Towards Distilled Language Model Interpretability

Chelsea (Zixi) Chen
zixichen@g.harvard.edu

Kevin Huang
kevin_huang@college.harvard.edu

Steve Li
steveli@college.harvard.edu

Abstract

Large language models (LLMs) have recently seen a surge in popularity, but using these models with increasingly large parameter counts poses problems with respect to inference time and hosting scalability. In order to tackle this problem, knowledge distillation (KD) has been shown to be an effective framework for compressing large "teacher" models into smaller "student" models while still retaining many of the performance benefits. However, distillation methods differ in terms of internal architecture, loss function/objective function, and training regime, yielding wildly different distilled models. One thing that is not necessarily captured or measured in existing distillation methods is explanations. In this paper, we are the first to study explanation alignment between a "teacher" model and the distilled "student" models with results showing that certain "student" models are more aligned than others. We additionally propose AlignedDistil, a smaller, general-purpose language representation model that incorporates the attention weights of the teacher model during the knowledge distillation process.

1 Introduction

Large language models (LLMs) have recently seen a surge in popularity with models being able to achieve human-level performance on academic benchmarks such as the LSAT [15]. However, as parameter counts for these LLMs continue to skyrocket, using these models poses problems with respect to inference time and hosting scalability as model parameter counts have continued to skyrocket. For example, RoBERTa [10] contains 123 million parameters, BERT-Large [3] contains 340 million parameters, and GPT-3 [1] dwarfs both with a staggering 175 billion parameters. In order to tackle this problem, knowledge distillation (KD) has been shown to be an effective framework for compressing large models into smaller models while retaining many of the performance benefits. The fundamental intuition is to train a smaller, "student" model to mimic the behavior of a larger and better-performing "teacher" model, thereby learning the latent knowledge contained in the teacher model such that the student model may still perform similarly.

Distillation methods differ in terms of internal architecture, loss function/objective function, and training strategies, resulting in different distilled models. This is despite the fact that the general goal of all KD methods is to compress LLMs to a smaller model while still retaining the latent knowledge from those LLMs. In the literature, student models derived from KD methods have been assessed in terms of their performance. However, there is little work in terms of assessing the alignment of student model behavior in comparison to teacher models. To that end, post-hoc explanations function as a proxy for measuring behavior as explanations aim to measure the contribution of certain input variables to the final prediction.

Within this paper, we aim to do a thorough analysis of the alignment of post-hoc explanations between various LLMs and their distilled versions. Furthermore, we also evaluate the robustness of each

Final submission for CS 282.

student model and compare their performance with the teacher on various language tasks. Finally, to improve upon the post-hoc explanations between the student and teacher, we propose a new method of aligning explanations, dubbed AlignedDistil, by aligning the attention weights of the teacher and student during the knowledge distillation process.

2 Related Work

2.1 Large Langage Model Knowledge Distillation

There has been significant research efforts to improve distillation methods [18] to generate performant, "student" language models that are much smaller than their "teacher" models. Within NLP, distillation methods can be partitioned roughly into two separate categories: one-stage methods and two-stage methods.

One-stage methods largely perform distillation at the fine-tuning stage. BiLSTMSOFT [23] performs knowledge distillation with the teacher's logits on an augmented dataset, BERT-PKD [20] performs knowledge distillation with the teacher's logits and hidden states, and PD [24] uses a small pretrained masked language model as a student to enhance the effect of distillation.

Two-stage methods perform distillation at both the pre-training stage and the fine-tuning stage. DistilBERT [18] and MobileBERT [21] focus on the pre-training stage, aiming to get a task-agnostic model that can be fine-tuned or distilled on downstream tasks. TinyBERT [7] first distills a task-agnostic model during pre-training and then performs task-specific distillation on an augmented dataset to further improve performance.

2.2 Interpretability and Knowledge Distillation

In the context of interpretability, knowledge distillation has more traditionally involved distilling a complex model into more inherently interpretable models such as decision trees. Liu et al. [9] specifically apply KD to distill Deep Neural Networks into decision trees, but they do not consider the evaluation of the interpretability of neural network implementations of distilled, "student" language models.

