CSE 576 : Topics/Natural Language Proc (2023 Fall)

Final Project Progress Presentation

Auto Contrastive Decoding to tackle the Inverse Scaling problem

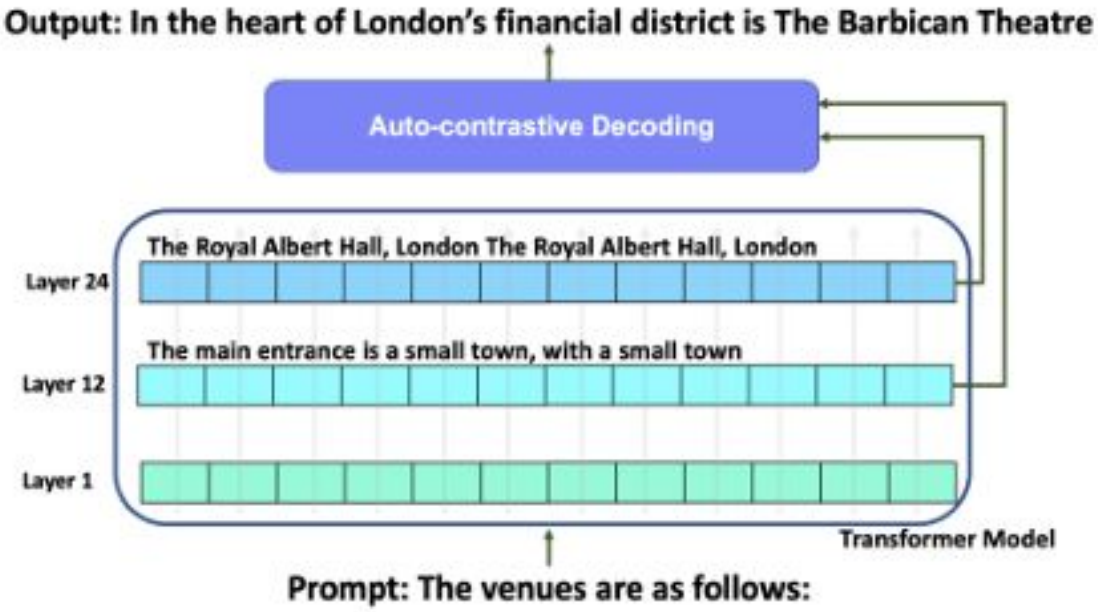
Team Members : Shyam Sundar, Suriya Prakash, Roshan Varghese, Som Sagar

