COMP 551 ASSIGNMENT 3

Antony Zhao, Peng Zhang, Shichen Li

November 12, 2023

1 Abstract

In this project, we assessed the efficacy of the Naive Bayes, the pre-trained BERT models, and our modified BERT model in classifying textual data. We utilized the emotion dataset from dair-ai for our analysis. Our experiments indicated that the finetuned BERT models exhibited superior accuracy compared to the Naive Bayes model. This finding highlights the enhanced performance of the BERT model with the dataset under consideration. Additionally, our modified BERT model marginally outperformed the pre-trained version. For the pre-trained BERT model, we conducted an analysis that compared attention matrices between words and class tokens for a subset of correctly and incorrectly predicted labels. This analysis uncovered discernible patterns that correlate with the model's prediction accuracy.

2 Introduction

In the field of Natural Language Processing (NLP), text classification has become a cornerstone for applications ranging from sentiment analysis to automated customer support. This report presents a comparative analysis of traditional and advanced methodologies in this domain, focusing on the Naive Bayes classifier and two variants of the Bidirectional Encoder Representations from Transformers (BERT) model – a standard pre-trained version and a custom-modified version.

The study employs the emotion dataset from dair-ai to assess these models' efficacy in classifying emotional tones in text. We begin with the Naive Bayes model, a simple yet historically effective probabilistic classifier, and then examine the standard and modified BERT models, known for their advanced performance in NLP tasks.

Our findings, based on accuracy metrics, indicate that the Naive Bayes model achieved 76.55%, the pre-trained specialized for 'emotion' dataset BERT model 93%, the base pre-trained BERT model 35% and the best modified base BERT version 92.9%. Further, we explore qualitative aspects by analyzing the attention mechanisms in BERT models, particularly how they prioritize textual elements in classification. This involves visualizing the model's attention across various tokens and interpreting the gradations in attention intensity.

The EmotionX-2019 paper by Luo and Wang3 presents a methodology highly relevant to our project. Their approach involves using a pre-trained BERT model to detect emotions from dialogue datasets, aligning closely with our aim of emotion classification. They successfully encode utterances into meaningful vector sequences and employ a softmax classifier for emotion prediction, a process we mirror in our work. Their impressive performance, demonstrated by high micro-F1 scores, validates the effectiveness of fine-tuning BERT for domain-specific tasks like ours, underscoring the potential of leveraging pre-trained models for nuanced text classification challenges.

3 System Models

3.1 Naive Bayes

Our implementation of Naive Bayes classifier is structured as a Python class, featuring methods for initializing parameters, fitting to data, making predictions, and evaluating accuracy. Upon initializing the class, attributes like log_prior, log_likelihood, and classes are set, which are essential for the model's functionality. During the fitting process, the fit function calculates the log probabilities of the prior and likelihoods for each emotion class, incorporating Laplace smoothing to manage zero probabilities. The prediction method employs these computed probabilities to determine the most likely emotion for each text instance in the dataset, using a combination of the log_prior and the dot product of the instance with the transpose of log_likelihood. Finally, the model's performance is evaluated through the evaluate_accuracy function, which compares the predicted emotion labels against the true labels, thus computing the accuracy. This implementation of Naive Bayes, a probabilistic model grounded in Bayes' Theorem and assuming feature independence within each class, is particularly suited for our text-based emotional classification task. It demonstrates effectiveness in handling categorical data, which is crucial for analyzing and interpreting the diverse range of emotions expressed in the dataset.

3.2 Pre-trained BERT Models

In our experiment, we utilize two distinct models from Hugging Face's Transformers library for emotion classification on the dair-ai/emotion dataset. The first model bert-base-uncased-emotion1, with a unique training background to assess their performance on the dair-ai/emotion dataset, pre-trained specifically on the dair-ai/emotion dataset, is presumed to exhibit a high degree of specialization and accuracy for this particular dataset, given its tailored training. For the second one, we are using the BERT-base-uncased model from Hugging Face's Transformers library2. This model has been pre-trained on a large corpus of text and learned a wide range of language representations. Despite being pre-trained, this model is not specifically trained on the dair-ai/emotion dataset. Therefore, to adapt this model to emotion classification, we will fine-tune it on our dataset.

3.3 Modified BERT Model

The core idea of this model is to classify emotions in text by investigating the impact of BERT model's architectural modifications and its subsequent fine-tuning on performance outcomes. This task utilizes the bert-base-uncased model because the bert-base-uncased-emotion variant is already fine-tuned and demonstrates high performance. Specifically, we experimented with different numbers of hidden layers (8, 12, and 16) in the BERT architecture. This modification was aimed at understanding how the depth of the model affects its ability to classify emotions in text accurately. For fine-tuning, we employed the train_Bert_model function for the critical task of fine-tuning our pre-trained BERT model. This function orchestrates the training process by iteratively processing the dataset in 64 batches, during which the model's weights are updated through the backpropagation algorithm. We employed the AdamW optimizer with learning rate 2e-5, a standard and effective approach for fine-tuning BERT models. The training phase spanned over 8 epochs, facilitating incremental learning from the training data. Subsequent to the training phase, our model underwent a rigorous evaluation on a test set. This crucial step assessed the model's performance and, importantly, its ability to adapt to the task of emotion classification, ensuring not only proficiency on the training data but also robust generalization to new, unseen data. A key component of our methodology was the feedback loop, where the function provided model accuracy as a tangible measure of performance. This metric was invaluable for gauging the effectiveness of our fine-tuning efforts and guided us in any necessary adjustments to optimize the training process. The training was conducted over 8 epochs, allowing each modified model version to learn incrementally from the dataset. By adjusting the model's structure and fine-tuning it, we aimed to strike a balance between the model's complexity and its ability to generalize well to unseen data, thereby optimizing its performance for emotion classification.

4 Experiments and conclusion

4.1 Datasets

The "Emotion" dataset, available on Hugging Face, is a collection of English Twitter messages categorized into six basic emotions: anger, fear, joy, love, sadness, and surprise. The dataset contains two types of data fields: text, which is a string feature representing the Twitter message, and label, a classification label indicating the associated emotion. The emotions are coded as sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5).

This dataset is structured into two configurations: a 'split' configuration comprising a total of 20,000 examples divided into training, validation, and test splits (16,000 for training, 2,000 for validation, and 2,000 for testing), and an 'unsplit' configuration with a single train split encompassing 416,809 examples.

4.2 Preprocessing method

4.2.1 Naive Bayes

For the Naive Bayes classifier, data preprocessing involves transforming text data into a numerical format suitable for the model. This process starts with data loading, where data from a JSONL file is converted into a Pandas DataFrame, with each line in the file representing a unique data point. Following this, the text is vectorized using the CountVectorizer from scikit-learn, creating a Bag of Words representation. Here, each unique word in the text documents is turned into a feature through token counts. Finally, the data is split into training and test sets, ensuring proper segmentation for effective model training and evaluation.

4.2.2 BERT Model

Our data preprocessing method for emotion classification using BERT involves loading data from JSONL files using a custom function, extracting texts and labels for training and testing datasets, and initializing a pre-trained BERT tokenizer designed for emotion analysis. It includes a custom EmotionDataset class, derived from PyTorch's Dataset, to handle text data tokenization, adding special tokens, attention masks, and padding. Finally, we prepared two BERT models for sequence classification: one fine-tuned for emotion classification and another generic pre-trained model, setting up an efficient pipeline for processing and classifying emotional content in text data.

4.3 Experiments settings

In our project, we utilized a standard Naive Bayes classifier for text classification, focusing on its probabilistic approach without specific architectural modifications. The primary emphasis was on preprocessing and feature extraction, particularly employing a Bag of Words model since Naive Bayes doesn't involve the same kind of hyperparameters as neural networks, our primary focus was on the preprocessing and feature extraction phase, particularly the use of a Bag of Words model. For the Naive Bayes experiment, we utilized Python and scikit-learn libraries, aligning with the model's non-reliance on GPU acceleration. This setup made the Naive Bayes experiments less hardware-intensive compared to our BERT model experiments, allowing us to focus more on the algorithm's efficacy with our dataset.

For BERT models experiments, we utilized various configurations of the BERT model. Initially, we used the standard bert-base-uncased model and bert-base-uncased-emotion to find their accuracy, then we focused on modifying the architecture of the bert-base-uncased model, specifically by varying the number of hidden transformer layers and finetuning. We experimented with configurations of 8, 12, and 16 layers to observe how changes in depth impact the model's performance. For hyperparameters, we consistently used the AdamW optimizer with a learning rate of 2e-5. Each modified version of the model was trained over 8 epochs using a batch size of 64. This approach allowed us to systematically assess the influence of the network's depth on its learning and predictive capabilities. Regarding the hardware and software, the experiments were conducted using PyTorch on a GPU-enabled setup, ensuring efficient processing. The software environment included the Hugging Face Transformers library, crucial for accessing and manipulating BERT models. This combination of hardware and software is essential for replicating our results, which, including the accuracy achieved by each variant, are detailed and analyzed in the subsequent section, providing insights into the optimal structural configuration for this specific task.

