Orthogonal Projection Loss for Multi-Headed Attention

Stanford CS224N Default Project

Kyle McGrath
Department of Computer Science
Stanford University
mcgrathk@stanford.edu

Abstract

Orthogonality has been a desired trait in convolutional neural models and in linear transformations in transformer models due to the norm preserving nature of the computation. In this report, we demonstrate a novel finding that transformer embeddings and key weight matrices tend towards a more orthogonal representation without any influence. Following this empirical finding, we apply orthogonality constraints to weight matrices in self-attention to further assist this bias towards orthogonality. Our results reveal that incorporating these constraints not only enhances the model's representational efficiency but also leads to improvements in accuracy across various tasks when compared to the baseline models.

1 Key Information to include

Mentor: Sonia Chu • No External Collaborators • Project is not shared among classes.

2 Introduction

The representation space of modern transformer models is highly influenced by the learned embedding vectors in the network, and the associated positional encoding which allows the neural network to understand the word ordering of sentences and correlate distant concepts. It has been shown that modern large language models fail to perform with highly positional information, such as numerical encoding and research centered around this topic targets positional encoding as the primary mechanism in which to improve this accuracy Su et al. (2023) Zhou et al. (2024). Since positional encoding and embedding representation space are highly dependent, it is possible that the method in which embedding vectors are learned will also influence the length generalization and long distance dependency representations in the model.

Orthogonality constraints were introduced by Brock et al. (2017), for convolutional filters in image processing, as it allows for the norm of the original data to be unchanged by the filter. Later, Zhang et al. (2021a) apply this to transformer based networks an empirically prove that dissimilarities introduced by orthogonality constraints between projected vectors in affinity matrix (where the affinity matrix $\mathbf{A}$ is described by the product of the query and key matrices in self-attention) of the self attention module, the linear transformation in self-attention, and the linear transformations in position wise feed-forward networks increase the performance of transformer models Zhang et al. (2021a).

The goal the CS224N class project is to improve the performance of a pre-trained BERT model (which stands for Bidirectional Encoder Representations from Transformers) model by fine-tuning the model to improve the embeddings and representation of vectors on three downstream tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. Motivated by the work Brock et al. and Zhang et al., this paper attempts to further localize and understand which modules of

the network are well influenced by orthogonality constraints. While Zhang et al. (2021a) empirically demonstrate improvements in accuracy, the specific modules in which they apply orthogonality constraints are predicated by their empirical performance. Inspired by their work, this paper attempts to further localize where applying these constraints is beneficial, and demonstrate that applying these constraints to a pre-trained model with no special initialization can boost performance.

3 Related Work

Joint Effects of Positional Encoding and Learned Embeddings
Originally, Vaswani et al. (2023) utilized learned embeddings and an absolute positional encoding scheme to allow for the network to jointly understand word meaning and word positioning in a sentence. Later on, it was found that absolute positional encoding is less favorable, as the importance in deciphering word order came from understanding the relative positioning of each word Su et al. (2023). Since then, research focused on improving the positional and relational word relation understanding of neural networks has primarily focused on just improving the positional encoder module Zhou et al. (2024).

Wang and Chen (2020) attempt to understand the conceptual interaction between positional encoding and embeddings to see if embeddings themselves affect how position dependent information is processed. This study empirically probes the learned position embedding matrices on mainstream pre-trained models to understand if position embeddings accuratly learn positional information and how different positional embeddings affect NLP tasks Wang and Chen (2020). In the study, Wang and Chen (2020) find that encoder only models, especially BERT, do not learn absolute positional information, and instead only weakly learn relative positional information.

Haviv et al. (2022) demonstrated that transformer language models can learn positional information even if the positional encoding module is removed. This indicates that positional encoding and word embeddings are not mutually exclusive; they jointly represent both word meaning and positional information, sharing responsibility for these tasks. From this, our project explores the theory that word meaning and position information are jointly coupled in the embedded hidden state at the beginning of the network and possibly in further latent spaces in deeper layers.

Orthogonality
Building upon the above research, we aim to create a more optimal embedding space that can enhance the learned positional encoding layer in BERT. One potential approach is by introducing orthogonality into the system. Orthogonality is considered a desirable trait in certain machine learning systems. Brock et al. (2017) introduced orthogonality constraints in a neural photo editing system to ensure the norm of the original source image data remains unchanged after multiplication by an orthogonal convolutional filter. They also highlighted the success of initializing weights with orthogonal matrices.