Introduction

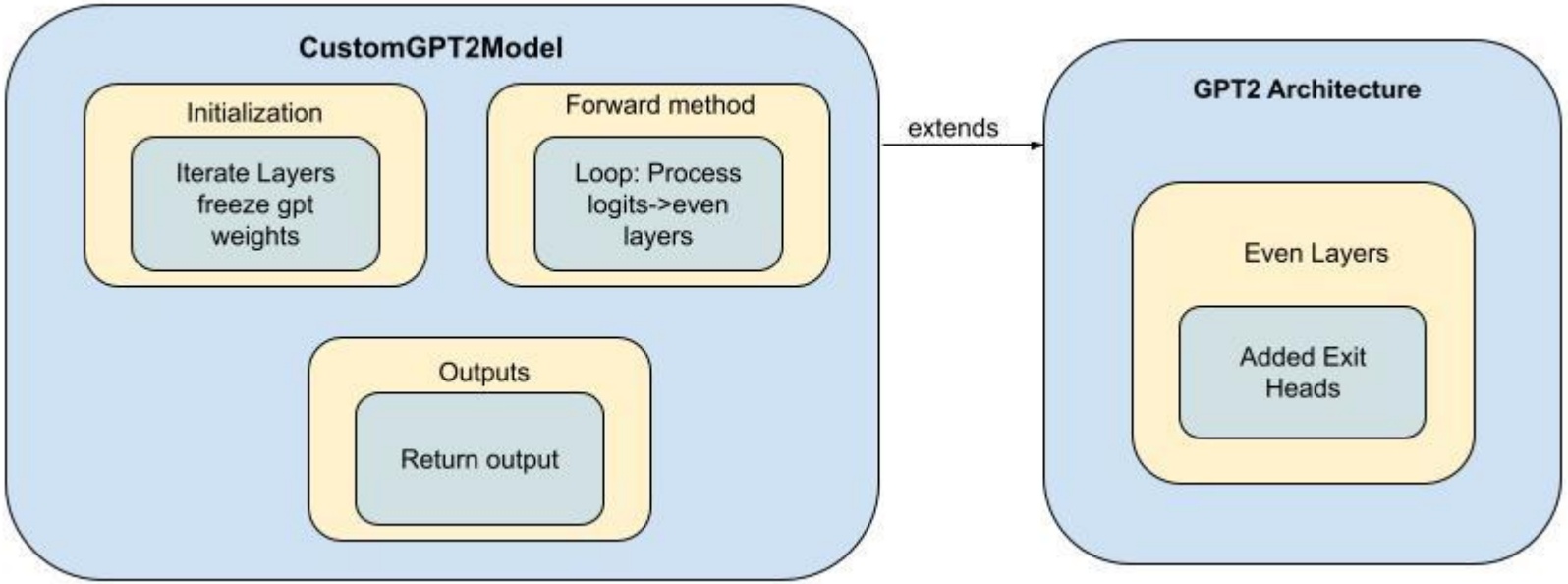
Utilizing language models for natural language processing tasks conventionally focuses on the representations in the final model layer, assuming that intermediate hidden layer representations are less informative. However, in this study, we contend that the progressive enhancement across model layers suggests that valuable insights can be derived from contrasting higher and lower layers during the inference process.

Auto Contrastive Decoding

Here the top layer(24) is taken as the expert and contrasted with layer 12, the amateur. As decoding is done token by token, we can only see the direct effect on the first token, where ACD leads to selecting an alternative high probability token



Augmentation



- **Linear layers appended to even indexed layers of the GPT-2 transformer**
- **Frooze GPT2 weights while training with cc100 dataset**

Dataset

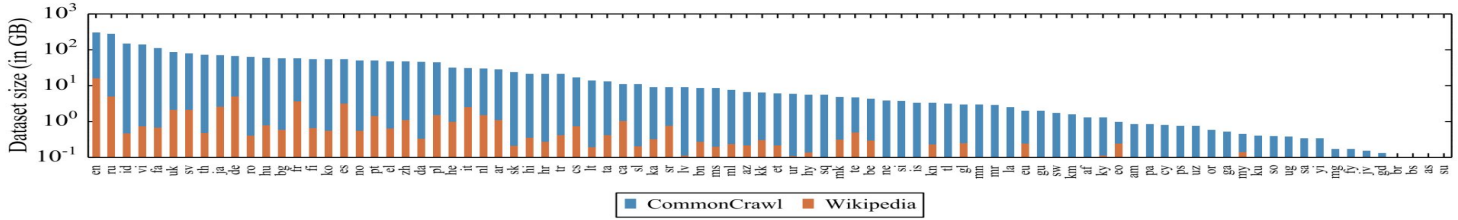


Figure 1: Amount of data in GiB (log-scale) for the 88 languages that appear in both the Wiki-100 corpus used for mBERT and XLM-100, and the CC-100 used for XLM-R. CC-100 increases the amount of data by several orders of magnitude, in particular for low-resource languages.

Dataset used : CC-100 -en

The Common Crawl (CC) dataset is a vast repository of web crawl data containing diverse text from internet sources. Within this corpus, the CC-100 subset comprises data from 100 languages, serving as a multilingual resource for natural language processing (NLP) tasks and model training. We trained the exit heads on the the english subset.

Training

Base model : GPT2-medium (with linear layers added in between)
Parameters : AdamW optimizer, learning rate 2e-4, batch size 64

The linear heads each model were trained for 3 epochs over the chunked texts from the dataset due to limited memory.

When training the heads we do not precisely replicate the original pre-training regime; specifically, we use different pre-training data and train for a smaller number of training step.