Looking specifically at distilling NNs into smaller NNs, there is a growing research area within distillation research that incorporates gradients as an important factor in distillation, whether it be aligning gradients [19] or using gradients as a weighting factor [29]. However, these methods are mostly focused on the computer vision field with limited implementations for LLMs. Wang et al. [26] propose a new KD method called Gradient Knowledge Distillation (GKD) which is the first to implement gradient alignment in the context of LLMs. Using a gradient alignment objective as part of the distillation process, Wang et al. [26] are able to achieve improved performance compared to existing KD methods while also improving consistency in terms of behavior. However, the way in which they prove that the behavior is more consistent with the teacher model is through a gradient-based assessment (Saliency Loyalty which is derived from the Grad method applied to the word saliency distribution from the teacher and student models respectively) which may point to confirmation bias instead of a real assessment.

However, the contribution by Ding et al. [4] of saliency-based word alignment interpretation of convolutional Neural Machine Translation (NMT) is a possible step toward interpreting LLMs without relying on external models.

3 Preliminaries

3.1 KD for LLMs

Vanilla Knowledge Distillation aims to train the student model not only on the task-specific object but to also use the soft targets produced by the teacher. More formally, we can write for any classification task, $\mathcal{D} := \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ is the training dataset with N instances, where $\mathbf{x}_i$ is the input sequence and $\mathbf{y}_i$ is the assigned label. Letting T represent the teacher model and S represent the student model, let's now define f^T and f^S to be the *behavior* functions of the teacher and student models respectively. *Behavior* function in this context means some function f^M that maps the inputs of a model M

to some informative representation such as the soft-targets [6], hidden states [18, 20] or attention heatmaps [27],

With these *behavior* functions f^T and f^S in hand, we can formally describe the Knowledge Distillation process as minimizing the objective function(s) which can be written as

$$\mathcal{L}_{\mathrm{KD}} \coloneqq \sum_{\mathbf{x}_i \in \mathcal{D}} \mathcal{L}\left(f^S(\mathbf{x}_i), f^T(\mathbf{x}_i)\right) \tag{1}$$

where $\mathcal{L}(\cdot)$ is some loss function that evaluates the difference between the output representations of T and S. Some KD methods utilize multiple difference objectives which result in $\mathcal{L}_{\mathrm{KD}}$ being a linear combination of multiple other objective functions, often including $\mathcal{L}_{\mathrm{CE}}$ as the cross-entropy loss of the task-specific objective.

4 Problem Statement & Methodology

We aimed to compare several post-hoc explanations methods as they are applied to BERT and distilled versions of BERT across a variety of tasks (GLUE benchmark [25]), to see if post-hoc explanations are preserved during distillation. To be able to use BERT and its distilled variants on the GLUE tasks, we first finetuned BERT and its distilled variants on different GLUE metrics. We were then able to run a number of post-hoc explanations and then compare the token-level or word-level attributions between a distilled BERT model and BERT. The three methods we have chosen are SHAP, LIME, and Integrated gradients, mainly for their ability to act as a blackbox against models to generate explanations, as well as their wide-implementation and usage in the academic community. We also perform sentence perturbations to evaluate robustness, comparing evaluation metrics to the original base model for each GLUE task. After that, we design our own distillation method, attention weight alignment, that optimizes for preserving post-hoc explanations as a method of aligning a student model with the teacher model.

4.1 Post-hoc explanations

We used different post-hoc methods to general local explanations for the trained models, both the teacher and the students, on the validation dataset. The alignment of explanations, or the similarity between explanations, was then measured with cosine similarity between attribution vectors.

SHAP We used the `shap` library to generate local SHAP explanations [11]. Using a language model as a predictor, we grab the logits for the positive, or "1", class to view the attributions towards that prediction. We also modified its highlighting functionality by outputting the highlight directly to HTML, allowing for easier integration in web-based contexts.

LIME We used the `lime` library to generate local LIME explanations [16]. We limited the total number of features to 20 and restricted the number of samples to 100 to reduce memory usage across GPUs. In order to compare token attributions across models, we only grabbed the attributions of tokens that were present in both teacher and student models for consistent comparisons, since LIME does not provide attributions to the same set of top tokens in a given example for different models.

