Title of your project

Stanford CS224N Default Project

Betty Wu
Department of Computer Science
Stanford University
bettyyw@stanford.edu

Abstract

This project focuses on developing a multi-task learning approach using the BERT model to simultaneously address three sentence-level tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. The model architecture is based on the miniBERT backbone, with task-specific modifications for each task. The sentiment analysis head remains unchanged from the default project, while the paraphrase detection task employs a Siamese network structure with concatenated sentence embeddings and a softmax function for binary classification. The semantic textual similarity task utilizes a similar architecture, with cosine similarity calculated between sentence embeddings and scaled to obtain the final output. The model is trained using a multi-task learning approach, combining the losses from all three tasks during training. Experimental results demonstrate the effectiveness of the proposed approach, with the best-performing model achieving an overall score of 0.738 on the test set and task-specific scores of 0.499, 0.804, and 0.822 for sentiment analysis, paraphrase detection, and semantic textual similarity. The project highlights the potential of multi-task learning with BERT for enhancing the performance of natural language understanding tasks.

- TA mentor: Johnny Chang. No external collaborators, no external mentor, not sharing project

1 Introduction

Multi-task learning has emerged as a promising approach to improve the performance and efficiency of natural language understanding models by leveraging shared knowledge and representations across different tasks [1]. This project explores the application of multi-task learning using the BERT (Bidirectional Encoder Representations from Transformers) model [2] to simultaneously address three sentence-level tasks: sentiment analysis, paraphrase detection, and semantic textual similarity.

The proposed approach builds upon the miniBERT backbone, which serves as the foundation for the multi-task learning framework. Task-specific modifications are made to the model architecture to accommodate the unique requirements of each task. For sentiment analysis, the head architecture remains unchanged from the default project. The paraphrase detection task employs a Siamese network structure, where the embeddings of two sentences are concatenated along with their absolute difference and passed through a fully connected layer followed by a softmax function for binary classification. Similarly, the semantic textual similarity task utilizes a Siamese network, with the cosine similarity calculated between the sentence embeddings and scaled to obtain the final output.

The model is trained using a multi-task learning approach, where the losses from all three tasks are combined during training. The total loss is calculated as the sum of the individual task losses, allowing for joint optimization across the tasks. Experimental results demonstrate the effectiveness of the proposed approach, with the best-performing model achieving an overall score of 0.738 on the test set and task-specific scores of 0.499, 0.804, and 0.822 for sentiment analysis, paraphrase detection, and semantic textual similarity, respectively.

The main contributions of this project are as follows:

- We propose a multi-task learning framework using the BERT model for simultaneously addressing sentiment analysis, paraphrase detection, and semantic textual similarity tasks.
- We design task-specific architectures and modifications to accommodate the unique requirements of each task within the multi-task learning framework.
- We demonstrate the effectiveness of the proposed approach through extensive experiments and evaluation using various metrics specific to each task.

2 Related Works

The landscape of multi-task learning in Natural Language Processing (NLP) has seen significant advancements, with various approaches and methodologies being explored to optimize performance across different tasks. Caruana (1997) laid the foundation for multi-task learning by suggesting that sharing representations across tasks can enhance generalization [3]. However, the challenges of conflicting gradients and task relatedness have necessitated the development of specialized optimization methods. Sener and Koltun proposed solving for Pareto optimal solutions, while Yu et al. introduced gradient surgery to mitigate conflicting task gradients [4] [5].

The efficacy of multi-task learning was further investigated by Liu et al. through the Multi-Task Deep Neural Network (MT-DNN), which incorporated BERTLARGE [6]. This study demonstrated that MT-DNN outperforms the BERT baseline by leveraging cross-task data and benefiting from regularization effects. Additionally, Reimers and Gurevych addressed limitations in single-task settings by introducing Sentence-BERT (SBERT), which facilitates the derivation of semantically meaningful sentence embeddings through siamese and triplet networks [7].