4.4 Experiment results

In our experiments, the performance of different models varied significantly. The pre-trained bert-base-uncased model had an initial accuracy of 0.35, while the pre-trained bert-base-uncased-emotion model, already fine-tuned for emotion detection, showed a much higher accuracy of 0.9265. Upon fine-tuning the bert-base-uncased model on our dataset, its accuracy improved remarkably to 0.929, nearly matching the pre-tuned emotion model. This fine-tuned variant also achieved a precision of 0.882, recall of 0.891, and an F1 score of 0.886, indicating a well-balanced performance across these metrics. In contrast, the Naive Bayes model showed a moderate accuracy of 0.7655, underscoring the enhanced capability of fine-tuned BERT models in emotion classification tasks. These results highlight the superiority of fine-tuned deep learning models like BERT in complex tasks like emotion classification compared to traditional machine learning models.

4.4.1 Performance of Naive Bayes vs. Pre-trained BERT

In Table 1, a comparative analysis is presented between Naive Bayes and two variants of the pre-trained BERT model: 'bert-base-uncased' and 'bhadresh-savani/bert-base-uncased-emotion'. Notably, the 'bert-base-uncased' model yielded the lowest accuracy at 35%. This underperformance is hypothesized to stem from the model's lack of specialized training on the 'emotion' dataset. Conversely, the 'bhadresh-savani/bert-base-uncased-emotion' model, which is specifically tailored for the emotion dataset, demonstrated superior performance with an accuracy of 92.65%. This significantly outpaced the Naive Bayes model, which achieved an accuracy of 76.55%. The superior performance of the BERT model in this comparison can be attributed to its advanced capabilities in handling context, semantics, and its specialized training for the specific dataset.

Models	Test Accuracy
Naive Bayes	76.55 %
Pre-trained BERT (base)	35 %
Pre-trained BERT ('emotion' specialized)	93 %
Modified base BERT (fine-tuned)	92.9%
Best modified base BERT (weights & number of layers)	89.7 %

Table 1: Comparision of the Accuracy of Naive Bayes and Pre-trained BERT

4.4.2 Performance of Pre-trained BERT vs. Our modified BERT

Upon initial testing, the bert-base-uncased model's performance on the emotion dataset's test set was relatively low, with an accuracy of just 35%. This modest performance can be largely attributed to the fact that some of the model's weights were not pre-initialized from the checkpoint and instead had to be initialized from scratch. To address this, we performed a fine-tuning process, adjusting the model's weights by training it specifically on the training set of the emotion dataset. Our objective was to adapt the model more closely to the nuances of emotion classification based on textual sentences. The modified BERT model has a marked improvement in performance, the model's accuracy surged to 92.9%. This significant leap in accuracy not only demonstrates the model's adaptability but also highlights the importance of domain-specific training in achieving high precision in specialized tasks. The results from this exercise reinforce the concept that while pre-trained models offer a strong foundation, their effectiveness in specific tasks can be greatly enhanced through careful and targeted fine-tuning on relevant datasets.

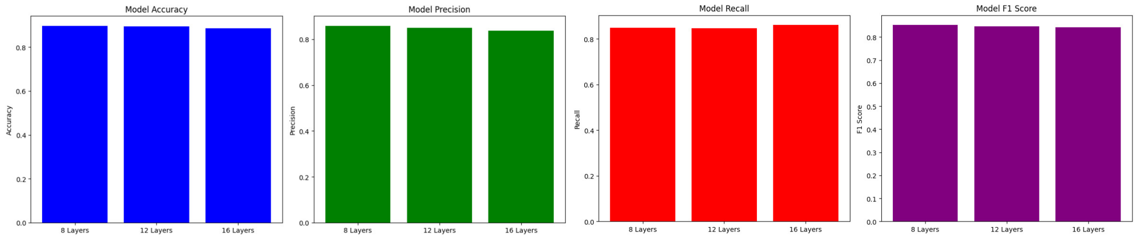


Figure 1: Performance Metrics of BERT Models with Varying Hidden Layers

Moreover, we also created three different BERT models by modifing the 'bert-base-uncased' model's architecture by setting the number of hidden layers to 8, 12, and 16 respectively in each configuration (config_bad8, config_bad12, config_bad16). These models are then instantiated using BertForSequenceClassification, which is a BERT model adapted for sequence classification tasks, with the custom configurations. The results from varying the number of hidden layers in our BERT-based models are visually represented in the provided bar charts(Figure 1). The models with 8, 12, and 16 layers demonstrate remarkably consistent performance across all metrics: accuracy, precision, recall, and F1 score. Each set of bars, corresponding to the different layer configurations, exhibits only minor variations, indicating that the number of hidden layers in this range does not significantly impact the model's ability to classify emotions within our dataset. This suggests that for our specific emotion classification task, model depth within the tested range is not a decisive factor for performance improvement.

Observing similar accuracy across BERT models with varying hidden layers (8, 12, 16) can be attributed to several factors. Primarily, if the complexity of the emotion classification task doesn't necessitate a deeper model's full capacity, simpler models with fewer layers might suffice in capturing the essential patterns. Additionally, the balance between overfitting and underfitting plays a crucial role; simpler models could be adequately complex for the given task, making extra layers redundant. Also, the dataset's size and quality may not fully leverage a more complex model's capabilities. Lastly, all models might have reached a performance plateau with effective training, beyond which additional layers don't significantly enhance performance. Essentially, the specific task and dataset may not require the added complexity of more hidden layers, resulting in similar performance across varying model architectures.

4.4.3 BERT Attention Matrix

We employed seaborn's heatmap() function to visualize attention matrices as heatmaps, the results of which are presented in the Appendix, Figures 1 to 16. The green heatmaps correspond to correctly predicted sentences, while red ones indicate incorrect predictions, with deeper colors signifying greater attention on specific tokens.

Upon close inspection of the attention matrices for the BERT model, we spotted a pattern wherein the model preferentially concentrates on emotional tokens such as "uncomfortable" and "rotten" in sentences it classifies correctly. This focused attention underscores the model's proficiency in identifying and weighing emotional language, which is pivotal in sentiment analysis tasks. For sentences predicted incorrectly, the attention is often inappropriately cast on neutral or contextually insignificant phrases like "dropped a" or "and will," suggesting a misalignment in the model's understanding of the text's emotional expressions. The attention matrices further reveal that the model sometimes neglects critical linguistic modifiers, which are essential for accurate sentiment interpretation. For example, failing to account for the word "not" could result in a positive sentiment prediction instead of a negative one. This observation indicates that while the model is adept at recognizing individual emotional words, its capacity to grasp modified sentiments or complex emotional expressions may need refinement.

Expanding on this analysis, future work might involve probing the model's attention across sentences of varying structures and lengths, exploring how attention is distributed between syntactic elements and semantic content. Such in-depth investigation could not only enhance the interpretability of BERT's attention mechanism but also inform targeted improvements for more precise language understanding.

4.5 Conclusion

In conclusion, our experiments demonstrate the superior performance of the specialized pre-trained BERT model in sentiment prediction tasks, with the 'emotion' dataset from dair-ai, when compared to the Naive Bayes classifier and other non-specialized base BERT models. The effectiveness of BERT lies in its ability to capture contextual information, a crucial aspect for understanding the complexities of language. Unlike the Naive Bayes classifier, which operates on a simplifying assumption of feature independence, BERT comprehensively considers the syntactic and semantic relationships between words within their sentence context.

Furthermore, the specialized pre-training of BERT on extensive text corpora relevant to a specific domain endows it with an enhanced understanding of language patterns and structures. This specialized knowledge is instrumental in performing sentiment analysis, as it enables BERT to discern nuanced language cues and contextual meanings, leading to more accurate sentiment predictions. On the other hand, the Naive Bayes classifier, despite its limitations in handling complex language structures, offers benefits in scenarios requiring a lightweight model with fast execution times. Its probabilistic approach, while simplistic, can be effective in tasks with well-defined feature relationships and limited contextual dependencies.

In contrast, while base versions of BERT offer general language analysis capabilities, they may not achieve the same level of performance in specialized tasks as their pre-trained counterparts. This highlights the importance of model selection and customization in natural language processing tasks, where the intricacies of language require models with both contextual understanding and domain-specific knowledge.

