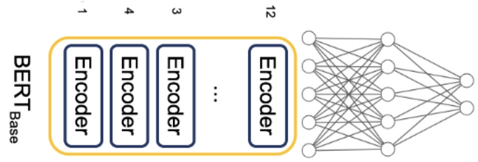

Figure 2: Bert Base + MLP

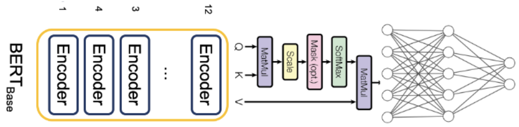

Figure 3: Bert Base + Attention Mechanism + MLP

3.3 Experimental Design

To rigorously evaluate the impact of these model architectures and data preprocessing techniques, we compared each model's performance between the raw, uncleaned dataset and the cleaned dataset. This approach allows for a systematic assessment of how various text preprocessing strategies and model architectures influence machine learning model effectiveness.

We evaluated our models using 3 metrics: Accuracy, Class Disparity, and Fairness, defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Class Disparity} = \max(\text{class accuracies}) - \min(\text{class accuracies})$$

$$\text{Fairness Metric} = 1 - \frac{\text{Class Disparity}}{\max(\text{class accuracies})}$$

where class accuracies are defined as the set of accuracies of the classes. In our case, for example, the class accuracy for the positive class would be the number of correct positive predictions over total number of positive predictions, i.e. $\frac{TP}{TP+FP}$.

4 Experiments & Results

4.1 Accuracy

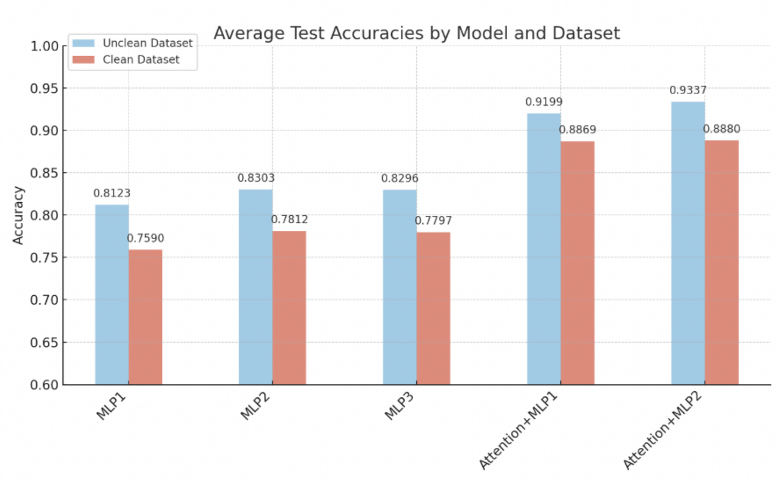

Figure 4: Accuracy

Contrary to expectations, our general data cleaning procedure unexpectedly decreased model performance. This counterintuitive result can be attributed to the pre-training characteristics of the BERT model, which is already trained on diverse, uncleaned textual data. The standard preprocessing pipeline appeared to strip away contextually rich information that the model leverages for nuanced understanding.

The loss of contextual information through traditional cleaning techniques suggests that for transformer-based models like BERT, aggressive text normalization may inadvertently remove subtle semantic cues crucial for accurate classification.

The most significant performance enhancement emerged from the attention mechanism heads. Across all experimental configurations, the introduction of an attention head dramatically increased model accuracy. This finding underscores the importance of adaptive feature weighting in neural text classification models, allowing the network to effectively focus on the most informative textual elements.

Supplementary experiments involving the addition of multiple multi-layer perceptron (MLP) layers at the model head demonstrated modest but consistent performance improvements, while less transformative than the attention mechanism.

These results also challenge conventional text preprocessing wisdom, particularly for pre-trained transformer models. They highlight the need for nuanced approach to data preparation that preserves contextual richness and leverages advanced architectural techniques like attention mechanisms.

4.2 Class Accuracies & Disparities

Our analysis of class-specific accuracies on the raw dataset revealed nuanced performance characteristics across different model architectures. Figure 5 illustrates the accuracy for positive and negative sentiment classes for each model configuration on the raw dataset.