Years later, Zhang et al. (2021a) noted the lack of research on applying these methods to transformer models. Their rationale for applying orthogonality constraints is that it enhances numerical stability by upper bounding the Lipschitz constant of linear transformations, which measures the rate of change or variation in representations (Zhang et al. (2021a). Consequently, Zhang et al. (2021a) theorized that applying orthogonality constraints to a transformer model, specifically at the points in a transformer model where linear transformations occur, could lead to more robust representations that are less sensitive to data perturbations. They empirically prove this by applying these constraints to the linear transformed affinity matrix and linear transformation layers, and these results lead to significant performance improvements on benchmarks.

This project seeks to demonstrate that orthogonality constraints on an unconstrained pretrained model can be beneficial. Furthermore, it aims to identify which modules within the transformer network benefit most from these constraints. The exploration includes examining orthogonality in self-attention weight matrices, with the hypothesis that these learned matrices also have a role in keeping positional and contextual information bounded. This approach differs from Zhang et al. (2021a)'s mathematical theory-based application, as it empirically investigates the optimal placement for orthogonality constraints, challenging the assumption that affinity matrices are the best candidates for such constraints.

4 Approach

The approach to this project is three-fold. Firstly, the minimum goal of this project at is to expand upon a baseline pretrained BERT model to achieve sufficient generalization on the three downstream tasks as provided by the CS224N project description Manning (2024). Secondly, the project attempts to empirically understand if the network tends towards an orthogonal form in any key linear projections in self-attention or embedding. Lastly, multiple methods are attempted that attempt to improve upon baseline performance and demonstrate that orthogonalization constraints can be applied to pretrained models, not just specially initialized models.

4.1 Implementing minBERT and Optimizing Downstream Tasks

First I implement the minBERT model using the specififfications from the project handout Manning (2024). I utilized the following approaches to create each task specific head and calculate loss for each task.

Sentiment analysis (SST) The sentiment analysis tasks requires that we predict the sentiment of a sentence and categorize the sentiment into one of five categories. I created a task head with a single linear layer with an input the size of the hidden dimension, and a singular output dimension of 5. To calculate the loss of this, I applied cross entropy logits on the `[CLS]` token output of the network, which is a special token generated by BERT which can be used for classification tasks Devlin et al. (2019).

Paraphrase Detection (Para) The paraphrase detection tasks requires that the network determine if two input sentences are a paraphrase of each other. Two approaches can be taken to this, the first is to run the network twice, then concatenate the output `[CLS]` token together and run a classifier layer on top of this. In my experimentation this proved to lead to much lower results than expected. Another method is to first concatenate the two sentences together by inserting the `[SEP]` token in between then, then running this single sentence through BERT and classifying the `[CLS]` token from this. This method worked much better in my experimentation. Lastly, I use a single linear layer for the task head with an input dimension equal to the hidden size, and a single output node. I calculated the loss utilizing binary cross entropy with logits.

Semantic Textual Similarity (STS) Lastly, for the STS task I utilized a similar approach to paraphrase detection by concatenating the two sentences together with a `[SEP]` token, and utilizing a linear layer with a single output node. From there, I calculated the cosine embedding loss on the output node to allow for the network to predict the semantic textual similarity score from a linear scale of 0-5.

I iterated through each dataset evenly, where in a single iteration I calculated the loss for each piece of data and then backpropagated the sum of each task head specific loss. This method forms my baseline training approach.

4.2 Application of Various Orthogonality Constraints

Previous research demonstrates that orthogonality can improve numerical stability and preserve the norm of data being projected by the orthogonalized matrix. Further, I was interested in testing if orthogonality introduced by learned embeddings helps assist the positional encoding module or subsequent hidden states. This theory is based on the notion that confining the tasks of learned embeddings and positional encoding to be mutually exclusive may enable each module to learn a richer representation space.

Zhang et al. (2021b) demonstrate that applying orthogonality constraints to the affinity matrix and the linear transformation layers prove beneficial. In this study, I test whether or not applying these same constraints to the query, key and value matrices in self-attention prove beneficial, and further, whether or not applying this orthogonality constraint directly to the embedding matrix helps improve the embedding space representation.

