Enhancing minBERT for Multi-Task NLP: Architectural and Training Innovations

Stanford CS224N Default Project

Xinxie Wu
Department of Computer Science
Stanford University
xinxiewu@stanford.edu

Abstract

This project introduces minBERT, a tailored version of the 'bert-base-uncased' model, optimized for multi-task learning across three pivotal NLP tasks: Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity. By refining the architecture and employing sophisticated pre-training techniques, we enhance the model's performance on these tasks. Specifically, we explore various architectural adjustments such as different headers for each task and advanced pooling methods. We also implement a strategic pre-training regimen that includes within-task, cross-domain, and staged pre-training approaches to optimize task-specific performance. Our experimental results demonstrate significant improvements over baseline models, achieving 53.8% accuracy in Sentiment Analysis, 89.2% in Paraphrase Detection, and a 0.774 Pearson correlation in Semantic Textual Similarity. These outcomes underscore the potential of targeted optimizations in transforming the capabilities of standard language models for specialized tasks.

1 Key Information to include

• Mentor: Aditya Agrawal • No External Collaborators • Not Share Project Across Classes

2 Introduction

In the realm of Natural Language Processing (NLP), the development of models that can comprehend and interpret human language with sophisticated understanding remains a pivotal challenge. The advent of transformer [1]-based BERT (Bidirectional Encoder Representations from Transformers) [2], has significantly advanced the state-of-the-art, offering a pre-trained model capable of capturing the nuanced contextual relationships within text.

This project explores the implementation and fine-tuning of minBERT across three critical NLP tasks: Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity (STS). Our initiative, dubbed minBERT, refines the 'bert-base-uncased' model through targeted architectural modifications and optimized further pre-training strategies. These enhancements are designed to improve the model's efficiency and effectiveness across the designated tasks, achieving test accuracy of 53.8% in Sentiment Analysis, 89.2% in Paraphrase Detection, and 0.774 Pearson correlation in STS.

3 Related Work

The foundational architecture of our work, BERT, has revolutionized NLP through its deep contextualized training, as established by Devlin et al. (2018) [2]. It has set benchmarks across various NLP applications, laying the groundwork for task-specific adaptations. Pertinent to our enhancements are studies like those by Sun et al. (2019) [3], which underscore the necessity of task-adaptive training

and fine-tuning methodologies to amplify BERT's performance specifically. These modifications are crucial for multi-task applications as they allow for nuanced training that aligns closely with task-specific requirements. Additionally, the works of Reimers and Gurevych (2019) [4] on siamese BERT-networks have informed our approach to semantic similarity, optimizing the way our model processes and understands sentence relationships. Moreover, Loshchilov and Hutter's (2019) [5] insights into decoupled weight decay regularization have been instrumental in refining our training regime to prevent overfitting while maintaining training efficiency. Our research builds on these critical insights, pushing the boundaries of task-specific performance while maintaining the model's adaptability for multi-task learning, aiming to mitigate the noted limitations of prior models and extend their applicability to more specialized or restricted-resource environments.

4 Approach

To enhance the minBERT model for multitask learning, our methodology was bifurcated into two primary strategies: intricate model architecture modifications [4] and advanced pre-training protocols.

Model Architecture [6]. The base for our experiments was the 'bert-base-uncased' model, which we adapted to each specific task by integrating distinct headers designed to process and interpret the representations provided by minBERT effectively (Figure 1).