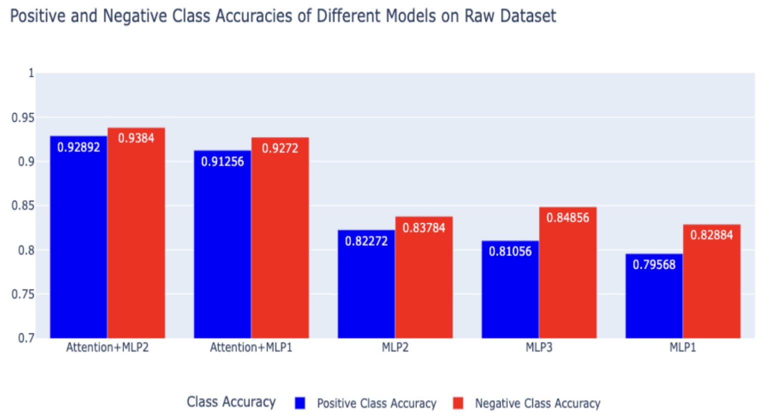

Figure 5: Class Accuracies

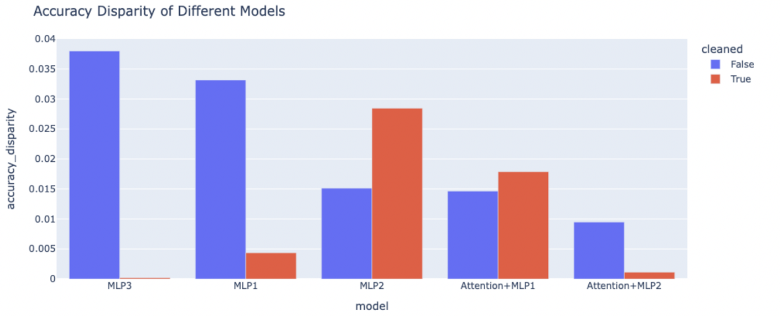

Figure 6: Class Disparity

A consistent trend emerged across all model variants: a subtle but notable bias towards negative class prediction, most likely attributed to model initialization, which predicts solely negative sentiment. This systematic inclination suggests underlying complexities in the sentiment classification task that transcend individual model architectures.

The magnitude of class accuracy disparity varied significantly between model designs for the raw dataset:

- **MLP3 Model:** Exhibited the most pronounced class accuracy imbalance, with the largest gap between positive and negative class predictions. This substantial disparity indicates potential challenges in the model's sentiment discrimination capabilities.
- **Attention+MLP2 Model:** Demonstrated the most balanced performance across positive and negative classes. The minimal accuracy disparity suggests that the attention mechanism combined with a two-layer MLP head may provide the most robust sentiment classification approach.

The observed class-level performance variations highlight the importance of carefully designing model architectures to minimize class prediction bias, utilizing techniques like the attention mechanisms proposed in this study, to improve balanced classification.

These findings underscore the complexity of sentiment classification and the need for carefully designed DNN architectures that can effectively capture subtle semantic distinctions.

4.2.1 Fairness

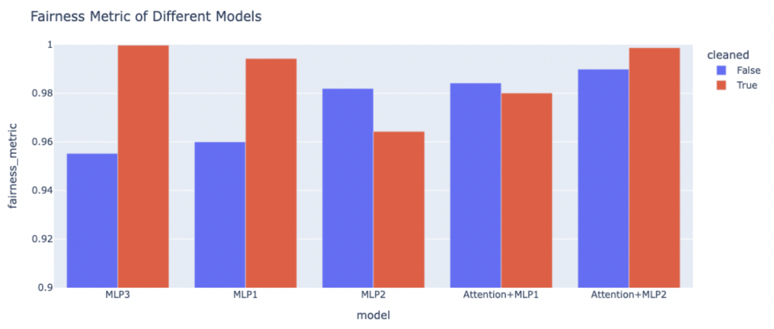

Figure 7: Fairness

Our fairness analysis examined the performance disparities across different model architectures and preprocessing conditions. Figure 7 illustrates the fairness metrics for each model configuration across raw and cleaned datasets.

On the raw, uncleaned dataset, the Attention+MLP model emerged as the most fair classification approach. This model demonstrated the most balanced performance across different sentiment classes, indicating its robustness in handling potential biases inherent in the original text data.

Conversely, the MLP3 and MLP2 models exhibited the least fair performance on the raw dataset. These models showed significant disparities in prediction accuracy between positive and negative sentiment classes, suggesting potential inherent biases in their classification mechanisms.

Interestingly, the fairness landscape shifted when applied to the cleaned dataset. The MLP3 model demonstrated the highest fairness metric on the preprocessed data with the attention+MLP model nearly matching its performance, highlighting the nuanced impact of data cleaning techniques on model performance and bias but also the robustness of the attention-based models across various dataset types.

The variability in fairness metrics across different model architectures and preprocessing conditions underscores several critical insights:

- Model architecture plays a crucial role in mitigating classification biases and improving performance
- Preprocessing techniques can significantly alter a model's fairness characteristics
- No single model consistently outperforms others across all fairness dimensions

These findings emphasize the importance of comprehensive model evaluation that extends beyond traditional accuracy metrics to include nuanced fairness assessments.

5 Conclusion

Our findings challenge conventional wisdom regarding text preprocessing for transformer-based models like BERT, while proposing promising architecture modifications for enhanced text classification. While data cleaning often aims to improve quality, it can strip away contextual cues that BERT leverages to achieve high accuracy and balanced performance. In contrast, incorporating a single-headed bidirectional self-attention mechanism, even without extensive data normalization, consistently enhanced model performance as measured by both accuracy and fairness. Ultimately, careful consideration of preprocessing steps and architectural choices—particularly attention mechanisms—can significantly impact sentiment analysis model performance in terms of both fairness and accuracy.

References

Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness and machine learning limitations and opportunities. 2018. URL `https://api.semanticscholar.org/CorpusID:113402716`.

Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

Jiequan Cui, Beier Zhu, Xin Wen, Xiaojuan Qi, Bei Yu, and Hanwang Zhang. Classes are not equal: An empirical study on image recognition fairness, 2024. URL `https://arxiv.org/abs/2402.18133`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL `https://api.semanticscholar.org/CorpusID:52967399`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In *Neural computation*, volume 9, pp. 1735–1780. MIT Press, 1997.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv e-prints*, art. arXiv:1910.01108, October 2019. doi: 10.48550/arXiv.1910.01108.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, art. arXiv:1706.03762, June 2017. doi: 10.48550/arXiv.1706.03762.