4.2.1 Orthogonalization using QR Decomposition methods

For an exact orthogonal solution, QR decomposition methods can be used to decompose any matrix into an orthogonal matrix $\mathbf{Q}$, and an upper triangular matrix $\mathbf{R}$. Two popular methods for this involve the modified Gram-Schmidt algorithm and decomposition by Householder reflections Gander (1980).

We can create a loss function for both techniques by subtracting the returned $\mathbf{Q}$ matrix from either method from the original input matrix $\mathbf{A}$, and then take the Frobenius norm of either to return a loss, where λ is a loss scaling factor:

$$\mathcal{L}_{QR} = \lambda \|Q - A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |q_{ij} - a_{ij}|^2} \tag{1}$$

Unfortunately, both of these methods did not work in my usecase. Both lead to more than half of the accuracy of my baseline described above, even with minimal weighting $\lambda = 1 \times 10^{-6}$. I believe this may be due to the fact that the specific orthogonalized matrix form of the $\mathbf{Q}$ decomposed matrix was far too different from the original $\mathbf{A}$ matrix, leading to significant error when trying to backpropogate the difference. Further, the computational complexity of either algorithm took too long to compute, even with accelerated libraries.

4.2.2 Orthogonalization using Regularization Methods

Brock et al. (2017) introduce a method in their paper that takes the sum of the Gram matrix $\mathbf{AA^T}$ minus the identity matrix $\mathbf{I}$, which allows for an imperfect, yet simple method of encouraging the input matrix $\mathbf{A}$ to take a more orthogonalized form. Following Ranasinghe et al. (2021), I instead reform the matrix to be applied to be used as a loss function for any input matrix $\mathbf{A}$, by taking the Frobenius norm of the Gram matrix minus the identity matrix:

$$\mathcal{L}_{ortho} = \lambda \|AA^T - I\|_F \tag{2}$$

Where $\| \bullet \|_F$ denotes the Frobenius norm, and λ denotes the loss scaling factor. This approach works better, as the computational complexity is greatly reduced, and still converges an input matrix towards an orthogonalized form.

4.3 Measuring Orthogonality

Orthogonality can be measured from the equation designed by Kenefake (2020). The method calculates the Gram matrix $\mathbf{G} = \mathbf{QQ^T}$ of a given input matrix $\mathbf{Q}$, then setting each of the diagonal elements to zero $\mathbf{G_0}$. In a properly orthogonal matrix, this resulting matrix $\mathbf{G_0}$ would be equivalent to the identity matrix $\mathbf{I}$. Then calculate the maximum error by taking the largest value from this matrix:

$$\text{err\_orth} = \max_{i \neq j} |(\mathbf{QQ}^T)_{ij}| \tag{3}$$

This measure then allows for us to measure the orthogonality of any input matrix, and similarly measure the effectiveness of the orthogonality constraint loss imposed on any matrix.

5 Experiments

5.1 Data

Three datasets are used to fine-tune each downstream task in the BERT model. Each of the downstream tasks are based on sentence-level data. The model utilizes three different task heads to evaluate the data accordingly. The datasets used are listed below:

Stanford Sentiment Treebank
For the sentiment analysis task the model is fine-tuned on the Stanford Sentiment Treebank (SST) Socher et al. (2013), which consists of 11,855 sentences used in movie reviews. The dataset represents 5 classes, where each phrase in a row corresponds to one of the following sentiment classifications: negative, somewhat negative, neutral, somewhat positive, or positive. The network must classify the sentence into one of these 5 tasks, which correspond to the labels 0-4, respectively.

Quora Dataset
The paraphrase detection task is designed to check whether or not two separate sentences are

paraphrases of each other. In this scenario, a paraphrase is defined as a "restatement (or reuse) of text given the meaning in another form" Fernando and Stevenson (2009). Quora is a question answering platform, where multiple users may ask the same question on the board with different wording. Given two sentences, the Quora dataset labels each pair of questions with a 1 or 0, where if the two questions are paraphrases of each other they are labelled with a 1, and if not they are labelled with a 0. The dataset is composed of 408,298 question pairs.

SemEval STS Benchmark Dataset
The semantic textual similarity task utilizes the SemEval Semantic Textual Similarity Benchmark (STS) dataset which consists of 8,628 different sentence pairs that have varying similarity. The goal of the network is to parse the two sentences and determine whether they fit on a scale from unrelated (score of 0) to equivalent meaning (score of 5), and can assume any value in between. Each pair of sentences in the dataset was labelled by a human judge on a continuous scale to mark whether the pair of sentences had any semantic similarities in meaning.