Training

For each exit head in the model, it calculates the loss using CrossEntropyLoss between the model's output and the target IDs derived from the example ID. The total loss is then averaged over each exit head.

LAMBADA Benchmark

(1) *Context:* "Yes, I thought I was going to lose the baby." "I was scared too," he stated, sincerity flooding his eyes. "You were ?" "Yes, of course. Why do you even ask?" "This baby wasn't exactly planned for."
Target sentence: "Do you honestly think that I would want you to have a _____ ?"
Target word: miscarriage

(2) *Context:* "Why?" "I would have thought you'd find him rather dry," she said. "I don't know about that," said Gabriel. "He was a great craftsman," said Heather. "That he was," said Flannery.
Target sentence: "And Polish, to boot," said _____.
Target word: Gabriel

(3) *Context:* Preston had been the last person to wear those chains, and I knew what I'd see and feel if they were slipped onto my skin-the Reaper's unending hatred of me. I'd felt enough of that emotion already in the amphitheater. I didn't want to feel anymore. "Don't put those on me," I whispered. "Please."
Target sentence: Sergei looked at me, surprised by my low, raspy please, but he put down the _____.
Target word: chains

(4) *Context:* They tuned, discussed for a moment, then struck up a lively jig. Everyone joined in, turning the courtyard into an even more chaotic scene, people now dancing in circles, swinging and spinning in circles, everyone making up their own dance steps. I felt my feet tapping, my body wanting to move.
Target sentence: Aside from writing, I 've always loved _____.
Target word: dancing

(5) *Context:* He shook his head, took a step back and held his hands up as he tried to smile without losing a cigarette. "Yes you can," Julia said in a reassuring voice. "I 've already focused on my friend. You just have to click the shutter, on top, here."
Target sentence: He nodded sheepishly, through his cigarette away and took the _____.
Target word: camera

(6) *Context:* In my palm is a clear stone, and inside it is a small ivory statuette. A guardian angel. "Figured if you're going to be out at night getting hit by cars, you might as well have some backup." I look at him, feeling stunned. Like this is some sort of sign.
Target sentence: But as I stare at Harlin, his mouth curved in a confident grin, I don't care about _____.
Target word: signs

- The prompts in the dataset are primarily used to check the models prediction power of the target word
- A lot of challenges involve in these prompts. Coreference resolution, long contextual dependencies to name a few
- This served as a good benchmark for us to go further with the evaluation of GPT2 on the Inverse Scaling dataset

Model	Accuracy	Perplexity
GPT-Medium+ ACD	0.55	15.4

Table 3: Text generation results on LAMBADA.

Contrastive logit calculation

1. Softmax Calculation:

$\mathbf{p} = \text{softmax}(\mathbf{u})$

$\mathbf{q} = \text{softmax}(\mathbf{v})$

U - lower layer logit
V - Upper layer logit

2. Probability Thresholding:

$\text{plausible\_token\_probability\_threshold} = \max(\mathbf{p}) \cdot \alpha$

$\text{min\_threshold} = \min(\text{plausible\_token\_probability\_threshold}, \text{Top-}k \text{ probability in } \mathbf{p})$

3. Contrastive Operation:

$$\text{contrasted\_logits}_{ij} = \begin{cases} \log(\mathbf{p}_{ij}) - \log(\mathbf{q}_{ij}), & \text{if } \mathbf{p}_{ij} \geq \text{min\_threshold} \\ \mathbf{p}_{ij}, & \text{otherwise} \end{cases}$$

4. Softmax for Selected Tokens:

$\text{softmax\_for\_included\_new} = \text{softmax}(\text{contrasted\_logits})$

5. Probability Redistribution:

$$\text{adjusted\_contrasted\_logits}_{ij} = \text{softmax\_for\_included\_new}_{ij} \cdot \sum_k \mathbf{p}_{ik}$$

6. Final Logarithmic Transformation:

$\text{contrasted\_logits}_{ij} = \log(\text{adjusted\_contrasted\_logits}_{ij})$

Let's consider a sentiment analysis example with three classes: "Positive," "Neutral," and "Negative."

