COMP 551 Assignment 3

Carl, Machaalani; Zafeerah Rohomutally; Kris Boudreau

November 16, 2023

Abstract

In our project, we investigated applied machine learning and deep learning NLP techniques to predict emotions from textual data. Specifically, we implemented and investigated the performance of Naive Bayes, a classical algorithm for text classification, and BERT, a powerful transformer-based model that had revolutionized NLP. Additionally, we delved into the impact of fine-tuning BERT to improve model performance. From our analysis of all three approaches, we concluded a BERT model significantly outperforms a Naive Bayes model, and in fact a fine-tuned BERT model exhibits the best results. Moreover further investigated the attention matrices of the BERT-based model so as to understand the attention mechanism in emotion predictions tasks, using visualization tools. Based on our examination of the performance of both model types, we have reached a conclusion that a deep learning model is the most suitable choice for text classification.

1 Introduction

The significance of Emotional Intelligence on humans was highlighted by Daniel Goleman in his work 'Emotional Intelligence: Why It Can Matter More Than IQ' [1] and this concept has now been extended into the domain of machines to enhance interactions and user experiences with AI systems. In our project to explore this area, the focus was on on the practical application of machine learning and deep learning NLP techniques and how well they can predict emotions from textual data. This was achieved by creating two distinct models —Naive Bayes and BERT— which was applied to the Emotion Dataset [2].

The model employing Naive Bayes was implemented from scratch, where the training process used log priors and Laplace-smoothed log likelihoods of each word based on the Emotion training dataset. We achieved an accuracy of 75.66% with our Naive Bayes model, which in fact mirrors the accuracy obtained using the 'scikitlearn' implementation for Multinomial Naive Bayes classifier [3].

As for the BERT-based model, we first explored the performance of a pre-trained model by B. Savani [4]. The accuracy of this pre-trained model stands at 92.65%, showcasing a significant improvement compared to the Naive Bayes model. We further fine-tuned the pre-trained model, resulting in a higher accuracy at 93.05%

For the Emotion dataset, Kannan, E., Kothamasu, L.A. (2022) [5] implemented a fine-tuned pretrained BERT model which they termed as M-BERT and it had average F1-score of 0.976. While our model had an F1-Score of 0.931, there could be future improvements to increase that value, as the M-BERT, employed similar approach to fine-tuning as our model.

In the last section of the project, we investigate the attention matrix, exploring the relationships between words and class tokens in both accurately and inaccurately predicted documents, in order to gain insights on the attention mechanism of the BERT-based model.

2 System Models

Naive Bayes Model

The Naive Bayes model is based on the "naive" assumption that the features, in our case, the words are conditionally independent given the class label. The Naive Bayes model takes an input string x and calculates the posterior class distribution $p(y|x)$. For each class c, $p(y = c|x)$ is proportional to $p(y = c) * p(x|y = c)$. The naive assumption that the input features are conditionally independent gives us $p(x|y = c) = \prod_d p(x_d|y = c)$ where $\{x_d\}$ is the set of features of x. The prediction of the model is then the class c that maximizes $p(y = c|x)$.

BERT Model

A BERT-based model is one that employs bidirectional training within the Transformer architecture. In fact BERT captures contextual information by considering both left and right context in every layer of its architecture. The pretrained BERT model by B. Savani, which we employed in our investigation [4], has already learned rich contextual representations from large amount of textual data. On the testing data set, the model yielded an accuracy of 92.65% and F1 Score of 0.926.

To further improve the accuracy of the model for our specific task, we fine-tuned the pre-trained model. We used the Trainer class from the 'transformers' library [6] for this process. Trainer handles the tasks in the fine-tuning process such as loading the data, setting up the optimizer and learning rate scheduler, training the model, and evaluating it on a validation set. The method we chose for fine-tuning is to have Trainer updating the all of the model's weights, based on the gradients computed during backpropagation on the training dataset.

Moreover, we systematically experimented with various hyperparameter configurations, so as to identify the optimal combination of learning rates, batch sizes, and number of epochs that yielded the highest accuracy or recall on our validation set. From plots of accuracy vs the hyeperparamters, **??** we finally identified that fine-tuning the model with 4 epochs, a learning rate of 1e-5, and a batch size of 64 resulted in the best performance.