5.2 Evaluation method

For paraphrase detection, model performance is assessed using accuracy, calculated as the ratio of correct predictions to the total number of samples.

In sentiment analysis, accuracy is again employed, where a prediction is deemed correct if the model accurately identifies the sentiment class, otherwise it is considered incorrect.

For semantic textual similarity (STS), we measure model efficacy using the Pearson correlation coefficient, which quantifies the degree of linear correspondence between the predicted scores and the true scores, ranging from 0 to 5. This metric effectively rewards predictions that closely approximate the actual label values.

Lastly, to evaluate matrix orthogonality in our models, we use the orthogonal error measurement, defined in Equation 3, which quantifies the deviation from perfect orthogonality.

5.3 Experimental details

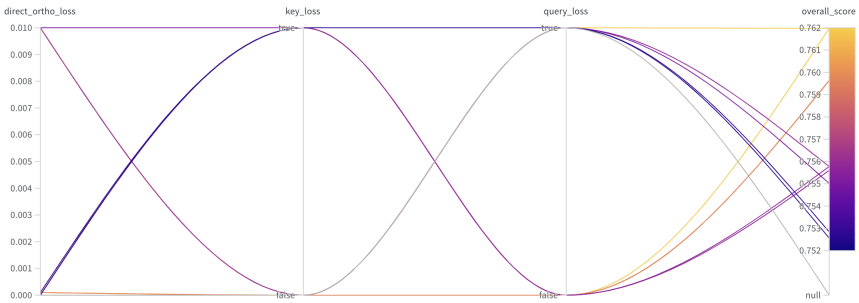

Figure 1: Weights And Biases Sweep Parameters

The experiment was run in two parts. Firstly, a single baseline run is performed which measures the orthogonality utilizing Equation 3 on the embedding matrix, the key, query and value weight matrices, as well as the affinity matrix.

Secondly, orthogonality constraints are applied to the baseline model to determine if applying orthogonality constraints improve the performance of the fine-tuned model. To enable this, a Weights and Biases grid sweep was run on the following parameters:

Orthogonality Loss Scaling Factor λ: $1e^{-2}$, $1e^{-4}$, $1e^{-6}$
Key Orthogonalization: True, False
Query Orthogonalization: True, False

The Weights and Biases sweep can be visualized in Figure 1.