1. Softmax Calculation:

$\mathbf{u} = [0.6 \quad 0.3 \quad 0.1]$

$\mathbf{v} = [0.2 \quad 0.5 \quad 0.3]$

$\mathbf{p} = [0.7 \quad 0.2 \quad 0.1]$

$\mathbf{q} = [0.2 \quad 0.5 \quad 0.3]$

2. Probability Thresholding:

$\text{min\_threshold} = 0.25$

3. Contrastive Operation:

$\text{contrasted\_logits} = [\log(0.6) - \log(0.2) \quad \log(0.3) - \log(0.5) \quad 0.1]$

4. Softmax for Selected Tokens:

$\text{softmax\_for\_included\_new} = [0.7 \quad 0.2 \quad 0.1]$

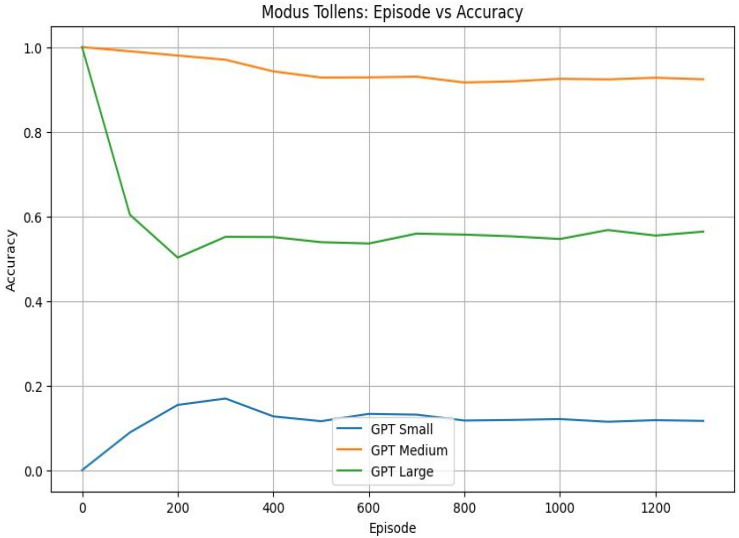
5. Probability Redistribution:

$\text{adjusted\_contrasted\_logits} = [0.7 \quad 0.2 \quad 0.1]$

6. Final Logarithmic Transformation:

$\text{final\_logits} = [-0.2 \quad -1.6 \quad -2.3]$

Results



- The way we evaluate is by taking the probabilities of all tokens and get probabilities for "Yes" and "No" and compare those to yield a prediction
- ACD seems to have improved the prediction power of GPT-small for these types of prompts
- GPT-Medium + ACD has almost the same performance as GPT-Medium
- Logical reasoning on smaller models with ACD is worth exploring

Layer	Accuracy
6	0.997
12	0.998
14	0.998

Table 3: Modus tollens results by contrasting GPT2-Medium with layer 24.

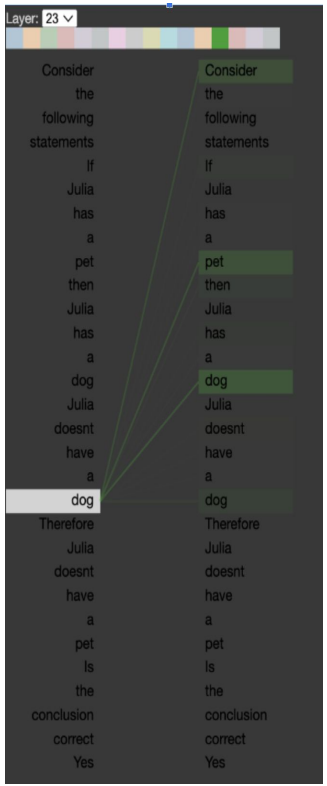
Layer	Accuracy
4	0.5161
6	0.4919
8	0.5056