3 Experiments and conclusion

3.1 Dataset

The emotion dataset [2] is a collection of short texts of length ranging from 7 to 300 characters, composed of English words. These texts were sourced from tweets and are labeled based on the emotions they convey. Each of these texts is classified by the emotion it conveys. These emotions include anger, sadness, joy, surprise, love and fear. The dataset of 20000 texts has been split into training (16,000 texts), validation (2,000 texts), and testing (2,000 texts) sets.

In Figure A.2, we display the frequencies of some typical words that occur in these texts as well as the frequency of each emotion category within the training dataset, where we found 'Joy' to exhibit the highest frequency and 'Surprise' the lowest.

3.2 Preprocessing Method

3.2.1 Naive Bayes Model Preprocessing

For the Naive Bayes model, we employed the CountVectorizer class from the scikit-learn library [3]. This process, known as vectorization, is crucial for transforming textual data into a numerical format that the model can interpret. The CountVectorizer converts the text data into a sparse matrix of token counts, effectively creating a Bag-of-Words (BoW) representation. The preprocessing steps for the Naive Bayes model included tokenization to split texts into words, vocabulary building from unique tokens in the training set, and occurrence counting for each word in the documents. The result was a sparse matrix, X_train, sized (16000, 15186), representing 16,000 documents and 15,186 unique words, which served as input features for model training.

3.2.2 BERT Model Preprocessing

For the BERT models (both pre-trained and fine-tuned), preprocessing involved tokenizing the text data using the BertTokenizer from the transformers library [6]. BERT requires its specific tokenization because it was pre-trained using a particular vocabulary and tokenization approach. The BertTokenizer entails appending [CLS] and [SEP] tokens to the sequences, segmenting text into WordPiece tokens to manage unknown words, and standardizing text lengths through padding and truncation. This process tailored the text for compatibility with BERT's inputs, enabling the utilization of its pre-trained embeddings.

3.3 Experiment Settings

3.3.1 Naive Bayes Model Settings

The Naive Bayes model was custom-implemented. Key components included computing log priors for classes and log probabilities for features, crucial for handling underflow issues in probability calculations. Moreover, Laplace smoothing was applied. This technique adds a small, non-zero count to every word in the vocabulary, ensuring that each term contributes positively to the probability calculations.

3.3.2 Pre-trained BERT Model Settings

The 'bhadresh-savani/bert-base-uncased-emotion' pre-trained model was employed using the Hugging Face pipeline. This model, pre-trained on a similar emotion classification task, was directly applied to the test dataset without further training,leveraging its inherent understanding of contextual relationships in text.

3.3.3 Fine-tuned BERT Model Settings

The same BERT model was used for fine-tuning to adapt it more closely to our dataset. Some important hyperparameters that we set were the learning rate, the batch size, and the number of epochs. After a lot of trials on different combinations of hyperparameters, we found the best combination to be a learning rate of 1e-5, a batch size of 64, and 8 epochs (giving 93.05% test accuracy). Furthermore, we enabled the "load_best_model_at_end" training argument with the metric set to "f1" to automatically select the best model among the different epochs based on its F1 score. This was particularly crucial due to the imbalanced nature of the dataset. Also, we applied a weight decay (set to 0.01) as a regularization measure to prevent overfitting.

3.4 Experiment 1: Model Comparisons

The performance of each model has been summarized Table 1 below.

Model	Accuracy	F1 Score
Naive Bayes	75.66	0.801
Pre-Trained BERT-based	92.65	0.926
Fine-tuned BERT-based	93.05	0.9309

Table 1: Accuracy and F1 scores for models

As observed above Naive Bayes struggles with emotion classification due to its simplistic assumptions and thus results in an accuracy of 75.66%, which is significantly lower than that of both pre-trained and fine-tuned BERT-based models, producing an accuracy of 92.65% and 93.05% respectively. These two models showcase remarkable performance by leveraging their contextual understanding and nuanced linguistic representations, which makes them ideal for the task of text classification.

However, for our fine-tuned model, we observed that even after testing different hyperparameters to find the optimal one, we could only achieve an increase of 0.4% in accuracy. A possible explanation for this could be that the pre-trained model has already converged to a near-optimal solution, thus it hard to observe a significant improvement when fine-tuning using the 'Trainer' class, which could have certain limitations. For future improvements, implementing a manual training loop for fine-tuning with custom training strategies that caters for our dataset could potentially yield a more accurate model.

