TRANSFORMER AUGMENTATIONS FOR THE INVERSE SCALING PROBLEM

Pranav Hegde, Satyanarayana Mulagala, Shaili Shah & Shuchi Talati
Department of Computer Science
Arizona State University
Tempe, AZ 85281, USA
{phegde7,smulaga2,sshah205,stalati1}@asu.edu

ABSTRACT

Large Language Models exhibit worsening performance with an increase in model size for certain task types, which is known as the Inverse Scaling Problem. In this paper, we propose an augmentation to the transformer architecture to guide it to pay attention to certain crucial linguistic features. We find that the augmented transformer model does perform better on certain inverse scaling tasks whose data includes these linguistic features. However, the gains are quite minimal and we find that the inverse scaling problem is not entirely solved, even though the performance of the models improved.

1 PROBLEM STATEMENT

Inverse Scaling (IS) is the phenomenon where task performance worsens as the large language model scales and the training loss increases. While the model performs better on its training objective as the size increases, it exhibits worsening performance on certain subtasks. This goes against the common conception of Deep Learning i.e., the more data the better the performance. It is acknowledged that understanding scalability and its connection with the transformer architecture can have significant practical impacts. Mckenzie et al. (2023) collected 11 datasets that exhibit IS behavior. It is hypothesized that as models scale, they might lose their ability to generalize and rather just begin to memorize the training data. Thus, the transform architecture might need to be modified to increase the capability of models to generalize and prevent memorization.

2 APPROACH

Minor changes to a sentence can change its meaning entirely. A single word such as 'not' could change a positive sentence to a negative one. Transformers are unable to capture the weight of such words/groups of words effectively. For example, the NeQA dataset in the Inverse Scaling paper (Mckenzie et al., 2023) tests if transformers can handle answering negated questions and it is shown that they perform poorly and also see a decrease in performance with bigger models.

Kovaleva et al. (2019) showed that many of the attention heads simply pay attention to the [CLS] and [SEP] tokens in BERT. This might indicate that the training objective is not sufficient enough to pay much attention to other tokens and features of the sequence.

We propose an augmentation to the transformer loss function, derived from the works of Deshpande & Narasimhan (2020), to guide attention heads to pay more attention to such words and other linguistic features. The method is described below.

Let a single head's attention activations, which is a function of s, be denoted by the following

$$\mathbf{H}(s) = softmax(\frac{QK^T}{\sqrt{d_k}}) \in \mathbb{R}^{n \times n} \tag{1}$$

A mean squared error loss using a predefined pattern P is imposed on H to guide the attention heads.

$$L_{attn}(\mathbf{H},\mathbf{P}) = \|\mathbf{H} - \mathbf{P}\|_F^2 \tag{2}$$

An example of P which makes heads focus their attention on the word 'not' in a sequence is shown below

$$\mathbf{P}_{[not]}[p,q] = \begin{cases} 1 & \text{if q = 'not'} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The overall attention guidance loss is defined as

$$L_{AG}(\mathbf{x}) = \sum_{k=1}^{l} \sum_{j=1}^{h} L_{attn}(\mathbf{H}_{kj}, \mathbf{P}_{kj}) \times \mathbb{I}(k,j) \tag{4}$$

where $\mathbb{I}(k,j)$ denotes an indicator function which is 1 only if head j in layer k is being guided. This loss is added to the original transformer loss function and the model is trained based on this new loss to guide the attention heads.

We hypothesize that the augmentation to the loss function will force the attention heads to pay more attention to crucial linguistic features which can completely change the meaning of a sentence. This will enable the transformer to better model the differences between sentences that have completely different meanings when a few words are changed.

We performed three different experiments with this method on the **gpt2** and **gpt2-medium** models, guiding different heads to different linguistic features in each experiment. To keep things simple, we guided a particular head number to the same pattern for all layers, i.e. $\mathbf{P}_{\cdot j}$ was the same for all layers. In experiment 1, we guided head 0 to pay attention to a set of 27 commonly occurring negation tokens. In **experiment 2**, heads 0, 1, and 2 were guided to the same set of tokens. In **experiment 4**, we aimed to guide the heads to multiple types of transition words in the English language. Head 0 paid attention to tokens that enforce order (e.g. 'Finally'), head 1 to negation tokens, head 2 to tokens that are used to add further information (e.g. 'Moreover'), and head 3 to tokens that add emphasis (e.g. 'Obviously').