Another critical advancement was the introduction of the SMART framework by Jiang et al., which integrates smooth-inducing adversarial regularization into the loss function to combat overfitting during the fine-tuning phase [8]. This method has proven effective in maintaining model robustness across various tasks. Sun et al. also highlighted the importance of additional pre-training, showing that stages involving masked language modeling and next-sentence prediction before fine-tuning can significantly enhance performance [9].

In the realm of post-training techniques, methods such as Stochastic Weight Averaging (SWA) and knowledge distillation have been explored to improve generalization. Izmailov et al. demonstrated the benefits of SWA in vision models, suggesting its potential applicability in NLP models [10]. Furthermore, Allen-Zhu discussed the theoretical underpinnings and benefits of self-distillation, which can sometimes outperform traditional ensemble methods [11].

While the original BERT paper (Devlin et al.) demonstrated BERT's capability in achieving state-of-the-art performance across eleven NLP tasks, it did not explore its effectiveness as a multi-task learning model [2]. This aspect is crucial for comprehending BERT's versatility and potential in real-world applications spanning various NLP domains. Liu et al. (2019) filled this research gap by presenting a Multi-Task Deep Neural Network (MT-DNN) with the BERTLARGE model incorporated [6]. They found that MT-DNN consistently outperformed the BERT baseline, showing BERT's applicability and high performance as a multi-task learning model. They believe that it is because MT-DNN can leverage large amounts of cross-task data and benefit from a regularization effect which leads to more general representations to help adapt to new tasks and domains.

3 Approach

The primary objective of this project is to develop a multi-task learning approach using BERT to tackle three sentence-level tasks simultaneously: sentiment analysis, paraphrase detection, and semantic textual similarity.

The baseline model for this project involves adding a single linear layer for each task. For sentiment analysis, the model consists of a BERT layer followed by a linear layer that maps the hidden size to the number of sentiment classes. The output is passed through a softmax function to obtain the final prediction, with cross-entropy loss. For paraphrase detection, the model uses a similar structure with a linear layer mapping to two classes, followed by a softmax function and cross-entropy loss. The

semantic similarity task employs a linear layer to map the hidden size to a single output, followed by a sigmoid function scaled by 5, with mean squared error (MSE) as the loss function. Each task is trained sequentially, fine-tuning only the last linear layer.

Our proposed model enhances this structure by incorporating deeper and more complex layers for each task. For sentiment analysis, the BERT layer is followed by two linear layers, each mapping the hidden size to another hidden size, before mapping to the sentiment classes and applying a softmax function. For paraphrase detection, a Siamese network structure is employed. The model processes two statements independently through BERT, concatenates their hidden representations along with their absolute difference, and passes this through three linear layers before the final softmax prediction, using cross-entropy loss. The semantic textual similarity task processes the input statements through BERT, then through three linear layers, and finally calculates the cosine similarity of the outputs, applying a ReLU activation and scaling by 5, with MSE as the loss function.

To address the varying sizes of datasets for the three tasks, we randomly select 1000 samples from each dataset to create balanced subsets for training. The model is trained using a multi-task learning approach, where the losses from all three tasks are combined during training. The total loss is calculated as the sum of the individual task losses:

$$L = L1 + L2 + L3$$

where $L1$, $L2$, and $L3$ represent the losses for sentiment analysis, paraphrase detection, and semantic textual similarity, respectively.

The dropout method is implemented to mitigate overfitting by introducing regularization. Specifically, a dropout layer is applied to the pooled BERT embeddings before they are passed to the task-specific heads. By randomly zeroing out a fraction of the input units during training, dropout prevents the model from relying too heavily on any individual neuron, thus promoting robustness and generalization across the tasks. This approach helps in achieving better performance on unseen data by reducing overfitting during the training phase.

During training, the model alternates between training on the three tasks, allowing for knowledge sharing and joint optimization across the tasks. This method enhances the model's ability to generalize by leveraging shared representations learned from multiple tasks, leading to improved performance across the board.