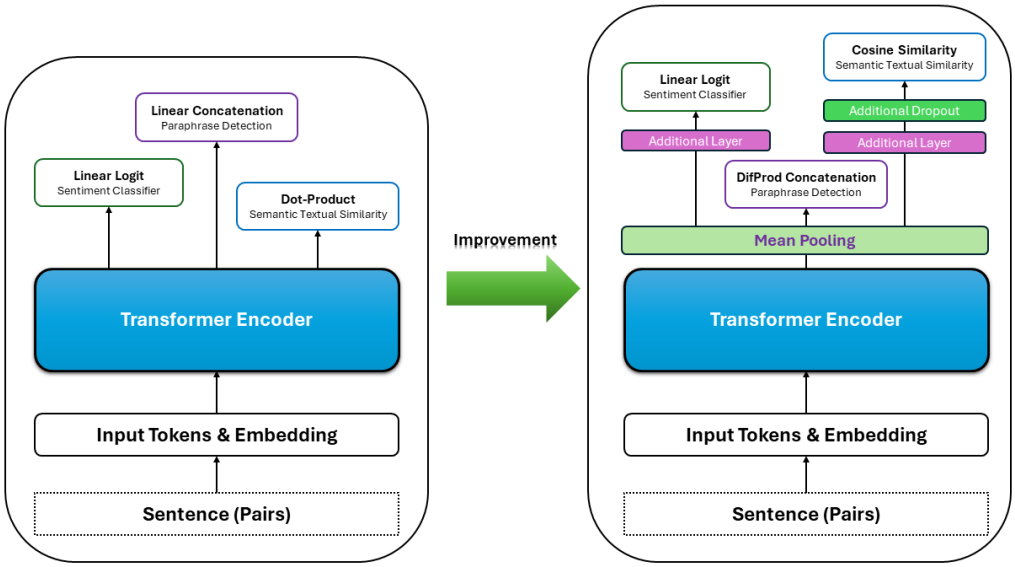

Figure 1: Model architecture: baseline (left) and improvement (right)

1. **Sentiment Classification**: This task necessitated a quinary classification layer where the output header comprises a single linear layer for generating logits, subsequently processed by a softmax function to derive the final class probabilities. The softmax function is articulated:

$$\textit{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{1}$$

2. **Paraphrase Detection**: Multiple architectures were scrutinized for this binary classifier:
   (a) Similar to the sentiment classifier, a simple linear layer directly producing logits.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

   (b) Cosine similarity for measuring the direct similarity between sentence embeddings.

$$\textit{Cosine\_Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{3}$$

(c) An amalgamation of absolute difference and element-wise product of embeddings, which generated the best results by capturing both contrastive and corroborative features between sentences, expressed by:

$$\mathbf{D} = |\mathbf{A} - \mathbf{B}| \tag{4}$$

$$\mathbf{P} = \mathbf{A} \odot \mathbf{B} \tag{5}$$

3. **Semantic Textual Similarity (STS)**: This regression task aims to quantify sentence pair similarity on a scale from 0 to 5, based on their semantic similarity. Cosine similarity was deemed most efficacious, aligning with contemporary research that underscores its utility in capturing semantic nuances across diverse contexts.

In addition to header configurations, embedding processing techniques were also explored:

1. Direct utilization of minBERT's output embeddings.
2. Implementation of mean and max pooling methods, with mean pooling providing superior performance by offering a more uniform and comprehensive representation.

$$p_{\text{mean}} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6}$$

$$p_{\text{max}}[j] = \max(x_1[j], x_2[j], \ldots, x_n[j]) \tag{7}$$

Further Pre-training [3]. Fine-tuning regimen incorporated multi-staged and domain-specific strategies:

1. **Within-task**: Concentrating on enhancing the model's acuity to each task's unique nuances.
2. **Cross-domain**: Boosting the model's generalization capabilities across related tasks.
3. **In-domain with staged training**: Sequential training starting with tasks that share more in common, progressively moving to more distinct tasks. This approach proved particularly beneficial, aligning the model's internal representations with the semantic requirements of each task in a graduated manner (Figure 2).

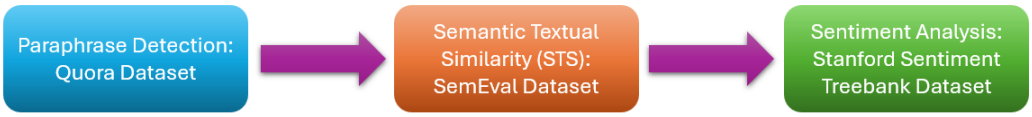

Figure 2: Three-staged in-domain training

Regularization Techniques. To curb overfitting and foster robust generalization, we implemented:

1. **Decoupled weight decay regularization [5]**: This method helps in refining the model without conflating the effects of weight decay with other aspects of the optimization process.
2. **Bregman projections in AdamW [7] optimizer**: This novel approach that ensures the optimizer's steps are both effective in reducing loss and stable enough to maintain performance across varied epochs and batches.

$$\text{AdamW: } \theta_{t+1} = \theta_t - \eta\left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda\theta_t\right) \tag{8}$$

$$\text{Adj. AdamW w/ Bregman: } \theta'_{t+1} = \arg\min_{u \in C} \frac{1}{2}\|u - (\theta_t - \eta \cdot (\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda\theta_t))\|^2 \tag{9}$$

Our approach, by marrying sophisticated architectural modifications with cutting-edge training methodologies, aims to optimize task-specific efficacy while preserving the flexibility requisite.