The set of tokens that were used for the fine-tuning of various models are listed below

CONTRAST TOKENS = ['Not', ' not', 'But', ' but', 'Though', ' though', 'Unlike', ' unlike', 'Nevertheless', ' nevertheless', 'Nonetheless', ' nonetheless', 'Despite', ' despite', 'Cont', ' cont', 'rast', 'Cont', ' cont', 'rary', 'Whereas', ' whereas', 'Alternatively', ' alternatively', 'Con', ' con', 'versely']

ORDER TOKENS = ['Following', ' following', 'Previously', ' previously', 'First', ' first', 'Second', ' second', 'Third', ' third', 'Finally', ' finally', 'Sub', ' sub', 'sequently', 'Before', ' before', 'Fore', ' fore', 'most']

ADDITION TOKENS = ['Too', ' too', 'Besides', ' besides', 'add', ' add', 'itionally', 'Moreover', ' moreover', 'Furthermore', ' furthermore', 'Also', ' also']

EMPHASIS TOKENS = ['Und', ' und', 'oubtedly', 'Un', ' un', 'question', 'ably', 'Obviously', ' obviously', 'Part', ' part', 'icularly', 'Especially', ' especially', 'Clearly', ' clearly', 'Import', ' import', 'antly', 'Def', ' def', 'initely', 'Absolutely', ' absolutely', 'Indeed', ' indeed', 'Never', ' never']

Due to resource and time constraints, the models were fine-tuned only on the first one million rows of the OpenWebText Dataset for two epochs. The vanilla gpt2 models without any augmentations were also fine-tuned on the same dataset to conduct accurate comparisons and ensure the accuracy changes obtained were not just due to the fine-tuning process. The gpt-medium model was trained for one epoch.

3 RESULTS

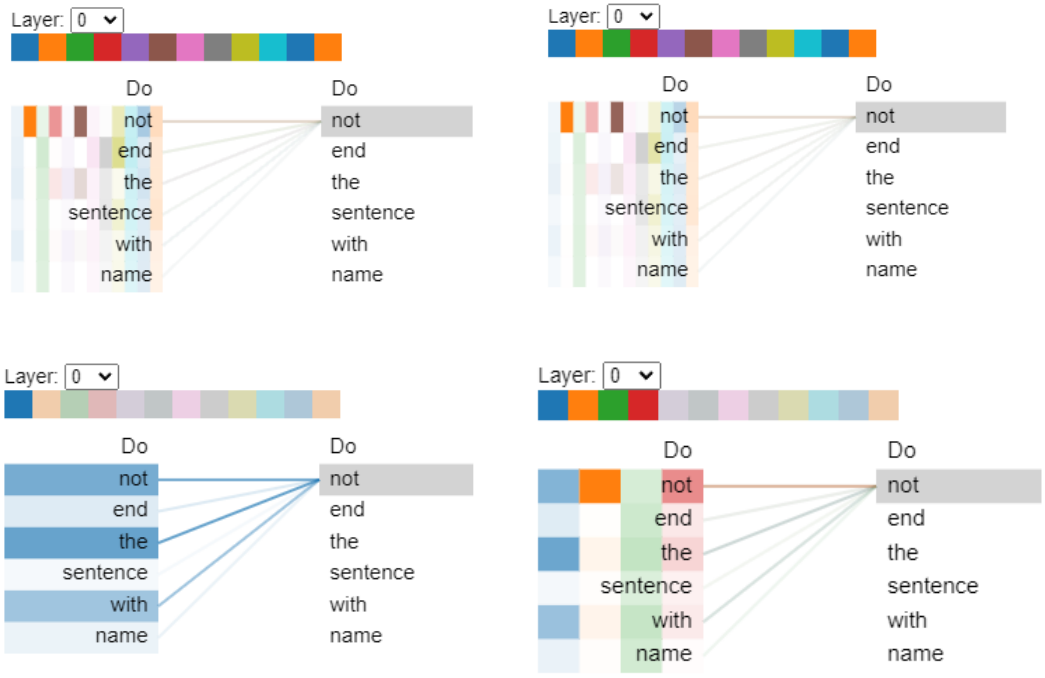

Figure 1: **The above graphs visualize the attention to the word 'not' in the given sentence. (a) Top left: gpt2 base model attention (b) Top right: gpt2 fine-tuned model (c) Bottom left: gpt2 fine-tuned on single attention head [experiment 1] (d) Bottom right: gpt2 fine-tuned on 4 attention heads on various transition tokens [experiment 3]**

The attention heads for various experiments are illustrated using 'bertviz' visualizing tool. The opacity of the head in Figure 1 illustrates the amount of attention that each word is paying to another. As we can observe in the figure above, after fine-tuning on the contrast words set, a higher amount of attention is paid to the word 'not' in the given sentence, indicated by the brighter lines on the images in the bottom row. This shows that fine-tuning with the attention guidance loss indeed guides the attention heads to pay attention to certain tokens in a sequence.

