Transformer Augmentations for Inverse Scaling Problem

Corpus Crusaders

Pranav Hegde, Shaili Tarun Shah, Shuchi Talati, Satyanarayana

Problem Statement

- Inverse Scaling (IS) is the phenomenon where task performance worsens as the large language model scales and the training loss decreases.

- This goes against the common conception of Deep Learning i.e., the more data the better the performance.

- Understanding scalability and its connection with the transformer architecture can have significant practical impacts.

- Thus, the transform architecture might need to be modified to increase the capability of models to generalize and prevent memorization.

Transformer Shortcomings

- Minor changes to a sentence can change its meaning entirely. A single word such as ‘not’ could change a positive sentence to negative.

- Transformers are unable to capture the weight of such words/groups of words effectively.

- Its shown that many of the attention heads simply pay attention to the [CLS] and [SEP] tokens in BERT.

- Training objective not sufficient enough to pay much attention to other tokens and features of the sequence.

Attention Guidance

Augment the transformer loss function to guide attention heads to pay more attention to such words and other linguistic features.

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

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

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

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

Force the attention heads to pay more attention to crucial linguistic features

What to pay attention to ?

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’]

Experiments

1. Guide 1 attention head on contrast tokens

2. Guide 3 attention heads on contrast tokens

3. Guide 4 different attention heads on contrast, order, addition, and emphasis tokens

We also trained gpt2 without any modifications to compare the results.

Results

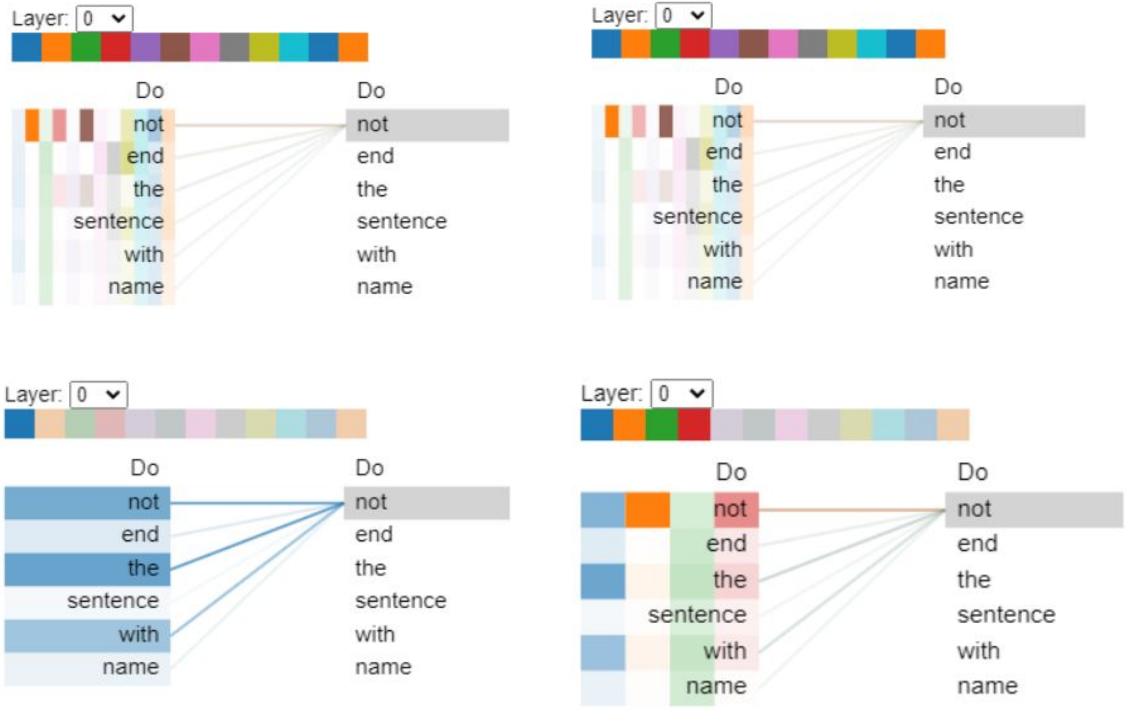

Figure 1: The above graphs visualize the attention to the word 'not' in the given sentence.(a) Top left: gpt2 base model attention (b) Top right: gpt2 fine-tuned model (c) Bottom left: gpt2 fine-tuned on single attention head [experiment 1] (d) Bottom right: gpt2 fine-tuned on 4 attention heads on various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

Results

	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

Results

	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

Analysis and Findings

- The gpt2 4-transition model, shows the best performance in four inverse scaling datasets.

- The 3-contrast model showed improved performance in datasets that contained a high number of negation tokens. Overcomes inverse scaling for neqa dataset.

- Both gpt2 models showed some performance improvements on the IS datasets. However, not enough to overcome the inverse scaling problem.

Future Work

- Finetune on gpt2-large models
- Model and include a wide range of linguistic features
- Train on the entire OpenWebText dataset rather than a subset

Learnings

- Deep understanding of the transformer architecture & its components.
- Model modification techniques through survey of transformers.
- How to leverage pre-trained models for improved performance by fine-tuning.
- How to use huggingface to train, fine-tune and use large language models.
- Team collaboration, effective communication are crucial for project success.

Q and A