5 Experiments

5.1 Data

Since we are doing the Default Final Project (DFP), our experiments utilized the provided Stanford Sentiment Treebank (SST) [8] dataset for Sentiment Analysis, Quora dataset for Paraphrase Detection [9], and SemEval dataset for Semantic Textual Similarity (STS) [10]. No additional dataset was involved for our project, and the basic data distribution as below.

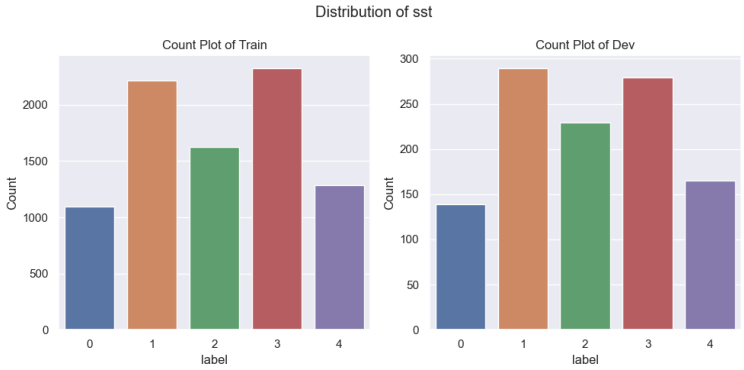

Figure 3: Data Distribution of Stanford Sentiment Treebank (SST)

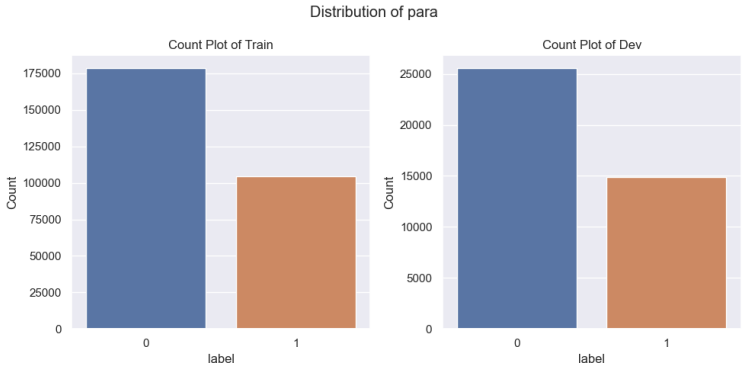

Figure 4: Data Distribution of Quora

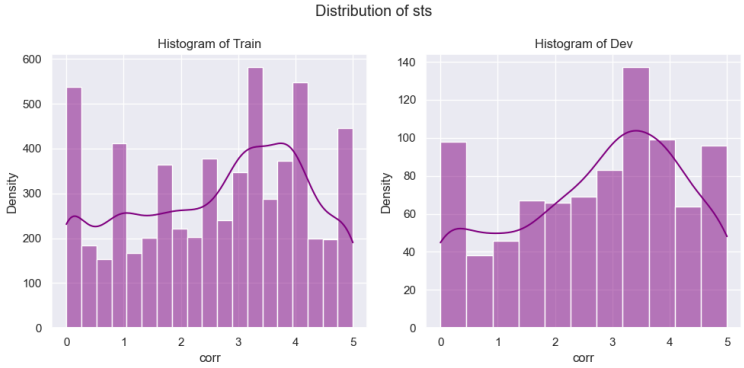

Figure 5: Data Distribution of SemEval (STS)

5.2 Evaluation method

Evaluation method is pre-determined in DFP, and so our project uses classification accuracy for Sentiment Analysis, binary prediction accuracy for Paragraph Detection, and Pearson correlation for Semantic Textual Similarity.

5.3 Experimental details

Our experiments were conducted using the 'bert-base-uncased' model as the foundational architecture. Learning rate is set up $1e-05$ initially and adjusting dynamically during the training. Batch size is set to 8 and the number of epoch is 15. Also, we applied Bregman projections ($\beta_{proximal} = 0.01$) within the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and utilized weight decay (0.1) as part of our regularization strategy.