3.5 Experiment 2: Investigating the attention mechanism

In a transformer-based model like BERT, the attention mechanisms play a pivotal role in enhancing the model's ability to understand and process sequential information. This mechanism allows the model to focus on different parts of the input sequence when making predictions, providing a way for the model to capture relationships between words in a more dynamic and context-aware manner.

For our model, we utilized BERTViz [7], a tool to used to visualize the attention matrices within the BERT model. By utilizing BERTViz, we aimed to gain insights into how the model allocates attention between words and class tokens, particularly in the context of correctly and incorrectly predicted documents.

3.5.1 Correct Predictions

We first looked into attention patterns for sentences that were correctly predicted by the model. As we can see in Figure 3(a), the attention mechanism assigns a higher attention weight to the term "feeling" when processing the word "vain" .In this specific instance, the model recognizes the semantic importance of "feeling" in understanding the context of "vain," leading to a higher attention weight. Similarly, as shown in Figure 3(b) , the term "feeling" has a stronger association to the words "rotten" and "ambitious". Additionally, it captures the comparative aspect introduced by "rather" showcasing the model's sensitivity to which feeling is being expressed.

3.5.2 Incorrect Predictions

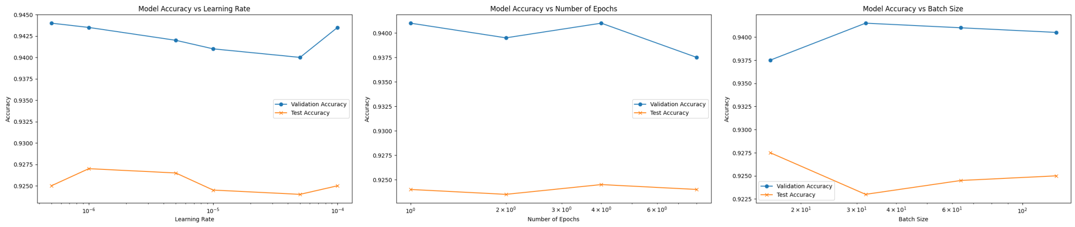
For the sentence is Figure 4(a), the sentence was incorrectly predicted to express "Joy", when the correct emotion was "Love". This is perhaps shown in the image, where "I" has a higher attention weight to the word "love", which the model would predict to be a happy emotion. However, on a positive note "Joy" and "Love" emotions that are highly correlated and thus it is likely that a more refined model would be able to correct predict it. As for Figure 4(b), it illustrates an instance of an incorrect prediction where the sentence's inherent ambiguity results in a lack of clarity in attention weight allocation. The model predicted that the emotion is "fear", when it should in fact be "anger". We thus looked at the attention pattern of the words in the sentence with another that was correctly predicted to be "fear". As per Figure 5, the attention mechanism allocated a considerable attention weight to the words "scared" and "anxious". This shows the model's heightened focus on specific emotionally charged words, thus impacting the overall classification outcome.

In regards to incorrect predictions, we encountered some miscellaneous cases where words were split into sub-words when not needed, such as "billiards" was split into "bill" and "##iards". This had an impact on the model's understanding of the context. For future improvements we can employ strategies such as special tokens for unusual words, expanding the model's vocabulary, implementing post-processing steps, and incorporating custom tokenization rules.

3.6 Conclusions

To conclude, our project delved into the practical application of machine learning and deep learning NLP techniques to predict emotions from textual data.Two models, Naive Bayes and BERT-based, were developed and applied to the Emotion Dataset. After an in-depth investigation, we concluded that for text classification, deep learning methods, that is the BERT-Based model outperforms traditional machine learning methods. To explore how the BERT-based models provides such accurate results, we conducted an in-depth analysis of the attention matrix to gain insights about the model attention mechanisms across diverse sections. Our findings highlights the pivotal role of pretraining, particularly for datasets like Emotions. The task of predicting emotions necessitates the model's ability to capture intricate linguistic patterns—a feat that cannot be achieved by Naive Bayes model. Through an in-depth examination of specific attention matrices between words and class tokens, the adeptness of a BERT-based becomes evident in accurately deciphering emotions in sentences. As we move forward, this project served as a foundation for continual improvements in machine learning and natural language processing.