4 Experiments

4.1 Data

- Sentiment analysis: SST dataset
- Paraphrase detection: Quora dataset
- Semantic textual similarity: SemEval dataset

4.2 Evaluation method

- Sentiment classification: Classification accuracy
- Paraphrase detection: Binary prediction accuracy
- Semantic textual similarity: Pearson correlation

4.3 Experimental details

The experiments are conducted using the pre-trained BERT model as the backbone for the multi-task learning approach. The pre-trained BERT model has a hidden size of 768 and consists of 12 layers. The dimension of the task-specific layers in the parameter adaptation methods is set to 204. The total number of parameters for each parameter adaptation method is calculated based on the number of tasks (T=3) and the dimensions of the BERT model (d_m) and task-specific layers (d_s).

All models are trained using a learning rate of $1e-5$, a batch size of 32, and for 20 epochs. The AdamW optimizer is used for training. The loss functions employed for each task are as follows:

cross-entropy loss for sentiment analysis, cross-entropy loss with softmax function for paraphrase detection, and Mean Square Error (MSE) loss for semantic textual similarity.

4.4 Results

Here are the experimental results on the development set (Table 1) and the test set (Table 2). Comparing them, we have a few observations:

Metric	Baseline	Model
SST accuracy	0.505	0.456 (-0.049)
Paraphrase accuracy	0.746	0.798 (+0.052)
STS correlation	0.364	0.835 (+0.471)
Overall score	0.644	0.724 (+0.080)

Table 1: Experiment result on dev set

Metric	Baseline	Model
SST accuracy	0.512	0.499 (-0.014)
Paraphrase accuracy	0.746	0.804 (+0.058)
STS correlation	0.339	0.822 (+0.483)
Overall score	0.643	0.738 (+0.095)

Table 2: Experiment result on test set

The results obtained from the experiments are somewhat mixed compared to my initial expectations. The performance of the model on the paraphrase detection and semantic textual similarity tasks is quite impressive and exceeds what I had anticipated. The significant improvements in accuracy and correlation scores for these tasks suggest that the multi-task learning approach, combined with the Siamese network structure and cosine similarity, is highly effective in capturing sentence-level semantic information.

However, the slightly lower performance on the sentiment analysis task compared to the baseline is a bit surprising and falls short of my expectations. I had hoped that the shared representations learned from the other tasks would also benefit sentiment classification, but the reduction was small. This suggests that while multi-task learning offers advantages in terms of knowledge sharing and joint optimization, it may also introduce complexity that slightly hinders the performance of individual tasks. I expect further fine-tuning of the model architecture and hyperparameters specific to the sentiment analysis task might help in mitigating this issue and achieving better overall performance.

This outcome indicates that while the model is promising for multi-task learning, there is still room for improvement in terms of balancing the performance across all tasks. It highlights the challenges of effectively training a model to excel in multiple tasks simultaneously, especially when the tasks have different characteristics and requirements. Meanwhile, it's worth noting that the model's performance is relatively consistent between the dev and test sets. The differences in metrics between the dev and test sets are small, suggesting that the model generalizes well to unseen data.

5 Analysis

I generated the confusion matrices for the SST task and Paraphrase task.

Figure 1 in Appendix presents a confusion matrix for the sentiment analysis task. The matrix shows the distribution of predicted labels against the true labels. The diagonal elements represent the correctly classified instances, while the off-diagonal elements represent misclassifications. The model performs relatively well in predicting the positive sentiment class (label 1), with 12,580 correctly classified instances. However, it struggles with the negative sentiment class (label 0), misclassifying a significant number of instances as positive. This suggests that the model may have a bias towards predicting positive sentiment and could benefit from further refinement to improve its performance on the negative class.

Figure 2 in Appendix displays a confusion matrix for the paraphrase detection task. The matrix reveals that the model accurately predicts a large number of true negative instances (label 0), with

88 correctly classified instances in the bottom-right cell. However, it also misclassifies some true positive instances (label 1) as negative, as shown in the bottom-left cell. The model's performance on true positive instances is better than on true negative instances, with a higher number of correctly classified instances in the top-left cell compared to the bottom-right cell.