Integrated Gradients For Integrated Gradients, we used `captum` [8], a model interpretability library that comes with an implementation of Integrated Gradients [22] for PyTorch. For each input sentence (passed in as a list of tokens), we used as our baseline the concatenation of the tokens `[CLS]`, `[PAD]`, and `[SEP]` such that the sequence

```
[CLS] + length * [PAD] + [SEP]
```

is the same length as the tokenized sentence that is passed as input to the model. We are then able to compute on the word embedding layer the integrated gradients of the input relative to the baseline. This outputs a 2-dimensional tensor which we can then use to compute the individual contribution of each word toward the prediction.

4.2 Robustness

To model the robustness of the models, we designed input perturbation methods as inspired by [14] and implemented them with the `nlpaug` library [13] and the `pattern.en` module [2] in Python.

The amount of perturbations is kept infinitesimal such that the ground truth labels/values should not change. Each validation data set (one for each task) was perturbed randomly 10 times to avoid random-chance results. We compared the model performance on unperturbed data and the average model performance on perturbed data to have a sense of the degree of robustness.

Random character substitution A character is randomly selected and substituted with another character in the alphabet.

Swapping adjacent words A word is randomly selected to be swapped with its adjacent word. The intent is to investigate whether a language model is sensitive to word order or only attends to the presence of certain words.

Synonym replacement A word is randomly selected to be replaced with a synonym from the WordNet lexical database [5].

Character substitution with common keyboard typo A character is randomly selected and replaced with a common keyboard typo based on the simulator.

Spelling mistake A word is randomly selected and replaced with a version with spelling mistakes (e.g. "receive" v.s. "recieve") from the misspelling dictionary `spelling_en.txt` [2].

Verb tense change All verbs are randomly mapped to themselves or different tenses (e.g. "receive" v.s. ["receive", "received", "receives", ...]).

4.3 Attention Weight Alignment

To improve on the post-hot explanation alignment, we hypothesized that aligning attention weights during distillation could yield a student model that better preserves teacher explanations. To achieve this, we interchanged the student's attention with a compressed version, such as taking the average, of the teacher weights: the i-th student layer align with the $2i$-th and the $(2i+1)$-th teacher layers.

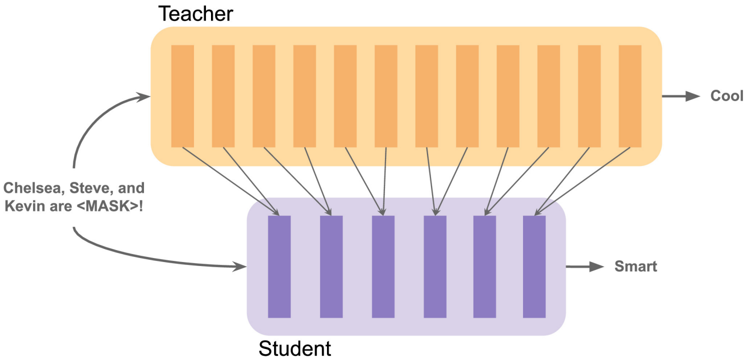

Figure 1: Attention weight alignment in the distillation process

Based off of the proposed approach shown above, we defined our distillation objectives based on work done in [28] and they are as follows:

- $\mathcal{L}_{\text{MLM}} = \sum_i \text{CE}(f^S(\mathbf{x}_i), \mathbf{y}_i)$ which measures divergence between student output logits and ground truth on masked tokens using cross-entropy loss.
- $\mathcal{L}_{\text{CE}} = \sum_i \text{CE}(f^S(\mathbf{x}_i), f^T(\mathbf{x}_i))$ which measures divergence between student and teacher's output logits on masked tokens using cross-entropy loss.
- $\mathcal{L}_{\text{Cos}} = \sum_i \text{Cos}(f^S_{\text{FinalLayer}}(\mathbf{x}_i), f^T_{\text{FinalLayer}}(\mathbf{x}_i))$ which measures divergence of contextualized representations on masked tokens in the last layer between teacher and student using cosine-embedding loss.
- $\mathcal{L}_{\text{Att}} = \sum_i \text{CE}_{\text{S}}(\text{Interchange}(f^S, N_T)(\mathbf{x}_i), f^T(\mathbf{x}_i))$ which measures divergence between student output logits after interchange and teacher output logits using smoothed-cross-entropy loss.

5 Experimental Results

We fine-tuned the teacher model BERT, and the students, DitilBERT [18], MobileBERT [21], and TinyBERT [7] on individual glue tasks and evaluated them with methods described in Section 4. For optimization, we performed distillation with attention weight alignment and trained with the specified losses as introduced in Section 4.3 specifically, and the resulting model was evaluated similarly.