6 Results

6.0.1 Orthogonalization Measurement of Self-Attention Weight Matrices

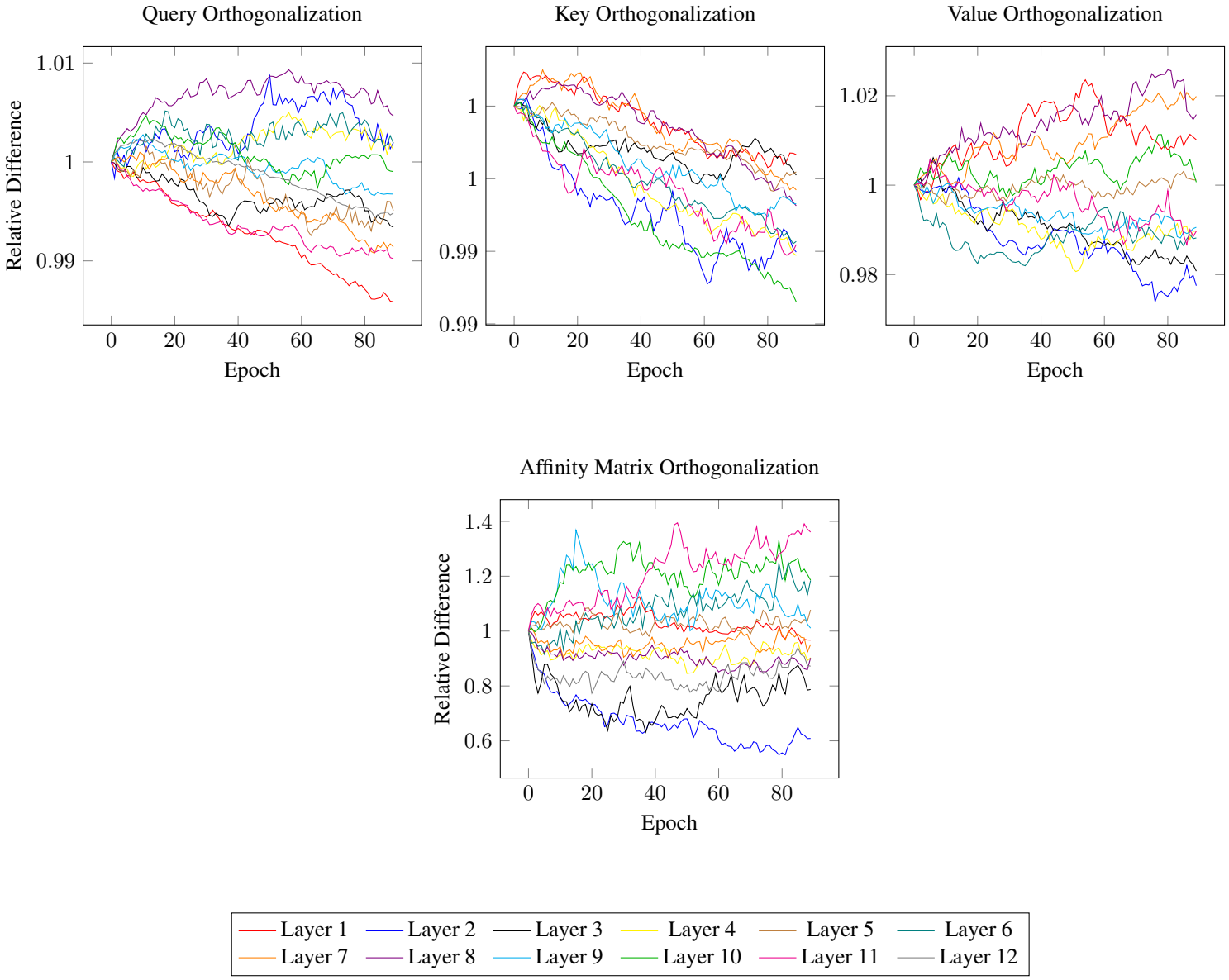

Figure 3: Orthogonalization Results Across Layers

Figure 3 demonstrates the orthogonalization measure of of the query, key and value weight matrices across each layer of the BERT baseline model without any influence from orthogonality constraints. Interestingly, we note that out of the query, key, and value matrices, that the key matrix has a natural tendency to move towards a more orthogonalized form. Further, we note that contrary to Zhang et al. (2021b), the affinity matrix seems to not tend towards an orthogonal form.

The take aways from this investigation are interesting, because to the best of my knoweledge, it has not been observed that the self-attention key weight matrix tends towards orthogonality. I suspect that this effect is caused by the proximity of the key matrix to the softmax layer, which may incentivize the key matrix to preserve the norms of the query matrix.

6.0.2 Application of Orthogonalization Constraints

	SST Task	Para Task	STS Task	Overall Score
Baseline	0.485	**0.876**	0.8358	0.7596
Baseline + Key Orthogonalization	**0.4968**	0.871	0.8362	**0.762**
Baseline + Query Orthogonalization	**0.4968**	0.871	0.8362	**0.762**
Baseline + Key & Query Orthogonalization	0.4687	0.871	**0.8377**	0.7528

Table 1: Ablation study on applying Orthogonality Constraints to Self-Attention

Table 1 shows how applying orthogonalization constraints to the self-attention module affects performance compared to the baseline. We experimented with constraining just the key matrix, just the query matrix, and both matrices together. The results indicate that the best overall scores are achieved by applying orthogonality constraints to either the key or query matrix alone. This was surprising because, based on previous studies, I expected that only constraining the key matrix would yield the best results. Additionally, the combination of key and query orthogonalization yielded better results for the STS task. The baseline model only outperforms the constrained models in the paraphrase detection task. This might be due to the nature of the constraint method, which could remove some of the rich representations necessary for the paraphrase detection task, especially given its larger dataset.

7 Analysis

Observing the results from Figure 3, we measured the orthogonalization of four different matrices in self-attention. Out of the three projection weight matrices present in self-attention, we notice that the key projection weight matrix tends towards orthogonalization in every single layer in the network. Analytically, it may be more favorable for the key projection to preserve the norms of the query projection, hence the tendency towards a more orthogonal form. The data in this figure is normalized against the original value that the matrix returned. From this, we can see that all weight matrices report an orthogonalization error less than 1, which is the original value of the layers error after normalization.