5.4 Results

The model architecture combining linear layers (Sentiment Classification) with difference-product (Paraphrase Detection) and cosine similarity (Semantic Textual Similarity) operations, enhanced by mean pooling, showed significant improvements across all tasks when compared to the baseline. Additionally, this architecture was further enhanced through staged pre-training and rigorous regularization techniques. The results are summarized in Table 1, which details the performance across various configurations and training stages.

1. **Baseline Performance**: As discussed, we use linear combo and dot-product for the baseline model structure, directly using minBERT's output, without any pre-training. In Table 1, we can see that the sentiment classifier reaches the accuracy of only 14.4%. This low baseline accuracy indicates significant challenges in handling sentiment classification with the initial model setup, potentially due to inadequate feature extraction or poor generalization from minBERT's pre-training data. Paraphrase detection's baseline accuracy is 36.9%, this moderate number suggests some capability to understand textual similarities but lacking precision and robust differentiation between nuanced text pairs. STS shows the extremely low correlation for the baseline, 0.075, indicating almost negligible effectiveness in capturing and scoring semantic relationships accurately.
2. **Impact of Model Architecture Enhancements**: Applied mean pooling without further pre-training, we updated each header. For sentiment header, we found that the additional layer helps to improve the accuracy to 24.3%, almost 10% higher than the baseline, showcasing the effectiveness of the newly integrated linear layers in enhancing feature discrimination. Paraphrase header has been updated to use difference-product algorithm, pushing the accuracy to over 60%. This improvement enhanced the model's ability to capture essential features indicative of paraphrases. With the addition of cosine similarity in the architecture, the STS layer returned the correlation of 0.533, which demonstrated a substantial increase in correlation, reflecting improved alignment with human judgment on semantic similarities.
3. **Benefits of Further Pre-training w/ Regularization**: Using 3-stage in-domain pre-training, with appropriate regularization techniques of Bregman and weight-decay learning rate, in the test dataset, we achieved 53.8% accuracy for sentiment classification, 89.2% accuracy for paraphrase detection, and 0.774 Pearson correlation for semantic textual similarity. All illustrated the profound impact of targeted pre-training in enhancing the model's ability.

Model Architecture	Pre-train	Baseline	SST Accuracy	Para Accuracy	STS Correlation
Linear+Linear+Dot-Product, bert	N.A.	Y	0.144	0.369	0.075
Linear+DifProd+CosSim, mp	N.A.	dev	0.243	0.608	0.533
Linear+DifProd+CosSim, mp	Para Only	dev	0.202	0.899	0.745
Linear+DifProd+CosSim, mp	3-stage	dev	0.530	0.894	0.606
Linear+DifProd+CosSim, mp	3-stage, reg	dev	0.548	0.894	0.776
Linear+DifProd+CosSim, mp	3-stage, reg	test	0.538	0.892	0.774

Table 1: Model performance

6 Analysis

We created confusion matrix for Sentiment Analysis / Paraphrase Detection, and scatter plot for Semantic Textual Similarity, baseline result in Figure 6, architecture result in Figure 7, and the final result in Figure 8.

In **Sentiment Analysis**, the confusion matrix from the baseline model shows a strong bias towards predicting extreme sentiments, either highly positive or highly negative, suggesting that the model was overly sensitive to more expressive language and often ignored subtler cues of sentiment. In contrast, the final results display a more balanced prediction across all sentiment classes. Notably, the correct predictions for neutral sentiments (class 2) and somewhat positive sentiments (class 3) have improved significantly. This indicates that the enhancements in model architecture and training methodology helped mitigate the initial bias towards extreme sentiments.

For **Paraphrase Detection**, the baseline model frequently predicted non-paraphrases over paraphrases, possibly indicating an initial conservative classification stance, where clearer signals of similarity were required to classify pairs as paraphrases. The final confusion matrix reveals a more balanced approach with improved detection of paraphrases, suggesting that our adjustments, particularly the use of absolute differences and product of embeddings, provided a more nuanced understanding of semantic similarities and differences, enhancing the model's ability to correctly classify paraphrases.