5.1 Data & Tasks

GLUE benchmark is designed to test models' ability to perform well on a variety of NLU tasks. Below are the tasks and the respective data we worked on:

WNLI Determine whether a given sentence entails, contradicts, or is neutral with respect to another.

CoLA Determine whether a given sentence is grammatically correct.

STS-B Determine the semantic similarity between pairs of sentences.

SST-2 Classify positive or negative sentiment of movie reviews

MRPC Determine whether two given sentences are paraphrases of each other.

RTE Determine whether a given pair of sentences have an entailment relationship.

QNLI Determine whether, in a given pair of sentences, one sentence is a question and the other is a potential answer.

5.2 KD Training

For knowledge distillation of AlignedDistil, we used pretrained weights of DistilBERT as a baseline. We then trained with a batch size of 40 for 3 epochs on the Wikitext dataset, and update the gradient every 6 batches. We distilled on a single A5000 GPU with 24 GB of memory for a total time of 72 hours. Note: as we perform backpropagation twice on the teacher and student model, the batch size effectively doubles, requiring a higher GPU memory amount.

We then evaluate the model performance on GLUE tasks with the distilled model, comparing against BERT and DistilBERT.

5.3 Evaluation on existing distillation methods

Robustness Tasks such as WNLI, SST-2, and QNLI show little deviation from the base evaluation metric across different models, i.e. similarly good robustness. For example, note that MobileBERT on the SST-2 task has a lower accuracy than DistilBERT, but relative to the base validation set without perturbations the accuracy is around the same. However, we see a major deviation in Matthews correlation for the CoLA task across models, which might be due to the nature of the task - grammatical correctness. We also find deviation in accuracy and F1 score for the MRPC task, specifically to BERT. The other models, interestingly enough, show little difference from the original metric. See Table 1 for results on SST-2 task, and all results for other tasks in the appendix (Table 3).

Perturbation	BERT	DistilBERT	MobileBERT	TinyBERT
/	0.9197	**0.9037**	0.6743	0.8933
Char - Replace	0.9005	**0.8834**	0.6462	0.8644
Word - Swap	0.9186	**0.9026**	0.6607	0.8864
Word - Synonym	0.9080	**0.8905**	0.6620	0.8794
Keyboard Typo	0.6247	0.5946	0.5424	**0.6239**
Spelling Mistake	0.9052	**0.8886**	0.6586	0.8710
Verb Tense Change	0.8608	**0.8278**	0.6669	0.8107

Table 1: SST-2 task - model accuracy over original validation data and model accuracy over perturbed data, averaged over 10 times perturbation.

Explanation Alignment DistilBERT's explanations are most similar across tasks and explanation methods. However, that is not necessarily an indication of whether the distillation process for

DistilBERT is any better than MobileBERT's or TinyBERT's; only that its distillation process keeps explanations consistent. This is especially clear when only for a few tasks are the results anywhere close to similar (CoLA and SST-2) while all the others have either very low or even negative average cosine similarities. For the other distillation methods, we find their cosine similarities to fall generally less than DistilBERT, with some tasks and methods actually beating DistilBERT, such as with LIME and SHAP with MRPC on TinyBERT. See Table 2 for average cosine similarity between explanations and Figure 2 for the distribution of cosine similarities between explanations.

5.4 Evaluation on AlignedDistil

Explanation Alignment We see in Table 2 that AlignedDistil performs relatively worse than DistilBERT on nearly all tasks and explanation methods, with the sole exception being MRPC on Integrated Gradients. That being said, however, we find that some explanations are relatively close to DistilBERT, such as with MRPC on SHAP, QNLI on LIME/SHAP, and RTE on all explanation methods. Therefore, this shows a promising line of research towards aligning the attention weights, given that our method is relatively simple; a more robust and mathematically sound attention weight compression could lead to even better results, potentially beating out DistilBERT.