5 Appendix

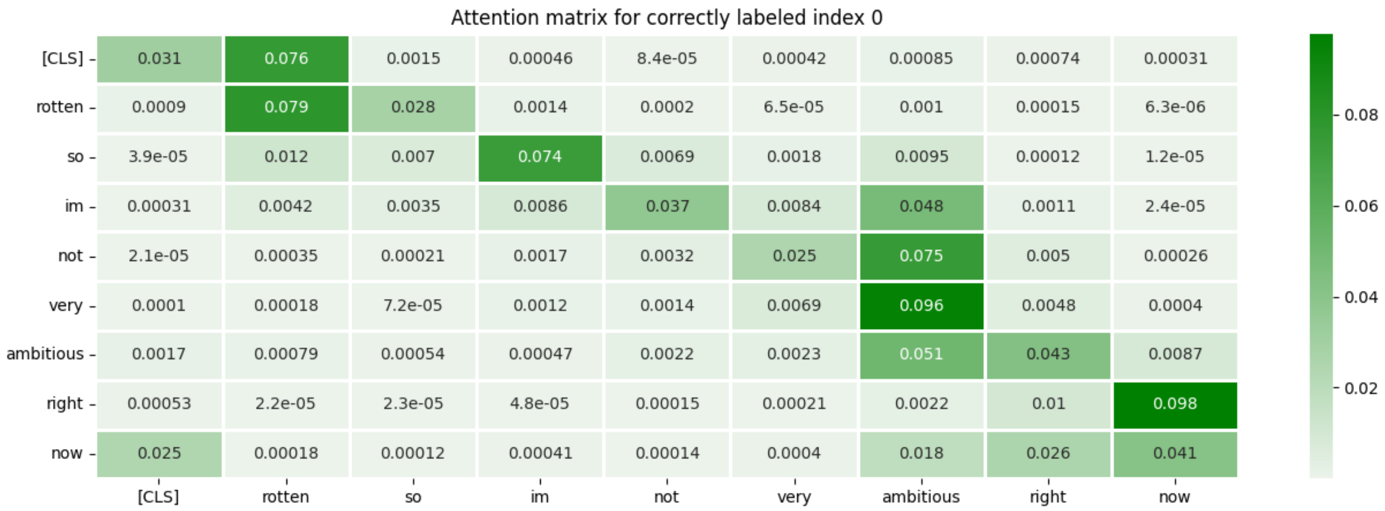


Figure 2: Attention Matrix of Correctly Predicted Sentence index 0

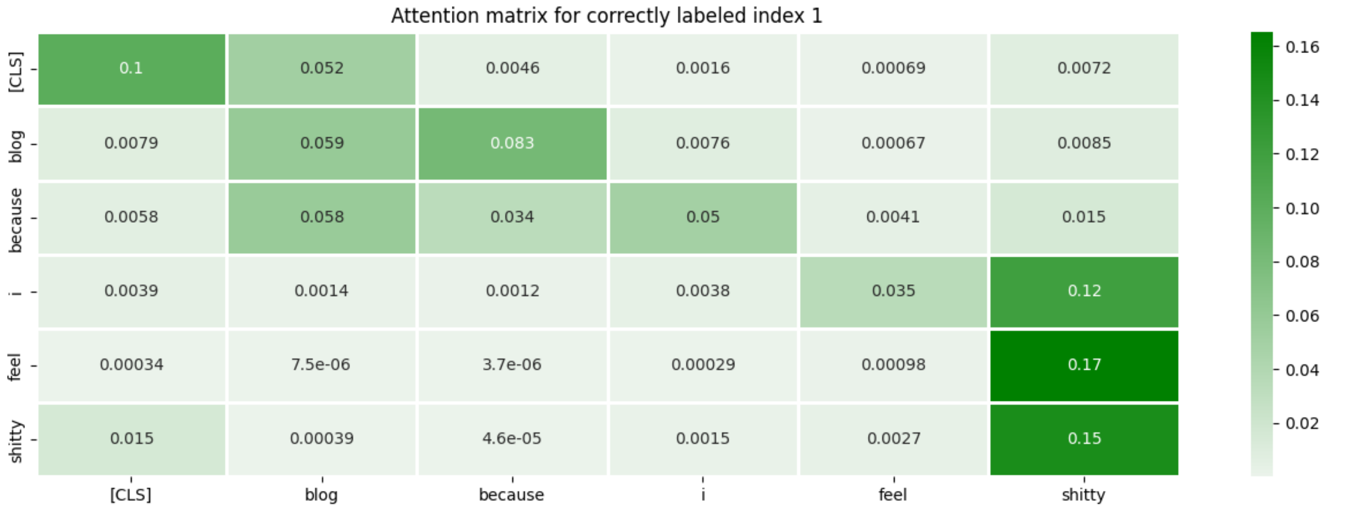


Figure 3: Attention Matrix of Correctly Predicted Sentence index 1

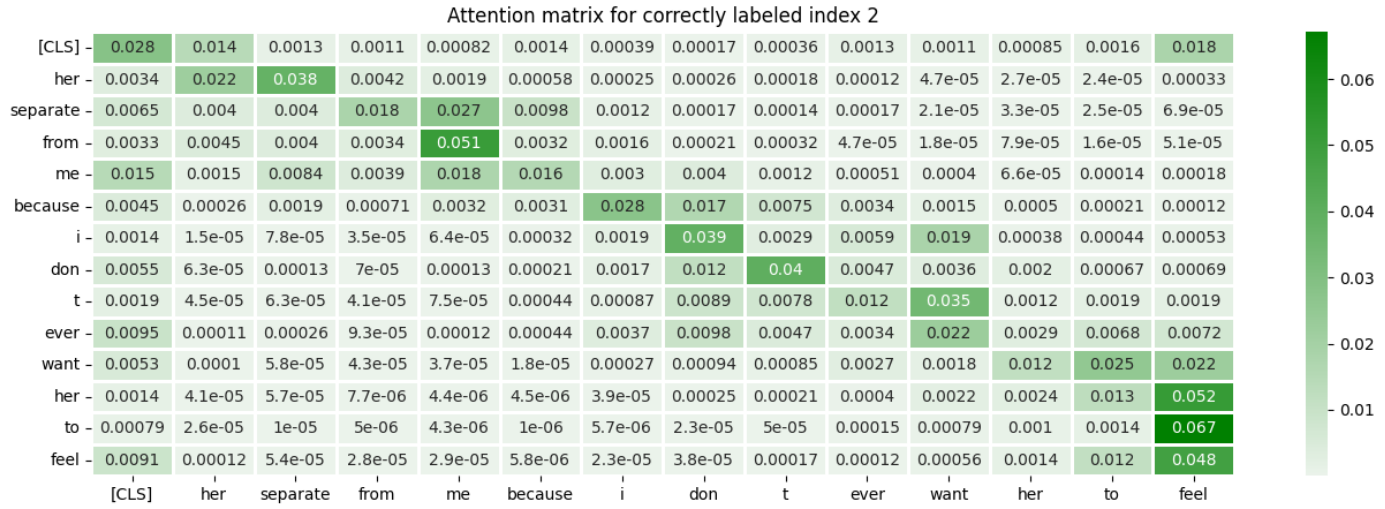
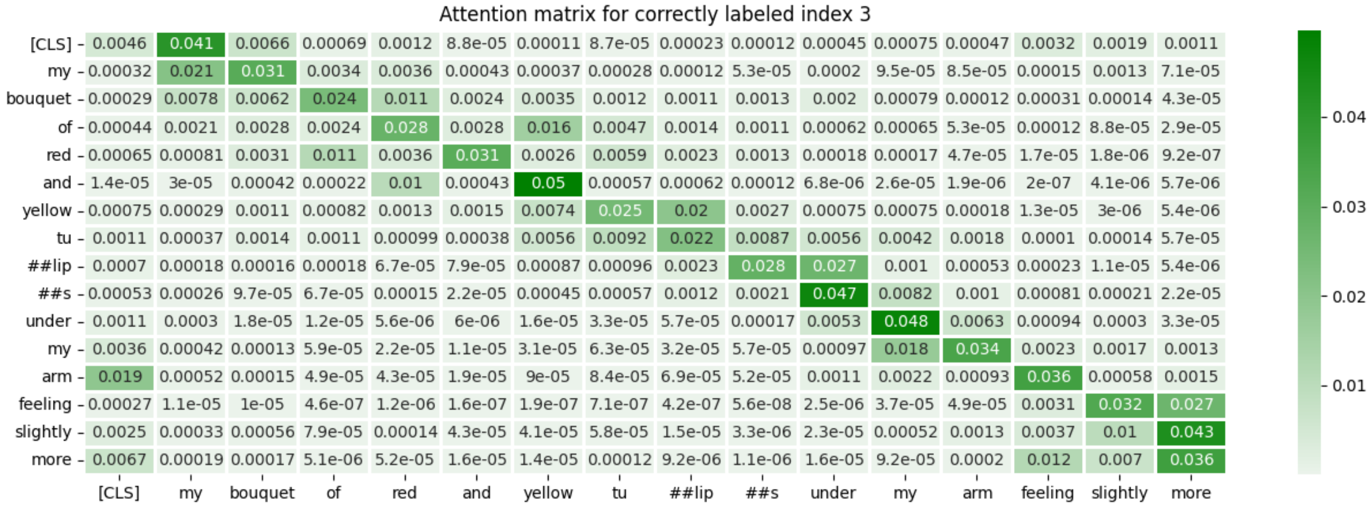
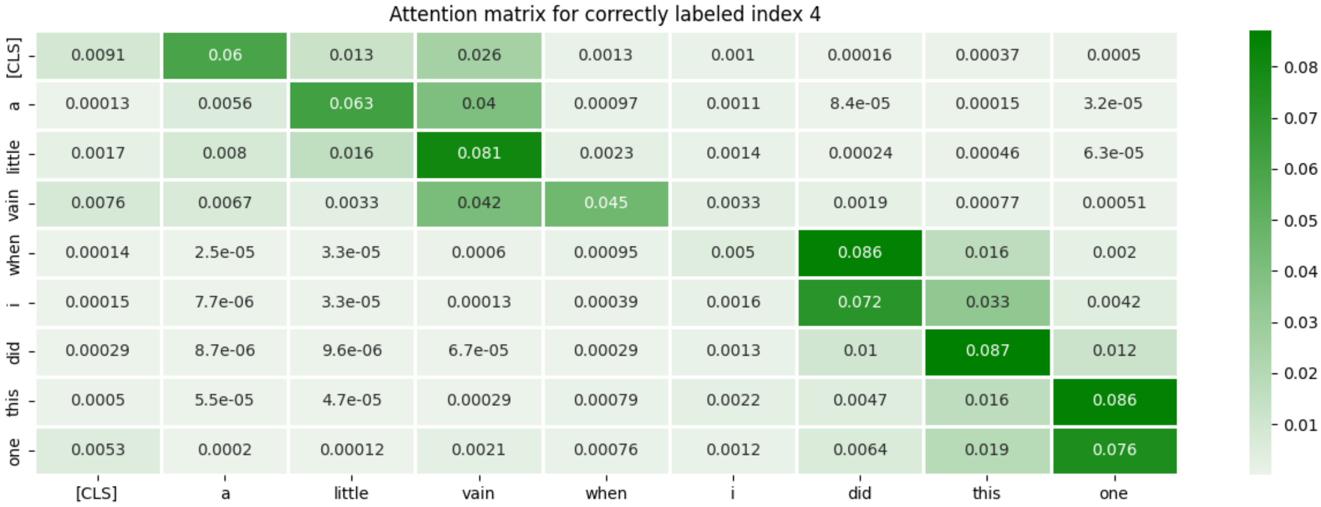