Next, we observe the performance by applying orthogonalization constraints to the key and query projection matrices in self-attention. Following the results from 3, it is expected that constraining the orthogonalization of the key matrix will result in higher performance. Interestingly, this is not the case, as both applying orthogonalization constraints to either the key or query matrix, but not both, results in the highest score at 0.762. We note the high performance of the combination of orthogonalization constraints on both the key and query matrices for the semantic textual similarity tasks at 0.8377, which marks a 0.0015 point increase from other tasks. Lastly, we note that the highest performance on the paraphrase detection task is achieved from the baseline model. This may be due to the orthogonality constraints over constraining the rich representations for this specific task head, since the paraphrase detection task has a much higher amount of data available to it.

8 Conclusion

In this study we perform two analyses. The first is a measurement of various locations in self-attention to measure whether or not transformer models naturally tend towards an orthogonal form. The second is to test whether or not orthogonalization constraints allow for a more rich representation of feature space and allow for a fine-tuned model to perform better on downstream tasks. We report the finding that all key matrix weights in our fine-tuned model tends towards orthogonality, which may be a novel finding. Further we demonstrate that applying orthogonality constraints in the form of a loss function for query and key projection weights allows for higher generalization on certain tasks.

Future work includes measuring if key projection weight matrices tend towards orthogonality in other models, and if this tendency exists during pre-training phases as well. Further, the exact reasons for this tendency are unknown which warrants further investigation.

9 Ethics Statement

When fine-tuning a large pretrained model, certain ethical considerations must be applied. One possible risk is that a fine-tuned model can biases strongly towards the downstream fine-tuned tasks. If the base pretrained model is deemed to be safe, yet the fine-tuning datasets lead to biases in the model that have not been accounted for, the model can be incorrectly biased towards any result in the dataset. In our scenario, we heavily biases the model towards the Quora dataset for the paraphrase detection task, which has not been verified for ethical speech. This dataset could introduce false biases towards a protected class, which would bypass any bias prevention mechanisms in the pretrained model.

Further, accuracy issues can arise in the model that incorrectly bias towards a single tasks when training in a multi task scenario. In our scenario, our data distribution across all downstream tasks was heavily biased towards the number of samples in the Quora dataset. If a fine-tuned model such as this were to be employed in a production environment, we would have to ensure that the accuracy of all tasks performed at an acceptable level. Ethically, if our model were to biases towards the Quora dataset, we may find that our model may overfit to the views of the authors in the dataset, and incorrectly perform on the STS and SST tasks as a result because it was not able to generalize properly.

References

Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. 2017. Neural Photo Editing with Introspective Adversarial Networks. ArXiv:1609.07093 [cs, stat].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Samuel Fernando and Mark Stevenson. 2009. A semantic similarity approach to paraphrase detection. *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*.

Walter Gander. 1980. Algorithms for the qr-decomposition. Research Report 80-021, Seminar für Angewandte Mathematik, Eidgenössische Technische Hochschule, CH-8092 Zürich.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer Language Models without Positional Encodings Still Learn Positional Information. ArXiv:2203.16634 [cs].

Dustin Kenefake. 2020. Orthoginalization - Modified Gram-Schmidt.

Christopher Manning. 2024. Cs 224n (spring 2024) default final project: minbert and downstream tasks. Course Project Description, CS 224N. Accessed: June 1, 2024.

Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat, Salman Khan, and Fahad Shahbaz Khan. 2021. Orthogonal Projection Loss. ArXiv:2103.14021 [cs].

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

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. RoFormer: Enhanced Transformer with Rotary Position Embedding. ArXiv:2104.09864 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. ArXiv:1706.03762 [cs].

Yu-An Wang and Yun-Nung Chen. 2020. What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding. ArXiv:2010.04903 [cs].

Aston Zhang, Alvin Chan, Yi Tay, Jie Fu, Shuohang Wang, Shuai Zhang, Huajie Shao, Shuochao Yao, and Roy Ka-Wei Lee. 2021a. On Orthogonality Constraints for Transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International*

Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 375–382, Online. Association for Computational Linguistics.

Aston Zhang, Alvin Chan, Yi Tay, Jie Fu, Shuohang Wang, Shuai Zhang, Huajie Shao, Shuochao Yao, and Roy Ka-Wei Lee. 2021b. On Orthogonality Constraints for Transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 375–382, Online. Association for Computational Linguistics.

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. 2024. Transformers Can Achieve Length Generalization But Not Robustly. ArXiv:2402.09371 [cs].