Task	Explanation	DistilBERT	MobileBERT	TinyBERT	AlignedDistil
WNLI	LIME	**0.3672**	0.2995	0.1822	-0.0285
	SHAP	-0.2008	**0.2798**	-0.2829	0.2782
	Int Grad	-0.0755	**-0.0665**	-0.2143	-0.2229
CoLA	LIME	**0.6654**	-0.1291	-0.0017	0.0113
	SHAP	**0.6248**	-0.0482	-0.2009	-0.3320
	Int Grad	**0.3233**	-0.0182	-0.1106	-0.1034
SST-2	LIME	**0.8446**	0.2901	0.7859	-0.0224
	SHAP	**0.8503**	0.1469	0.7960	-0.0537
	Int Grad	**0.4192**	-0.0534	0.3151	-0.2083
MRPC	LIME	0.4612	-0.0140	**0.4794**	0.0277
	SHAP	-0.0622	-0.1594	**0.0914**	-0.1686
	Int Grad	0.0451	-0.0024	0.0203	**0.1770**
RTE	LIME	-0.0606	**0.0671**	-0.0170	-0.0917
	SHAP	-0.2212	-0.2777	**-0.1709**	-0.3036
	Int Grad	0.0745	-0.0316	**0.1086**	0.0484
QNLI	LIME	-0.0138	**-0.0071**	-0.0093	-0.0149
	SHAP	**0.0749**	-0.0261	-0.0017	0.0245
	Int Grad	**0.1889**	0.0600	0.1527	-0.4639

Table 2: Average cosine similarity between explanations built on student v.s. on teacher BERT on the validation dataset.

6 Conclusion

In this paper, we analysed the behavior alignment between BERT and various distilled versions using post-hoc explanations as a proxy for model behavior. We observed that as a whole, current distillation approaches fail to mimic the explanation behavior of the original teacher model indicating that there is still a gap in terms of fully capturing the latent knowledge of the teacher. AlignedDistill is an attempt at filling that gap with the fundamental intuition being that attention gives some indication of the internal explanatory behavior for LLMs. AlignedDistill was able to show some improvement on existing distillation methods and indicates that attention head interchange is a promising avenue for explanation alignment.

There were some limitations to our distillation approach. Specifically, our method of assigning the i-th layer of the student model with the average of the $2i$-th and the $(2i + 1)$-th layers of the teacher model enforced a specific mapping between the teacher and student models which might not make sense. For future work, we aim to find better ways to incorporate attention weight permutation without enforcing a certain mapping.

References

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] T. De Smedt and W. Daelemans. Pattern for python. *The Journal of Machine Learning Research*, 13(1):2063–2067, 2012.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] S. Ding, H. Xu, and P. Koehn. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[5] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015.

[7] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling bert for natural language understanding, 2020.

[8] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[9] X. Liu, X. Wang, and S. Matwin. Improving the interpretability of deep neural networks with knowledge distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 905–912, 2018.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[11] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions, 2017.

[12] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[13] E. Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

[14] M. Moradi and M. Samwald. Evaluating the robustness of neural language models to input perturbations, 2021.

[15] OpenAI. Gpt-4 technical report, 2023.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[19] S. Srinivas and F. Fleuret. Knowledge transfer with Jacobian matching. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4723–4731. PMLR, 10–15 Jul 2018.

[20] S. Sun, Y. Cheng, Z. Gan, and J. Liu. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[21] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices, 2020.

[22] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks, 2017.

[23] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin. Distilling task-specific knowledge from bert into simple neural networks, 2019.

[24] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova. Well-read students learn better: On the importance of pre-training compact models, 2019.

[25] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

[26] L. Wang, L. Li, and X. Sun. Gradient knowledge distillation for pre-trained language models, 2022.

[27] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

[28] Z. Wu, A. Geiger, J. Rozner, E. Kreiss, H. Lu, T. Icard, C. Potts, and N. Goodman. Causal distillation for language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4295, Seattle, United States, July 2022. Association for Computational Linguistics.

[29] Y. Zhu and Y. Wang. Student customized knowledge distillation: Bridging the gap between student and teacher. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5037–5046, 2021.