Figure 4: Attention Matrix of Correctly Predicted Sentence index 2



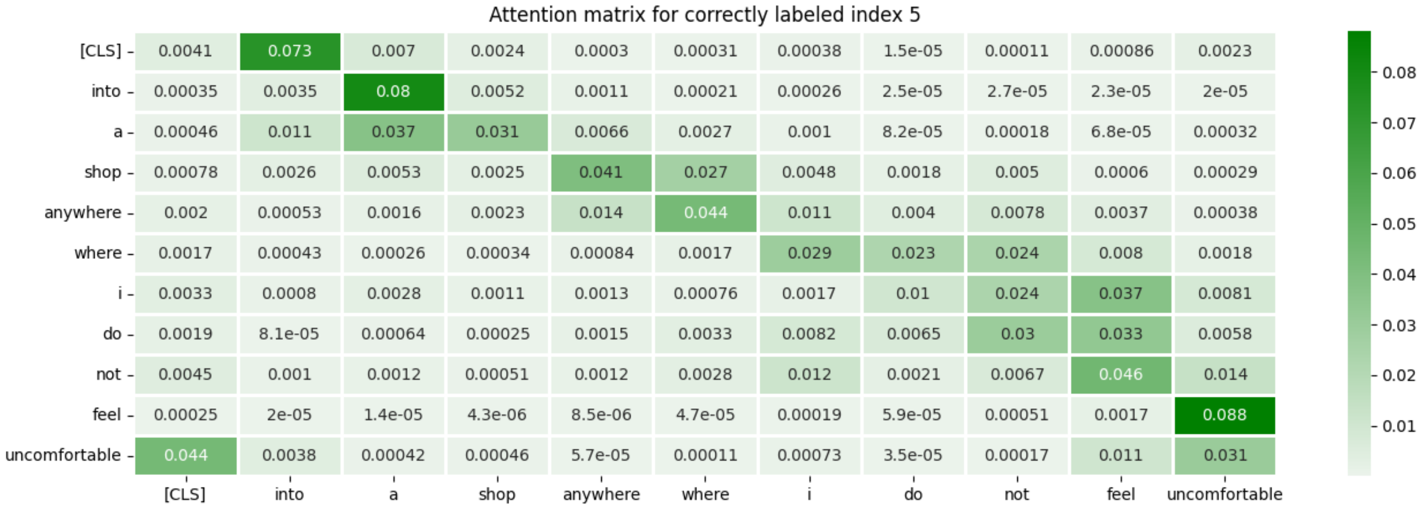
	[CLS]	my	bouquet	of	red	and	yellow	tu	##lip	##s	under	my	arm	feeling	slightly	more
[CLS]	0.0046	0.041	0.0066	0.00069	0.0012	8.8e-05	0.00011	8.7e-05	0.00023	0.00012	0.00045	0.00075	0.00047	0.0032	0.0019	0.0011
my	0.00032	0.021	0.031	0.0034	0.0036	0.00043	0.00037	0.00028	0.00012	5.3e-05	0.0002	9.5e-05	8.5e-05	0.00015	0.0013	7.1e-05
bouquet	0.00029	0.0078	0.0062	0.024	0.011	0.0024	0.0035	0.0012	0.0011	0.0013	0.002	0.00079	0.00012	0.00031	0.00014	4.3e-05
of	0.00044	0.0021	0.0028	0.0024	0.028	0.0028	0.016	0.0047	0.0014	0.0011	0.00062	0.00065	5.3e-05	0.00012	8.8e-05	2.9e-05
red	0.00065	0.00081	0.0031	0.011	0.0036	0.031	0.0026	0.0059	0.0023	0.0013	0.00018	0.00017	4.7e-05	1.7e-05	1.8e-06	9.2e-07
and	1.4e-05	3e-05	0.00042	0.00022	0.01	0.00043	0.05	0.00057	0.00062	0.00012	6.8e-06	2.6e-05	1.9e-06	2e-07	4.1e-06	5.7e-06
yellow	0.00075	0.00029	0.0011	0.00082	0.0013	0.0015	0.0074	0.025	0.02	0.0027	0.00075	0.00075	0.00018	1.3e-05	3e-06	5.4e-06
tu	0.0011	0.00037	0.0014	0.0011	0.00099	0.00038	0.0056	0.0092	0.022	0.0087	0.0056	0.0042	0.0018	0.0001	0.00014	5.7e-05
##lip	0.0007	0.00018	0.00016	0.00018	6.7e-05	7.9e-05	0.00087	0.00096	0.0023	0.028	0.027	0.001	0.00053	0.00023	1.1e-05	5.4e-06
##s	0.00053	0.00026	9.7e-05	6.7e-05	0.00015	2.2e-05	0.00045	0.00057	0.0012	0.0021	0.047	0.0082	0.001	0.00081	0.00021	2.2e-05
under	0.0011	0.0003	1.8e-05	1.2e-05	5.6e-06	6e-06	1.6e-05	3.3e-05	5.7e-05	0.00017	0.0053	0.048	0.0063	0.00094	0.0003	3.3e-05
my	0.0036	0.00042	0.00013	5.9e-05	2.2e-05	1.1e-05	3.1e-05	6.3e-05	3.2e-05	5.7e-05	0.00097	0.018	0.034	0.0023	0.0017	0.0013
arm	0.019	0.00052	0.00015	4.9e-05	4.3e-05	1.9e-05	9e-05	8.4e-05	6.9e-05	5.2e-05	0.0011	0.0022	0.00093	0.036	0.00058	0.0015
feeling	0.00027	1.1e-05	1e-05	4.6e-07	1.2e-06	1.6e-07	1.9e-07	7.1e-07	4.2e-07	5.6e-08	2.5e-06	3.7e-05	4.9e-05	0.0031	0.032	0.027
slightly	0.0025	0.00033	0.00056	7.9e-05	0.00014	4.3e-05	4.1e-05	5.8e-05	1.5e-05	3.3e-06	2.3e-05	0.00052	0.0013	0.0037	0.01	0.043
more	0.0067	0.00019	0.00017	5.1e-06	5.2e-05	1.6e-05	1.4e-05	0.00012	9.2e-06	1.1e-06	1.6e-05	9.2e-05	0.0002	0.012	0.007	0.036