The STS task is a regression task, and I have calculated the metrics listed below:
sts: corrcoef 0.8350568814918248
sts: mes 0.7362831745373741
sts: rmes 0.8580694462206274
sts: msle 0.08650326926682565
sts: median absolute error 0.5242195129394531
sts: mean absolute error 0.656100481949031
sts: explained variance score 0.6890392944845571
sts: r2 score 0.6617789754614027

The Multi-Task BERT (MT-BERT) model's performance on the semantic textual similarity (STS) task is evaluated using various metrics, including correlation coefficient (0.8351), mean squared error (0.7363), root mean squared error (0.8581), mean squared logarithmic error (0.0865), median absolute error (0.5242), mean absolute error (0.6561), explained variance score (0.6890), and R2 score (0.6618). These metrics collectively suggest that the model effectively captures the semantic similarity between sentence pairs, with a strong positive linear relationship between predicted and actual scores, reasonable prediction accuracy, and good performance in capturing the logarithmic relationship between scores. The model also explains a significant amount of variability in the data and provides a good fit, as indicated by the explained variance and R2 scores.

Overall, the MT-BERT model demonstrates good performance on the STS task, with strong correlation and reasonable error metrics. However, the confusion matrices for sentiment analysis and paraphrase detection tasks highlight areas for improvement. The model's bias towards positive sentiment and its misclassification of some true positive paraphrase instances suggest the need for further refinement and potentially the incorporation of techniques to handle class imbalances. Despite these challenges, the model's overall performance is promising, and with appropriate enhancements, it has the potential to achieve even better results on these tasks.

6 Conclusion

In this project, we developed a multi-task learning approach using BERT to simultaneously tackle sentiment analysis, paraphrase detection, and semantic textual similarity tasks. By leveraging a shared BERT backbone and incorporating task-specific architectures, such as Siamese networks and cosine similarity, our model demonstrated promising performance across all three tasks.

Experimental results on the test set showed an overall score of 0.738, with task-specific scores of 0.499, 0.804, and 0.822 for sentiment analysis, paraphrase detection, and semantic textual similarity, respectively. While the model slightly underperformed on sentiment analysis compared to the baseline, it achieved significant improvements in paraphrase detection and semantic textual similarity. The model's consistent performance between the development and test sets indicates good generalization to unseen data.

Analysis of the confusion matrices for sentiment analysis and paraphrase detection revealed some areas for improvement. The sentiment analysis model exhibited a bias towards positive sentiment, misclassifying a significant number of negative instances. Similarly, the paraphrase detection model struggled with some true positive instances. These findings suggest the need for further refinement and potential incorporation of techniques to handle class imbalances.

The semantic textual similarity task, evaluated using various regression metrics, demonstrated the model's effectiveness in capturing sentence-level semantic information. Strong correlation and reasonable error metrics highlight the model's ability to predict similarity scores accurately.

Throughout this project, we learned valuable lessons about the challenges and opportunities of multi-task learning with BERT. While sharing representations across tasks can lead to improved performance and efficiency, it also introduces complexities in balancing task-specific requirements. Fine-tuning the model architecture and hyperparameters for individual tasks while maintaining

overall multi-task performance is a delicate process that requires careful experimentation and analysis. Despite the promising results, our work has some limitations. The model's performance on sentiment analysis suggests room for improvement, and the confusion matrices indicate potential biases that need to be addressed. Future work could explore techniques for bias mitigation, such as data balancing and adversarial training. Additionally, investigating alternative architectures and fine-tuning strategies specific to each task could further enhance the model's performance.

In conclusion, our multi-task BERT model demonstrates the potential of leveraging shared representations for simultaneous learning of sentiment analysis, paraphrase detection, and semantic textual similarity. With further refinements and advancements, this approach could lead to more efficient and effective natural language understanding systems capable of handling diverse tasks in real-world applications.

7 Ethics Statement

Developing machine learning models, particularly large language models (LLMs) like BERT, presents several ethical challenges that must be addressed to ensure responsible and sustainable AI development. We'd like to highlight some ethical considerations specific to our project.