Appendix

Task	Perturbation	Metric	BERT	DistilBERT	MobileBERT	TinyBERT
WNLI	/	Accuracy	0.5634	0.4648	0.4648	0.5634
	Char - Replace	Accuracy	0.5634	0.5197	0.5282	0.5634
	Word - Swap	Accuracy	0.5634	0.5169	0.4718	0.5634
	Word - Synonym	Accuracy	0.5634	0.4972	0.4887	0.5634
	Keyboard Typo	Accuracy	0.5634	0.5634	0.5620	0.5634
	Spelling Mistake	Accuracy	0.5634	0.5197	0.5197	0.5634
	Verb Tense Change	Accuracy	0.5634	0.5437	0.4563	0.5634
CoLA	/	Matthews Corr	0.6041	0.5089	-0.02	0
	Char - Replace	Matthews Corr	0.3527	0.3109	-0.0458	0
	Word - Swap	Matthews Corr	0.1492	0.1462	-0.0258	0
	Word - Synonym	Matthews Corr	0.3888	0.3484	-0.0287	0
	Keyboard Typo	Matthews Corr	0.0511	0.0377	-0.0103	0
	Spelling Mistake	Matthews Corr	0.2975	0.2343	-0.0470	0
	Verb Tense Change	Matthews Corr	0.2799	0.2538	-0.0266	0
MRPC	/	Accuracy	0.4142	0.8407	0.6005	0.8235
	/	F1	0.2508	0.8893	0.7155	0.8710
	Char - Replace	Accuracy	0.4221	0.8238	0.6100	0.8088
	Char - Replace	F1	0.2681	0.8743	0.7221	0.8559
	Word - Swap	Accuracy	0.4733	0.8299	0.6130	0.8223
	Word - Swap	F1	0.3736	0.8823	0.7273	0.8689
	Word - Synonym	Accuracy	0.4627	0.8368	0.6120	0.8127
	Word - Synonym	F1	0.3529	0.8864	0.7261	0.8601
	Keyboard Typo	Accuracy	0.3162	0.4696	0.5877	0.5311
	Keyboard Typo	F1	0.0000	0.3858	0.7122	0.5070
	Spelling Mistake	Accuracy	0.4429	0.8284	0.6098	0.8145
	Spelling Mistake	F1	0.3126	0.8792	0.7235	0.8613
	Verb Tense Change	Accuracy	0.4502	0.8387	0.6211	0.8081
	Verb Tense Change	F1	0.3285	0.8858	0.7376	0.8562
RTE	/	Accuracy	0.4657	0.5812	0.4440	0.5523
	Char - Replace	Accuracy	0.4697	0.5921	0.4426	0.5426
	Word - Swap	Accuracy	0.4686	0.5971	0.4458	0.5542
	Word - Synonym	Accuracy	0.4679	0.5957	0.4444	0.5531
	Keyboard Typo	Accuracy	0.4729	0.5397	0.4592	0.5097
	Spelling Mistake	Accuracy	0.4675	0.5964	0.4419	0.5502
	Verb Tense Change	Accuracy	0.4733	0.5693	0.4632	0.5282
QNLI	/	Accuracy	0.4946	0.8448	0.6063	0.7986
	Char - Replace	Accuracy	0.4946	0.8273	0.5887	0.7853
	Word - Swap	Accuracy	0.4946	0.8429	0.5923	0.7903
	Word - Synonym	Accuracy	0.4946	0.8386	0.5928	0.7901
	Keyboard Typo	Accuracy	0.4946	0.6633	0.5459	0.6845
	Spelling Mistake	Accuracy	0.4946	0.8346	0.5916	0.7865
	Verb Tense Change	Accuracy	0.4946	0.8278	0.6009	0.7738
STS-B	/	Pearson Corr	0.2435	0.1936	0.1146	0.0998
	/	Spearman Corr	0.2132	0.1915	0.2924	0.0857
	Char - Replace	Pearson Corr	0.2529	0.2270	0.1286	0.0314
	Char - Replace	Spearman Corr	0.2122	0.2179	0.2669	0.0159
	Word - Swap	Pearson Corr	0.2578	0.2244	0.1221	0.0140
	Word - Swap	Spearman Corr	0.2098	0.2157	0.2831	-0.0022
	Word - Synonym	Pearson Corr	0.2528	0.2195	0.1379	0.0269
	Word - Synonym	Spearman Corr	0.2096	0.2140	0.2879	0.0096
	Keyboard Typo	Pearson Corr	0.1198	0.2489	0.0793	-0.0066
	Keyboard Typo	Spearman Corr	0.0986	0.2349	0.1588	-0.0050
	Spelling Mistake	Pearson Corr	0.2566	0.2298	0.1339	0.0284
	Spelling Mistake	Spearman Corr	0.2130	0.2209	0.2799	0.0107
	Verb Tense Change	Pearson Corr	0.2269	0.2103	0.1038	0.0675
	Verb Tense Change	Spearman Corr	0.1995	0.2078	0.2823	0.0525

Table 3: Model performance over perturbed data, averaged over 10 times perturbation.