Figure 5: Attention Matrix of Correctly Predicted Sentence index 3



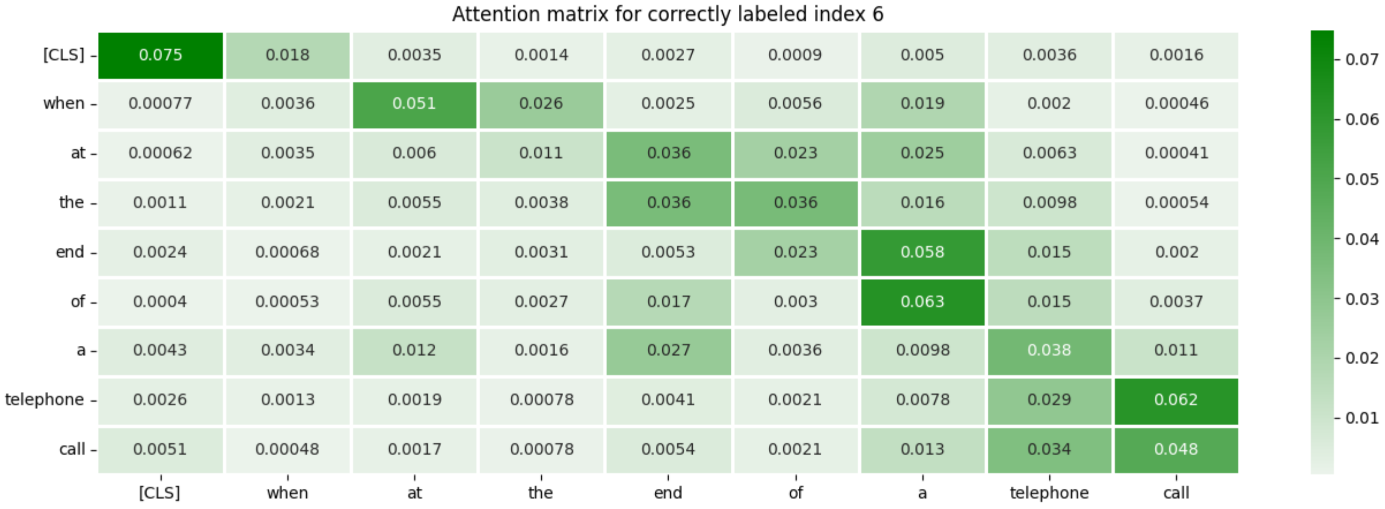
	[CLS]	a	little	vain	when	i	did	this	one
[CLS]	0.0091	0.06	0.013	0.026	0.0013	0.001	0.00016	0.00037	0.0005
a	0.00013	0.0056	0.063	0.04	0.00097	0.0011	8.4e-05	0.00015	3.2e-05
little	0.0017	0.008	0.016	0.081	0.0023	0.0014	0.00024	0.00046	6.3e-05
vain	0.0076	0.0067	0.0033	0.042	0.045	0.0033	0.0019	0.00077	0.00051
when	0.00014	2.5e-05	3.3e-05	0.0006	0.00095	0.005	0.086	0.016	0.002
i	0.00015	7.7e-06	3.3e-05	0.00013	0.00039	0.0016	0.072	0.033	0.0042
did	0.00029	8.7e-06	9.6e-06	6.7e-05	0.00029	0.0013	0.01	0.087	0.012
this	0.0005	5.5e-05	4.7e-05	0.00029	0.00079	0.0022	0.0047	0.016	0.086
one	0.0053	0.0002	0.00012	0.0021	0.00076	0.0012	0.0064	0.019	0.076

Figure 6: Attention Matrix of Correctly Predicted Sentence index 4



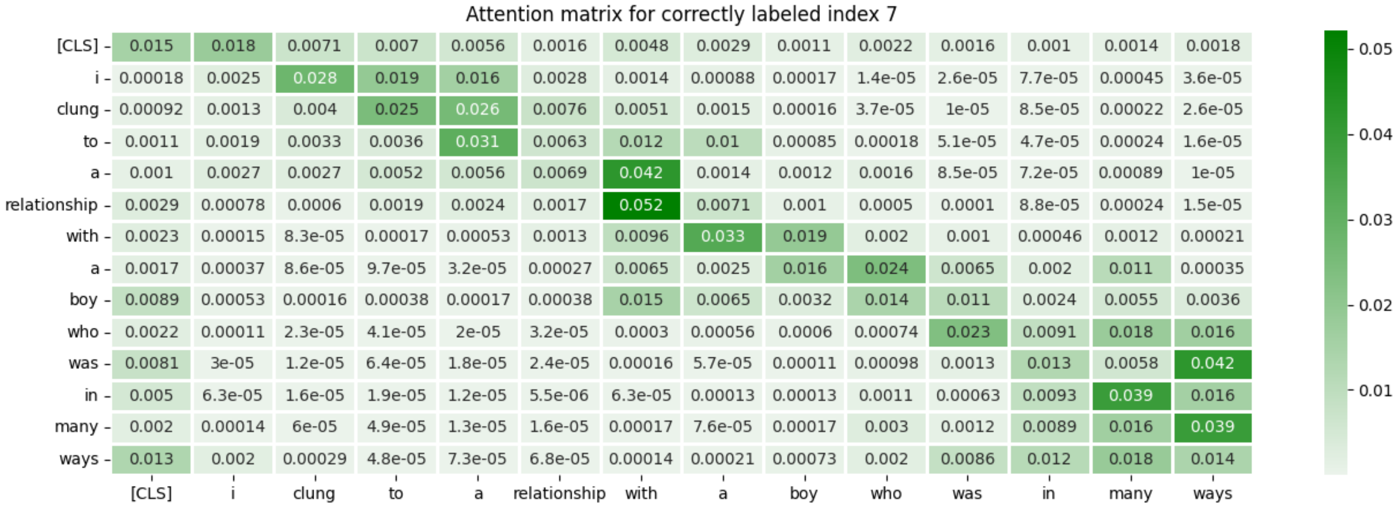
	[CLS]	into	a	shop	anywhere	where	i	do	not	feel	uncomfortable
[CLS]	0.0041	0.073	0.007	0.0024	0.0003	0.00031	0.00038	1.5e-05	0.00011	0.00086	0.0023
into	0.00035	0.0035	0.08	0.0052	0.0011	0.00021	0.00026	2.5e-05	2.7e-05	2.3e-05	2e-05
a	0.00046	0.011	0.037	0.031	0.0066	0.0027	0.001	8.2e-05	0.00018	6.8e-05	0.00032
shop	0.00078	0.0026	0.0053	0.0025	0.041	0.027	0.0048	0.0018	0.005	0.0006	0.00029
anywhere	0.002	0.00053	0.0016	0.0023	0.014	0.044	0.011	0.004	0.0078	0.0037	0.00038
where	0.0017	0.00043	0.00026	0.00034	0.00084	0.0017	0.029	0.023	0.024	0.008	0.0018
i	0.0033	0.0008	0.0028	0.0011	0.0013	0.00076	0.0017	0.01	0.024	0.037	0.0081
do	0.0019	8.1e-05	0.00064	0.00025	0.0015	0.0033	0.0082	0.0065	0.03	0.033	0.0058
not	0.0045	0.001	0.0012	0.00051	0.0012	0.0028	0.012	0.0021	0.0067	0.046	0.014
feel	0.00025	2e-05	1.4e-05	4.3e-06	8.5e-06	4.7e-05	0.00019	5.9e-05	0.00051	0.0017	0.088
uncomfortable	0.044	0.0038	0.00042	0.00046	5.7e-05	0.00011	0.00073	3.5e-05	0.00017	0.011	0.031