Environmetal Impact Training large language models like BERT from scratch is time-consuming and resource-intensive, and has significant environmental costs. High computational power required for extensive pretraining can lead to high energy consumption and carbon emissions, which contribute to environmental degradation. There is a trade-off between developing complex models with higher accuracy (i.e., size of training data, size of the model) and the associated energy consumption.

To address the environmental impact, we need to optimize and balance model complexity and performance with sustainability, which can include:

- Investigating more efficient pretraining methods to reduce computational resource requirements, such as using smaller and well-curated datasets, and transfer learning techniques that leverage pre-trained models with minimal additional training.
- Implementing sustainable computing practices, like using energy-efficient hardware and optimizing algorithms for reduced energy consumption. Working with cloud service providers that offer green energy options can also help mitigate individuals' environmental footprint.
- When I was doing the project, utilizing Jupyter Notebooks for development and experimentation really helped to prevent repeated training runs. By maintaining a comprehensive and reproducible record of experiments, Jupyter Notebooks can help avoid unnecessary retraining and hence reduce computational waste.
- Providing transparency about the energy consumption and carbon footprint of different model training processes. Different LLMs should share the environmental impact to raise awareness and promote more sustainable practices within and beyond the NLP community.

Bias and Fairness Another significant ethical concern is the potential for the model to learn and propagate biases inherent in the training datasets. Our sentiment analysis uses the SST dataset, paraphrase detection utilizes the Quora dataset, and semantic textual similarity relies on the SemEval dataset. These datasets, as well as other datasets online, may contain biased language and stereotypes, reflecting societal prejudices related to race, gender, or socioeconomic status. For instance, biased sentiment data can lead to unfair content moderation, while biased paraphrase detection might misinterpret expressions from different cultural backgrounds.

To mitigate the potential biases, we could implement the following strategies:

- Regularly auditing datasets to identify and mitigate biases, such as adding diverse examples that represent various demographic groups to balance the dataset, to ensure diverse representation within the data by sourcing additional data points that reflect underrepresented groups.
- Applying bias mitigation algorithms and techniques such as such as re-weighting, re-sampling, and adversarial training to reduce the impact of biases during model training. For example, using algorithms that specifically focus on fairness constraints to balance model predictions across different groups.

- Using fairness-specific evaluation metrics to monitor model performance across various demographics, ensuring that the model's outputs are equitable. For example, calculating and comparing the false positive and false negative rates of sentiment analysis predictions across different demographic groups can reveal if the model is biased against certain populations. If the model shows a higher false positive rate for a specific group, targeted adjustments can be made to address this bias.

References

[1] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[3] Rich Caruana. Multitask learning. *Machine Learning*, 28, 07 1997.

[4] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization, 2019.

[5] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.

[6] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding, 2019.

[7] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[8] Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[9] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification?, 2020.

[10] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2019.

[11] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning, 2023.

A Appendix (optional)