	GPT2				
	original	finetuned	1-negation	3-negation	4-transition
repetitive-algebra	0.204	0.301	0.29	0.393	**0.462**
pattern-matching-suppression	**0.077**	0.0693	0.0686	0.0574	0.0756
redefine	0.6639	0.6439	0.6471	0.6567	0.6302
resisting-correction	0.9965	0.9962	0.996	0.9949	0.9952
into-the-unknown	**0.4934**	**0.4934**	**0.4934**	0.4929	**0.4934**
memo-trap	0.7382	0.7372	0.735	0.7339	**0.7393**
modus-tollens	0.1634	0.2152	0.1861	0.1796	0.1861
sig-figs	0.3915	0.3915	0.3915	0.3915	0.3915
hindsight-neglect	0.4635	0.5016	0.5079	0.4762	**0.5111**
neqa	0.4567	0.4567	0.4567	0.4567	0.4567

Table 1: Accuracies of gpt2 model for the inverse scaling datasets

	GPT2 Medium				
	original	finetuned	1-negation	3-negation	4-transition
repetitive-algebra	0.067	0.069	0.067	0.41	0.377
pattern-matching-suppression	0.0007	0.0	0.0	0.0007	0.0
redefine	**0.6833**	0.6736	0.6688	0.6535	0.6712
resisting-correction	0.9944	0.9964	**0.9969**	0.9955	0.9958
into-the-unknown	0.4803	0.4868	0.4874	0.4923	0.4929
memo-trap	0.6410	0.6357	0.6335	0.625	0.6314
modus-tollens	**0.9992**	**0.9992**	**0.9992**	**0.9992**	0.9798
sig-figs	**0.3980**	0.3925	0.3934	0.3913	0.3903
hindsight-neglect	0.4825	0.4634	0.4571	0.4539	0.4571
neqa	0.4533	0.4567	0.4567	**0.46**	0.4567

Table 2: Accuracies of gpt2-medium models for the inverse scaling datasets

Table **??** and 2 show the performances of gpt2 and gpt2-medium models on the inverse scaling datasets. The numbers in bold indicate the best-performing models for that particular dataset.

4 Analysis and Findings

We notice that overall at least one of the augmentations indeed helps in improving the accuracy of the models, both gpt2 and gpt2-medium for most datasets. However, most datasets do not notice significant performance improvements to overcome the inverse scaling problem in the case of gpt2-medium.

The gpt2 4-transition model which was guided to various English transition tokens, shows the best performance in four inverse scaling datasets. However, the same performance does not translate to gpt2-medium, with performance worsening in some datasets compared to vanilla gpt2.

Another interesting finding is that the 3-contrast model showed improved performance in datasets that contained a high number of negation tokens. Compared to vanilla gpt2, it shows improvement in the neqa, into-the-unknown, and resisting-correction datasets where more than fifty percent of the training examples contain negation tokens. Conversely, the datasets where its performance degraded had close to zero percent examples with negation tokens. This could imply that guiding the attention heads to focus on different linguistic features affects the model's capability to understand sentences.

While the augmentations did improve the performance of the individual gpt2 and gpt2-medium models, they were not able to overcome the inverse scaling problem. This suggests that the models might still be memorizing the training data. It could also imply that the subset of the dataset we trained on was not enough to overcome the problem, and further training on the entire OpenWebText dataset might help improve performance. Another hypothesis is that the models need to be trained on guided attention heads in the pre-training stage, rather than fine-tuning stage where model parameters are not able to change as freely as in the pre-training step.

We didn't see much improvement in the fine-tuned gpt2 model for the modus-tollens classification dataset using this technique. The gpt2-medium model achieves almost perfect accuracy on this dataset and only the even bigger models exhibit inverse scaling performance, which we were not able to train on due to time constraints. Thus further work could involve training the larger gpt2 models to compare performance.

Further works could also include capturing various linguistic features using the attention guidance mechanism, as our current work focuses only on a few transition words, and exhibits performance improvements only in cases where these tokens exist. Including a wide range of linguistic features could help the model encompass the entire corpora and perform better. The same methods could also be used, but on the entire OpenWebText dataset rather than a subset, and analyze if the performance improves.

5 INDIVIDUAL CONTRIBUTIONS

Pranav Hegde: Attention Guidance Implementation, Inverse Scaling Inference implementation, Sol environment setup, training gpt2 3-negation and gpt2-medium 3-negation models

Shaili Tarun Shah: Fine-tuned GPT2 and GPT2-medium to compare the experiment results, worked on visualizing all the fine-tuned models from all the experiments to analyze the results using bertviz

Satyanarayana: Implemented Inverse scaling Inference to validate whether the model performance is consistent with Inverse-Scaling paper results. Trained gpt2 and gpt2-medium using attention guidance pattern with 4 different categories of tokens (transition words) assigned to each attention head. Performed analysis on model performance on datasets to find the data points where the model performance is worse so that we can incorporate some changes in the later iterations.

Shuchi Talati: Compiled and tokenized English transition words to feed model, implementation of inverse scaling inference, familiarization and setup of Sol environment, trained gpt2 and gpt2-medium using attention guidance pattern with contrast words on a single attention head and completed project documentation.

REFERENCES

Ameet Deshpande and Karthik Narasimhan. Guiding attention for self-supervised learning with transformers. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4676–4686, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.419. URL https://aclanthology.org/2020.findings-emnlp.419.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4365–4374, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL https://aclanthology.org/D19-1445.

Ian. R. Mckenzie, Alexander Lyzhov, Michael Martin Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Sam Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *ArXiv*, 2023.