Figure 7: Attention Matrix of Correctly Predicted Sentence index 5



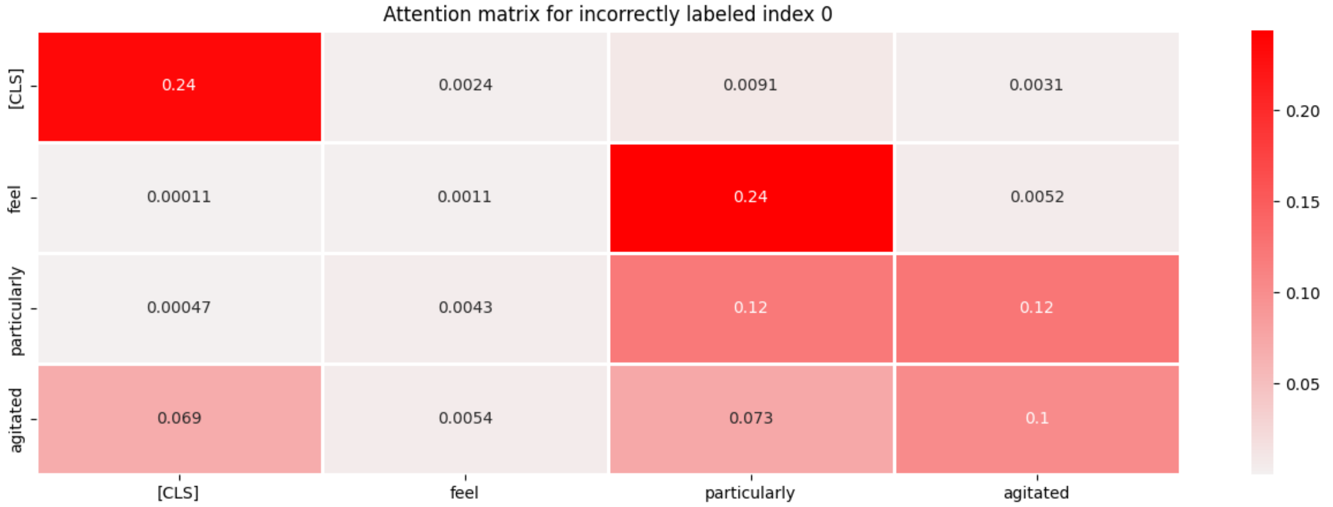
	[CLS]	when	at	the	end	of	a	telephone	call
[CLS]	0.075	0.018	0.0035	0.0014	0.0027	0.0009	0.005	0.0036	0.0016
when	0.00077	0.0036	0.051	0.026	0.0025	0.0056	0.019	0.002	0.00046
at	0.00062	0.0035	0.006	0.011	0.036	0.023	0.025	0.0063	0.00041
the	0.0011	0.0021	0.0055	0.0038	0.036	0.036	0.016	0.0098	0.00054
end	0.0024	0.00068	0.0021	0.0031	0.0053	0.023	0.058	0.015	0.002
of	0.0004	0.00053	0.0055	0.0027	0.017	0.003	0.063	0.015	0.0037
a	0.0043	0.0034	0.012	0.0016	0.027	0.0036	0.0098	0.038	0.011
telephone	0.0026	0.0013	0.0019	0.00078	0.0041	0.0021	0.0078	0.029	0.062
call	0.0051	0.00048	0.0017	0.00078	0.0054	0.0021	0.013	0.034	0.048

Figure 8: Attention Matrix of Correctly Predicted Sentence index 6



	[CLS]	i	clung	to	a	relationship	with	a	boy	who	was	in	many	ways
[CLS]	0.015	0.018	0.0071	0.007	0.0056	0.0016	0.0048	0.0029	0.0011	0.0022	0.0016	0.001	0.0014	0.0018
i	0.00018	0.0025	0.028	0.019	0.016	0.0028	0.0014	0.00088	0.00017	1.4e-05	2.6e-05	7.7e-05	0.00045	3.6e-05
clung	0.00092	0.0013	0.004	0.025	0.026	0.0076	0.0051	0.0015	0.00016	3.7e-05	1e-05	8.5e-05	0.00022	2.6e-05
to	0.0011	0.0019	0.0033	0.0036	0.031	0.0063	0.012	0.01	0.00085	0.00018	5.1e-05	4.7e-05	0.00024	1.6e-05
a	0.001	0.0027	0.0027	0.0052	0.0056	0.0069	0.042	0.0014	0.0012	0.0016	8.5e-05	7.2e-05	0.00089	1e-05
relationship	0.0029	0.00078	0.0006	0.0019	0.0024	0.0017	0.052	0.0071	0.001	0.0005	0.0001	8.8e-05	0.00024	1.5e-05
with	0.0023	0.00015	8.3e-05	0.00017	0.00053	0.0013	0.0096	0.033	0.019	0.002	0.001	0.00046	0.0012	0.00021
a	0.0017	0.00037	8.6e-05	9.7e-05	3.2e-05	0.00027	0.0065	0.0025	0.016	0.024	0.0065	0.002	0.011	0.00035
boy	0.0089	0.00053	0.00016	0.00038	0.00017	0.00038	0.015	0.0065	0.0032	0.014	0.011	0.0024	0.0055	0.0036
who	0.0022	0.00011	2.3e-05	4.1e-05	2e-05	3.2e-05	0.0003	0.00056	0.0006	0.00074	0.023	0.0091	0.018	0.016
was	0.0081	3e-05	1.2e-05	6.4e-05	1.8e-05	2.4e-05	0.00016	5.7e-05	0.00011	0.00098	0.0013	0.013	0.0058	0.042
in	0.005	6.3e-05	1.6e-05	1.9e-05	1.2e-05	5.5e-06	6.3e-05	0.00013	0.00013	0.0011	0.00063	0.0093	0.039	0.016
many	0.002	0.00014	6e-05	4.9e-05	1.3e-05	1.6e-05	0.00017	7.6e-05	0.00017	0.003	0.0012	0.0089	0.016	0.039
ways	0.013	0.002	0.00029	4.8e-05	7.3e-05	6.8e-05	0.00014	0.00021	0.00073	0.002	0.0086	0.012	0.018	0.014

Figure 9: Attention Matrix of Correctly Predicted Sentence index 7



	[CLS]	feel	particularly	agitated
[CLS]	0.24	0.0024	0.0091	0.0031
feel	0.00011	0.0011	0.24	0.0052
particularly	0.00047	0.0043	0.12	0.12
agitated	0.069	0.0054	0.073	0.1