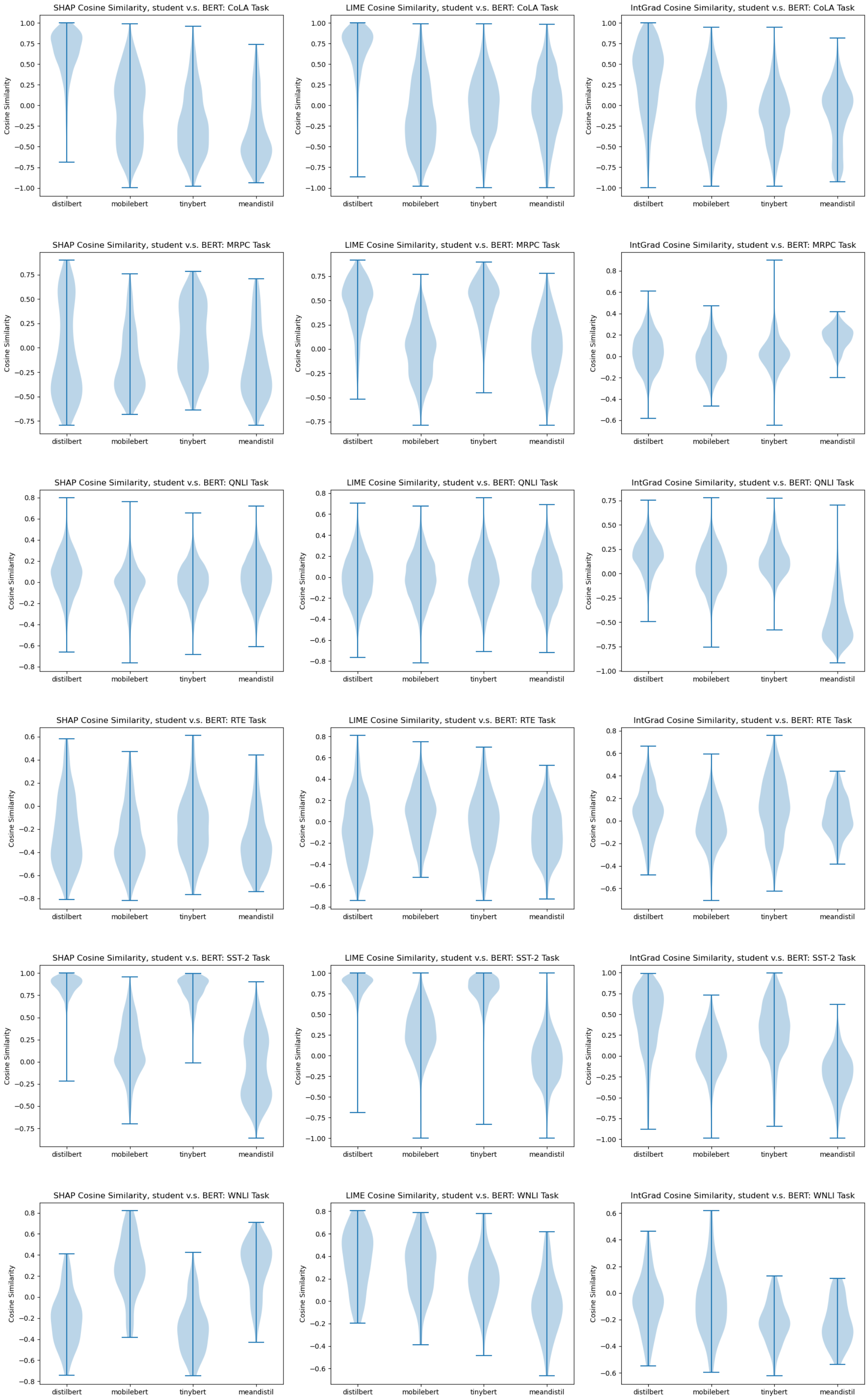

Figure 2: Cosine Similarity between student and teacher (BERT) explanations on the validation GLUE dataset.