In the task of **Semantic Textual Similarity**, the baseline scatter plot shows a clustered distribution of similarity scores around the lower scores, indicating a poor correlation between predicted and actual similarity scores. The final scatter plot, however, exhibits a broader and more upward-trending distribution of data points across the similarity scale. This change implies a significant improvement in the model's ability to assess and score textual similarity across a wider range of values, reflecting a more accurate understanding of nuanced semantic relationships.

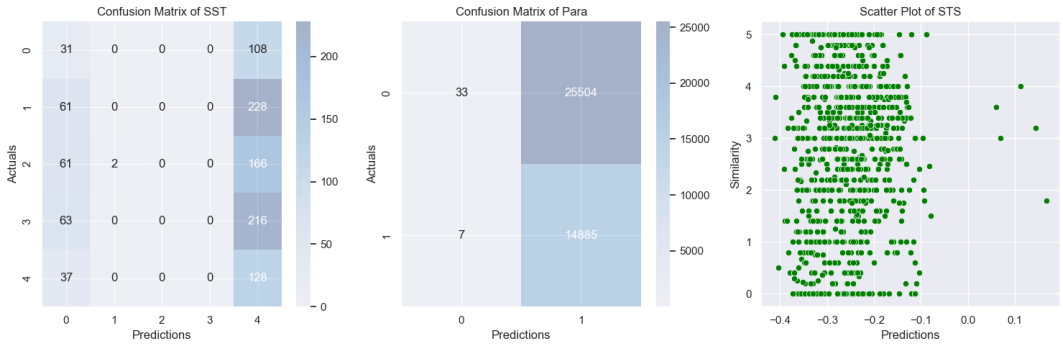

Figure 6: Baseline confusion matrix and scatter plot

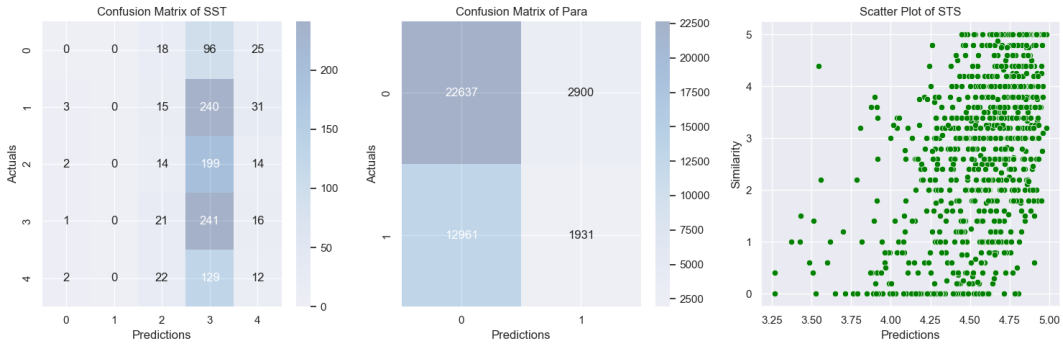

Figure 7: Architecture-only confusion matrix and scatter plot

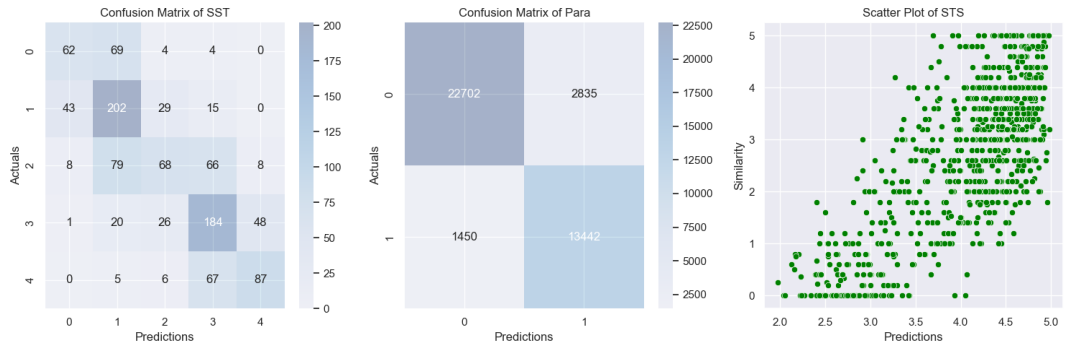

Figure 8: Final confusion matrix and scatter plot

7 Conclusion

Our project successfully demonstrated the effectiveness of a multitask learning approach using a minimally modified BERT architecture to address three distinct NLP tasks: Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity. By leveraging the 'bert-base-uncased' model, fine-tuned with task-specific adjustments in model architecture and training processes, our system achieved noteworthy improvements over baseline models on all evaluated tasks.