Figure 10: Attention Matrix of Incorrectly Predicted Sentence index 0

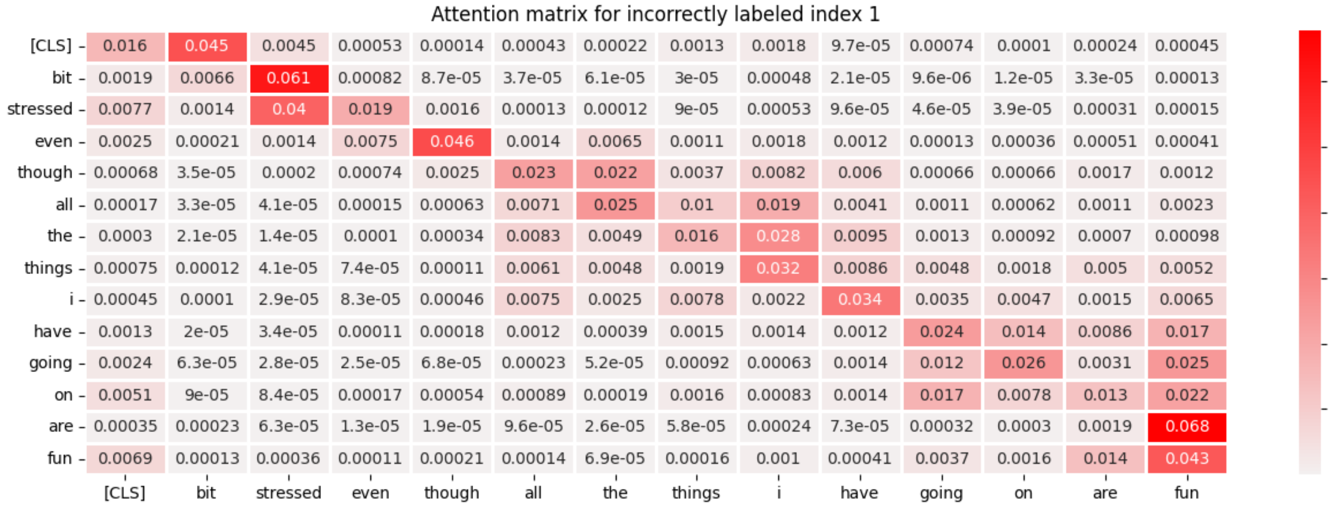


Figure 11: Attention Matrix of Incorrectly Predicted Sentence index 1

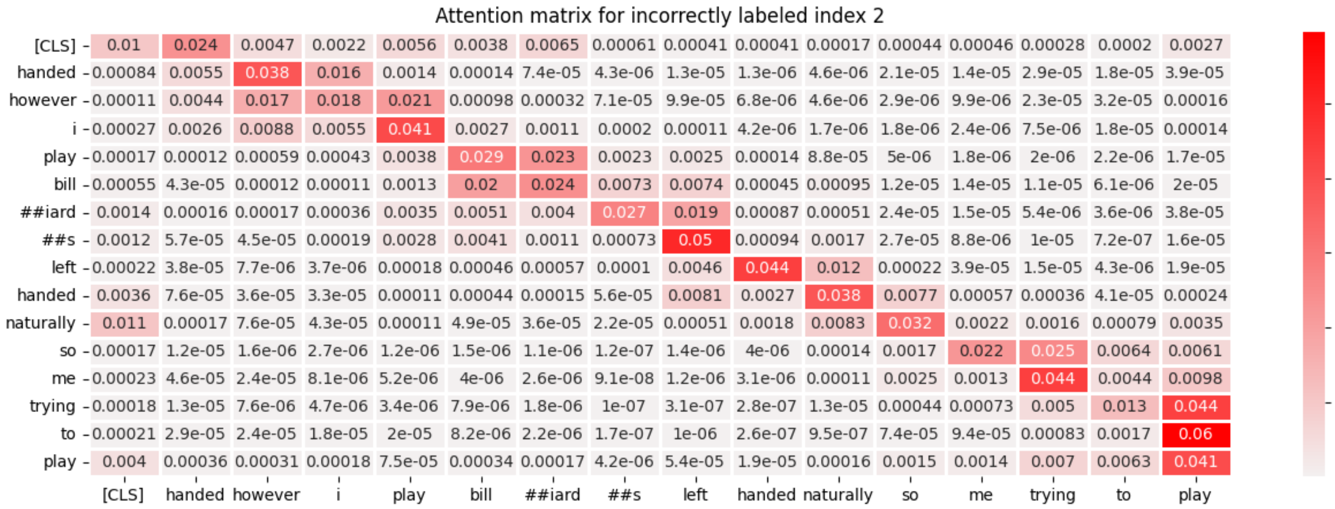


Figure 12: Attention Matrix of Incorrectly Predicted Sentence index 2

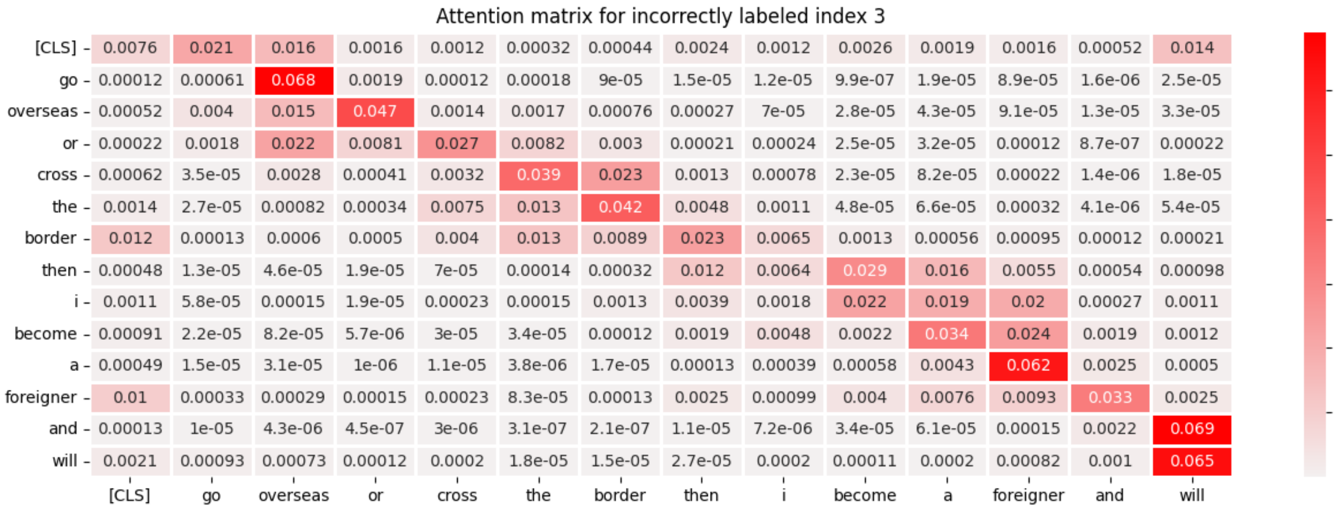


Figure 13: Attention Matrix of Incorrectly Predicted Sentence index 3

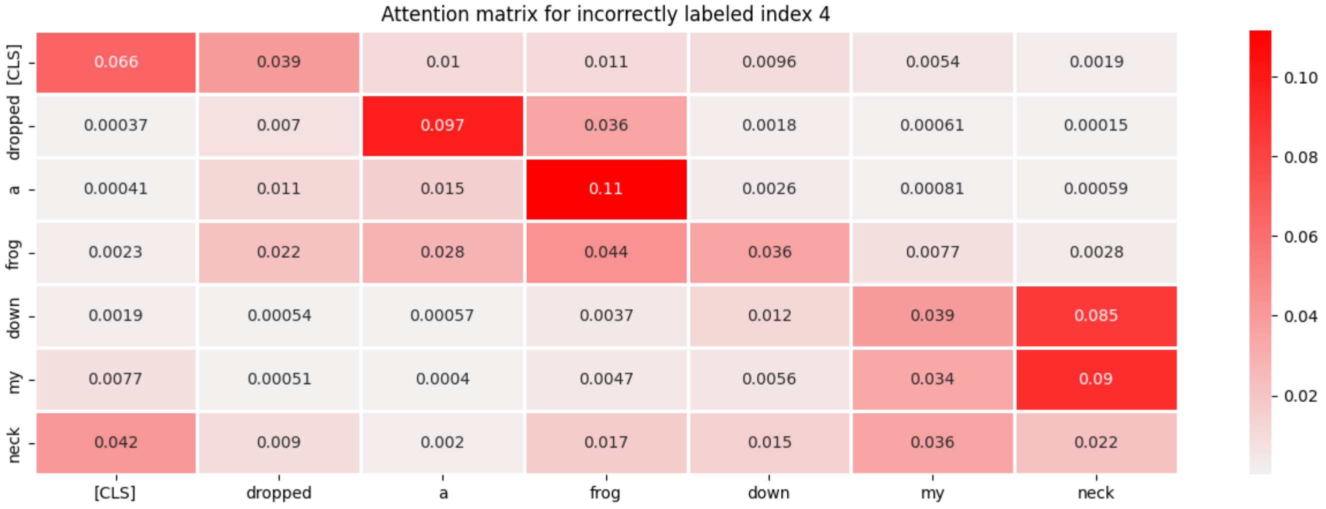


Figure 14: Attention Matrix of Incorrectly Predicted Sentence index 4

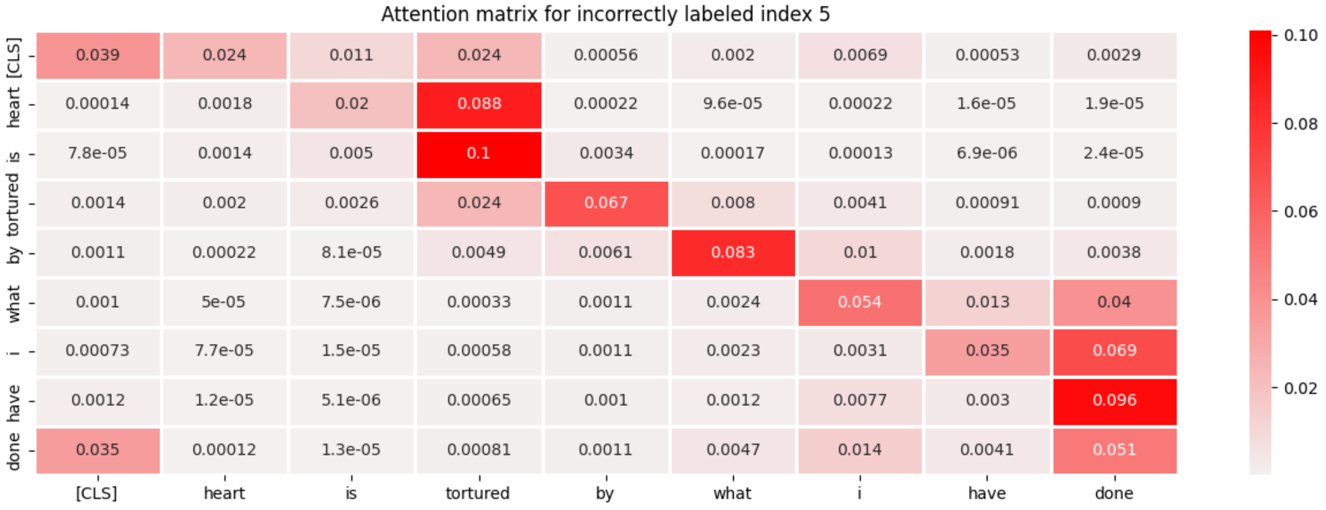


Figure 15: Attention Matrix of Incorrectly Predicted Sentence index 5

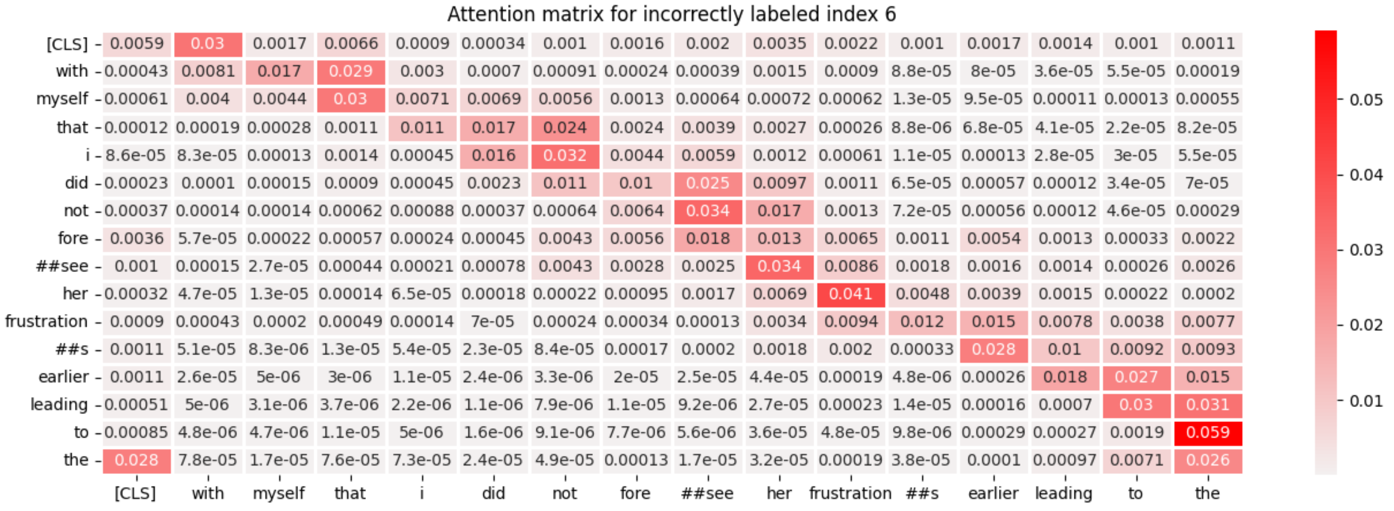
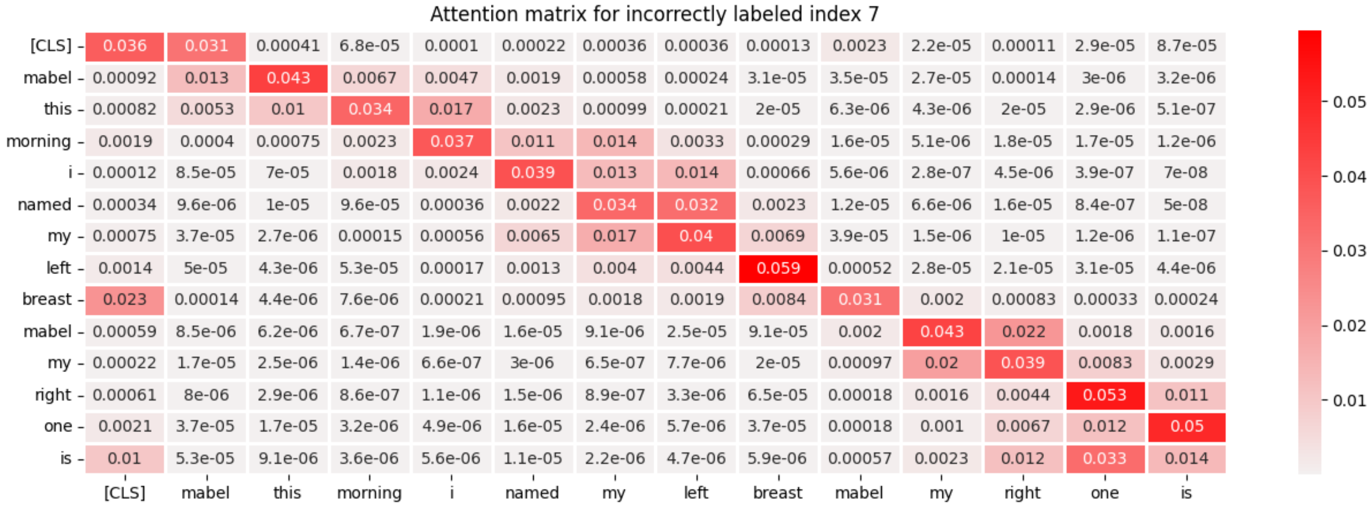


Figure 16: Attention Matrix of Incorrectly Predicted Sentence index 6



	[CLS]	mabel	this	morning	i	named	my	left	breast	mabel	my	right	one	is
[CLS]	0.036	0.031	0.00041	6.8e-05	0.0001	0.00022	0.00036	0.00036	0.00013	0.0023	2.2e-05	0.00011	2.9e-05	8.7e-05
mabel	0.00092	0.013	0.043	0.0067	0.0047	0.0019	0.00058	0.00024	3.1e-05	3.5e-05	2.7e-05	0.00014	3e-06	3.2e-06
this	0.00082	0.0053	0.01	0.034	0.017	0.0023	0.00099	0.00021	2e-05	6.3e-06	4.3e-06	2e-05	2.9e-06	5.1e-07
morning	0.0019	0.0004	0.00075	0.0023	0.037	0.011	0.014	0.0033	0.00029	1.6e-05	5.1e-06	1.8e-05	1.7e-05	1.2e-06
i	0.00012	8.5e-05	7e-05	0.0018	0.0024	0.039	0.013	0.014	0.00066	5.6e-06	2.8e-07	4.5e-06	3.9e-07	7e-08