3.7 Extra Experiments: Varying Hyperparameters



(a) Varying Learning Rate

(b) Varying Number of Epochs

(c) Varying Batch Sizes

Figure 1: Graph of accuracy vs different hyperparameters

3.7.1 Varying Learning Rate

In an experiment to determine the optimal learning rate for BERT model fine-tuning, we evaluated a range of rates from 5e-7 to 1e-4, holding other hyperparameters constant. The graph in Figure 1a shows that the model achieved peak validation accuracy at a learning rate of 5e-7 and peak test accuracy at a learning rate of 1e-6. This suggests that the model was slightly overfitting on a learning rate of 5e-7, and that 1e-6 is the most suitable learning rate for this task.

3.7.2 Varying Number of Epochs

In the investigation of the optimal number of training epochs, we fine-tuned the BERT model across a range from 1 to 8 epochs. Figure 1b illustrates that while validation accuracy peaks at 4 epochs, test accuracy does not significantly fluctuate, maintaining a plateau throughout the range. This indicates that extending training beyond 4 epochs does not yield substantial gains in generalization, as evidenced by the consistent test accuracy.

3.7.3 Varying Batch Sizes

The assessment of varying batch sizes revealed that the highest test accuracy was achieved with the smallest batch size of 16, as shown in Figure 1c. Despite a lower validation accuracy at this batch size, the model's generalization to the test set was superior. This outcome suggests that smaller batch sizes may offer a better gradient estimation for this particular task, leading to improved test performance.

4 Statement of Contributions

- Carl oversaw the acquisition and preprocessing of the Emotion dataset. Additionally, he implemented the Naive Bayes model and the pre-trained BERT model. He also ran the extra experiment of varying hyperparameters and discussed its results.

- Zafeerah contributed on fine-tuning the model, and worked mainly on running the experiments and report writing.

- Kris played a pivotal role in the creation of this sentence, ensuring its existence through unwavering support and the profound absence of action. His contribution is deeply engraved in the space between the words.

A Appendix

A.1 Dataset Visualization

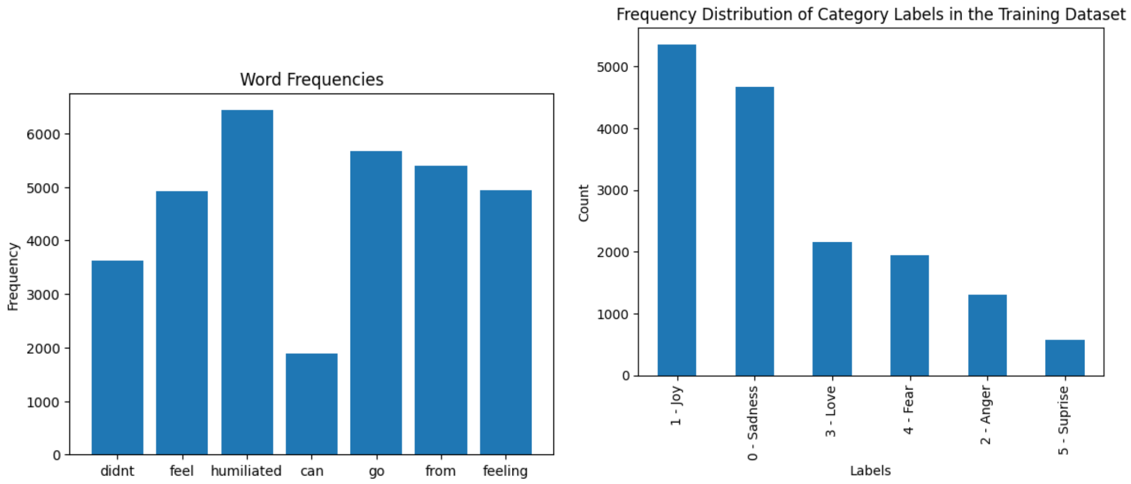
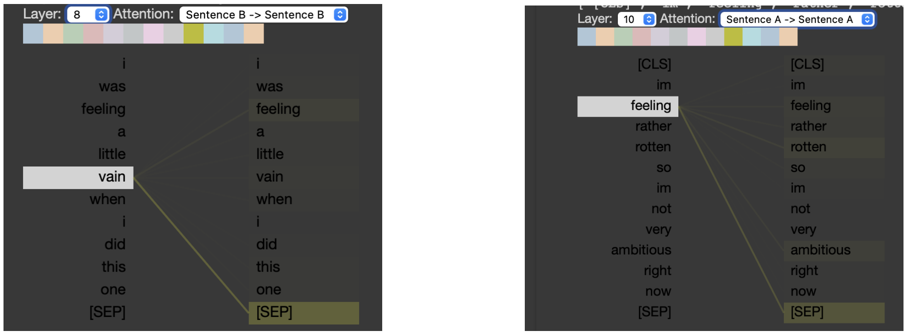