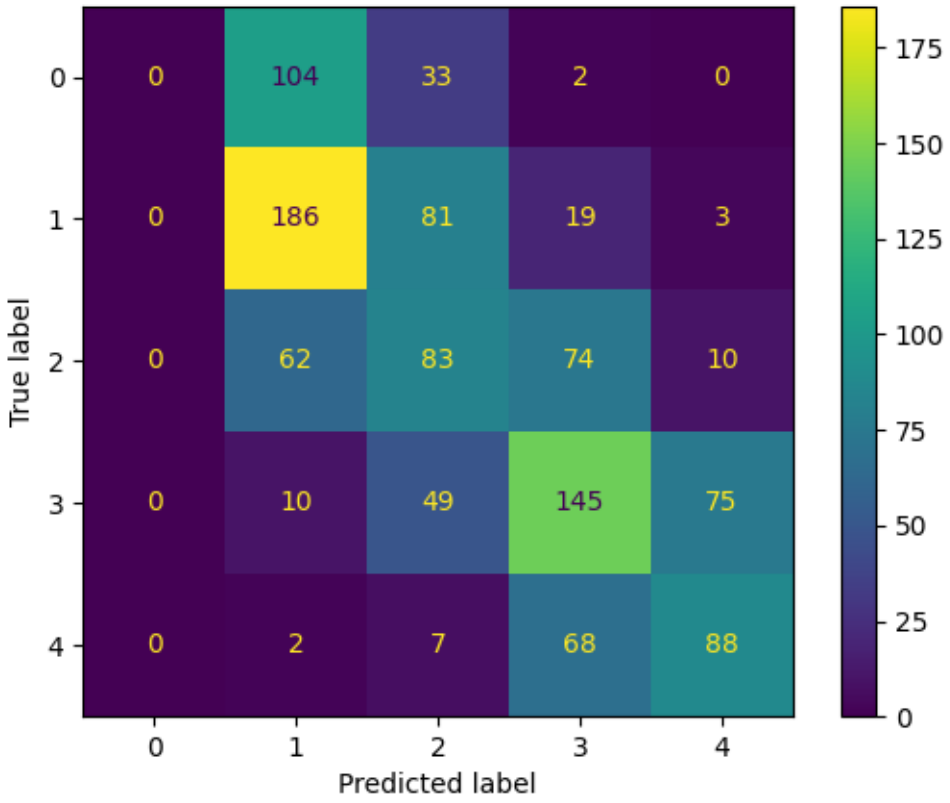

Figure 1: confusion matrix for the sentiment analysis task

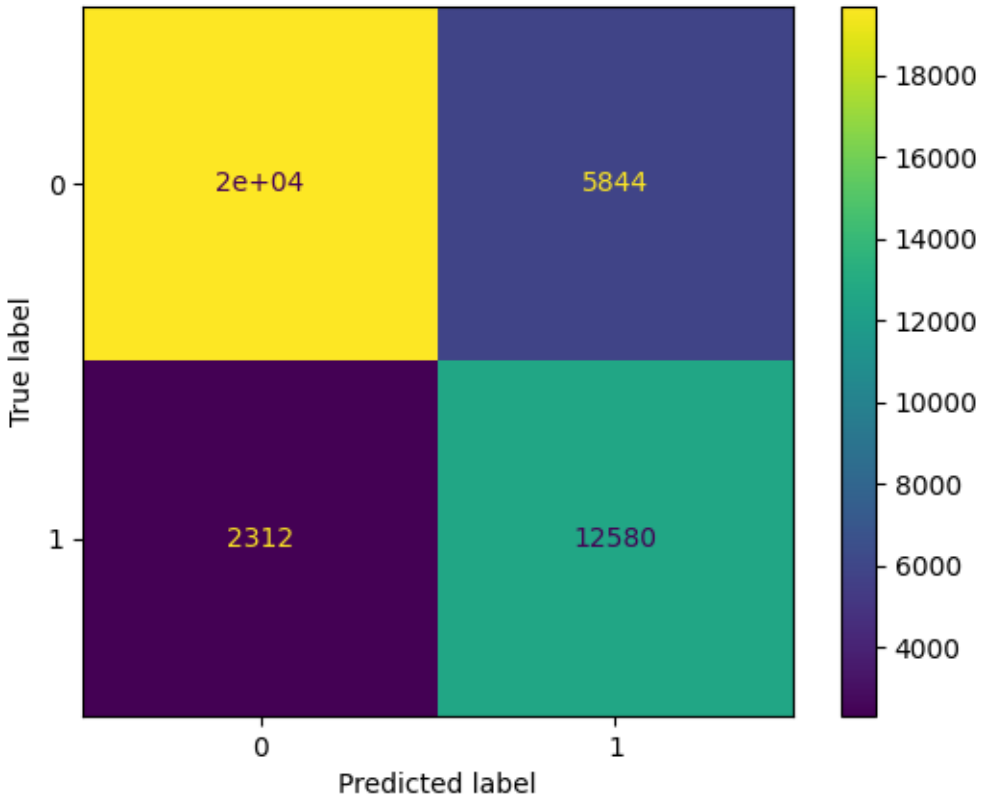

Figure 2: confusion matrix for the paraphrase