named	0.00034	9.6e-06	1e-05	9.6e-05	0.00036	0.0022	0.034	0.032	0.0023	1.2e-05	6.6e-06	1.6e-05	8.4e-07	5e-08
my	0.00075	3.7e-05	2.7e-06	0.00015	0.00056	0.0065	0.017	0.04	0.0069	3.9e-05	1.5e-06	1e-05	1.2e-06	1.1e-07
left	0.0014	5e-05	4.3e-06	5.3e-05	0.00017	0.0013	0.004	0.0044	0.059	0.00052	2.8e-05	2.1e-05	3.1e-05	4.4e-06
breast	0.023	0.00014	4.4e-06	7.6e-06	0.00021	0.00095	0.0018	0.0019	0.0084	0.031	0.002	0.00083	0.00033	0.00024
mabel	0.00059	8.5e-06	6.2e-06	6.7e-07	1.9e-06	1.6e-05	9.1e-06	2.5e-05	9.1e-05	0.002	0.043	0.022	0.0018	0.0016
my	0.00022	1.7e-05	2.5e-06	1.4e-06	6.6e-07	3e-06	6.5e-07	7.7e-06	2e-05	0.00097	0.02	0.039	0.0083	0.0029
right	0.00061	8e-06	2.9e-06	8.6e-07	1.1e-06	1.5e-06	8.9e-07	3.3e-06	6.5e-05	0.00018	0.0016	0.0044	0.053	0.011
one	0.0021	3.7e-05	1.7e-05	3.2e-06	4.9e-06	1.6e-05	2.4e-06	5.7e-06	3.7e-05	0.00018	0.001	0.0067	0.012	0.05
is	0.01	5.3e-05	9.1e-06	3.6e-06	5.6e-06	1.1e-05	2.2e-06	4.7e-06	5.9e-06	0.00057	0.0023	0.012	0.033	0.014

Figure 17: Attention Matrix of Incorrectly Predicted Sentence index 7

6 References

1. bert-base-uncased-emotion, https://huggingface.co/bhadresh-savani/bert-base-uncased-emotion

2. bert-base-uncased, https://huggingface.co/bert-base-uncased

3. Luo, L., Wang, Y. (Year). EmotionX-HSU: Adopting Pre-trained BERT for Emotion Classification.

Acronyms

BERT Bidirectional Encoder Representations from Transformers.

CNN Convolutional Neural Network.

NLP Natural Language Processing.