We learned that refining the model's architecture by integrating task-specific headers and employing different embedding pooling strategies (mean and max pooling) critically improved the model's ability to discern and predict nuanced linguistic features. The adoption of mean pooling, in particular, proved effective in capturing sentence-level semantic nuances, which was especially beneficial for the STS task.

Our achievements are underscored by the substantial improvements in model performance. For Sentiment Analysis, adjustments to the model's sensitivity to varying sentiment intensities led to more balanced accuracy across sentiment classes. In Paraphrase Detection, the introduction of composite embeddings (absolute differences and products) enabled the model to better gauge semantic equivalence. For the Semantic Textual Similarity task, enhancements in embedding processing facilitated a finer-grained similarity assessment, as evidenced by the expanded and upward-trending scatter of similarity scores.

Despite these successes, the project has limitations. The reliance on a single pre-trained model base may restrict the generalizability of our findings across other BERT-like architectures. Additionally, while improvements were significant, the absolute performance levels, particularly in finer-grained sentiment detection, suggest there remains room for optimization.

Future work could explore the integration of alternative contextual embedding techniques and the application of more sophisticated neural network layers specific to each task to further refine the understanding and processing of task-specific features. Investigating the effects of training with larger, more diverse datasets or implementing advanced regularization techniques might also yield further improvements. Moreover, expanding this approach to multilingual datasets could enhance the model's applicability and robustness in global NLP applications.

8 Ethics Statement

The deployment of NLP models like ours, which are designed to perform tasks such as Sentiment Analysis, Paraphrase Detection, and Semantic Textual Similarity, raises specific ethical challenges and societal risks.

1. **Bias Propagation, in Sentiment Analysis**:
   (a) *Problem*: Our sentiment analysis model has demonstrated a tendency to propagate bias present in the training data, where categories 1 and 3 are over-represented (Figure 3). In our final model result, this has led to higher prediction accuracies in these

categories—69.9% for category 1 and 65.95% for category 3—compared to significantly lower accuracies in categories 0 and 5, and less than 30% for category 2 (calculated based on the confusion matrix in Figure 8). Such disparities can skew the model's application in real-world scenarios, disproportionately favoring certain sentiments and potentially leading to unfair outcomes

(b) *Mitigation*: Enriching the training dataset with more diverse examples across all categories would be a good way. Or, we can apply data augmentation methods to balance class representation, such as over-sampling the classes 0, 2 and 5.

2. **Misuse, of Paraphrase and Semantic Textual Similarity Tools**:

(a) *Problem*: The paraphrase detection capabilities of our model, which achieve around 90% accuracy, pose a risk of facilitating academic dishonesty. Users could potentially exploit the model's error margin to manipulate texts just enough to evade plagiarism detection systems, thereby compromising academic integrity. Similarly, our semantic textual similarity model tends to overestimate the similarity scores, inaccurately predicting scores as high as 2 to 3.5 for pairs that should have an actual similarity of 0. Such inaccuracies can be exploited to craft misleading content that appears closely aligned with credible sources, thereby facilitating the spread of misinformation.

(b) *Mitigation*: To mitigate these kinds of issues, digital watermarking is a good choice because it provides a means to trace the origin of altered content. Making users aware of the consequences of misuse, both from ethical and legal perspectives, should also help to mitigate this problem, so related education is needed.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6000–6010. Curran Associates Inc., 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019.

[3] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer, 2019.

[4] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992. Association for Computational Linguistics, 2019.

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[6] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Efficient natural language response suggestion for smart reply. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6896–6910. Association for Computational Linguistics, 2020.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Association for Computational Linguistics, 2013.

[9] Somnath Fernando and Mark Stevenson. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics (CLUK 2008)*, pages 45–52. Citeseer, 2008.

[10] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. Semeval-2013 task 12: Multilingual semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43. Association for Computational Linguistics, 2013.