Figure 2: Emotion dataset

A.2 BertViz Results - Attention matrices

A.2.1 Correct Predictions



Layer: 8 Attention: Sentence B -> Sentence B

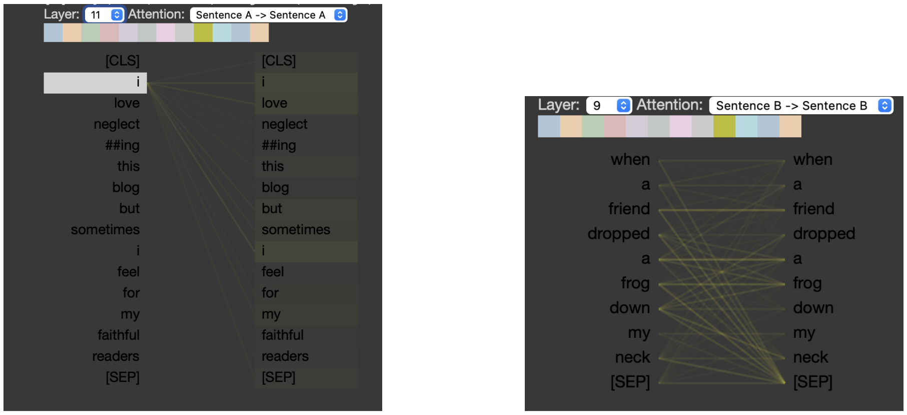
(a) Word-level attention for the correctly predicted sentiment at layer 8

Layer: 10 Attention: Sentence A -> Sentence A

(b) Word-level attention for the correctly predicted sentiment at layer 10

Figure 3: Attention patterns for correctly predicted emotions

A.2.2 Incorrect Predictions


Layer: 11 Attention: Sentence A -> Sentence A

Layer: 9 Attention: Sentence B -> Sentence B

(a) Word-level attention for the incorrectly predicted sentiment at layer 11

(b) Word-level attention for the incorrectly predicted sentiment at layer 9

Figure 4: Attention patterns for a incorrectly predicted emotions

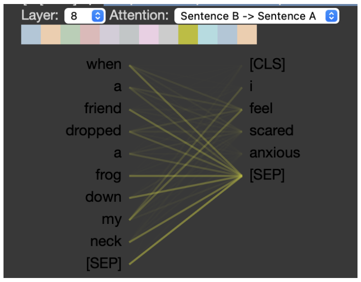

Layer: 8 Attention: Sentence B -> Sentence A

Figure 5: Attention weights at layer 8 for incorrectly predicted sentence

References

[1] D. Goleman, “Emotional intelligence: Why it can matter more than iq,” vol. 24, pp. 49–50, 01 1996.

[2] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, “CARER: Contextualized affect representations for emotion recognition,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 3687–3697, Association for Computational Linguistics, Oct.-Nov. 2018.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] B. Savani, “bert-base-uncased-emotion.” https://huggingface.co/bhadresh-savani/bert-base-uncased-emotion.

[5] E. Kannan and L. Kothamasu, “Fine-tuning bert based approach for multi-class sentiment analysis on twitter emotion data,” *Ingénierie des Systèmes d’Information*, vol. 27, no. 1, pp. 93–100, 2022.

[6] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Association for Computational Linguistics, IEEE, 2020.

[7] J. Vig, “A multiscale visualization of attention in the transformer model,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, (Florence, Italy), pp. 37–42, Association for Computational Linguistics, July 2019.