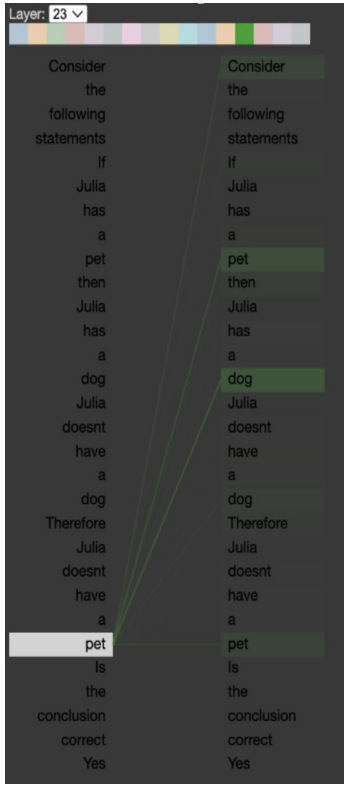
Table 3: Modus tollens results by contrasting GPT2-Small with layer 12.

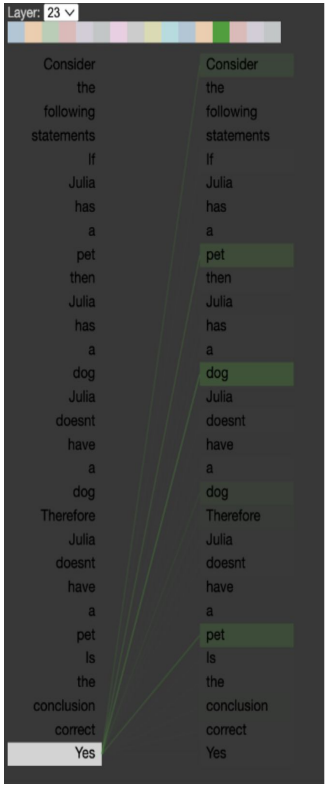
Attention weights of GPT small

- The GPT-small model struggles to find associations between 'dog' and 'pet' which would help infer the correct answer. Almost all of the 12 heads struggle.
- The conclusion 'Yes' should have higher weight w.r.t the words 'dog' and 'pet' while also taking into account words like 'doesn't' and 'then'
- Most of the attention head struggle to find these associations
- GPT-small + ACD was able to get these associations right

Attention weights of GPT medium

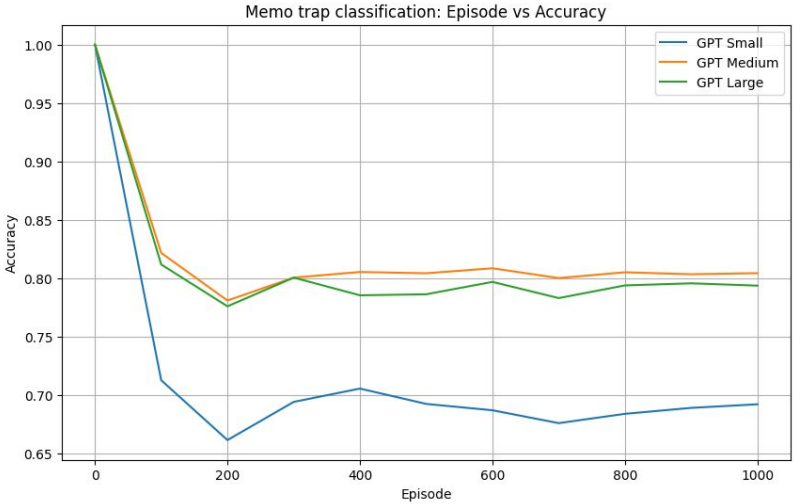






- Whereas, the gpt-medium does strongly associate the words 'dog' and 'pet'. The conclusion 'Yes' also has higher weight w.r.t the words 'dog' and 'pet' while also taking into account words like 'doesn't' and 'then'.
- Moreover, most of the attention heads find some of these associations as well as opposed to gpt-small.
- ACD might have enhanced these associations while rebalancing the logit scores of these particular words.

Memo Trap results



- The way we evaluate is by taking the probabilities of all tokens and get probabilities of the prompts with 2 choices (eg: [’thoughts’, ’words’]) and compare those to yield a prediction
- ACD seems to worsen the prediction power on these types of prompts
- Both GPT-small + ACD and GPT-Medium + ACD hamper the performance of the non augmented models
- Not resorting to parametric knowledge still remains a challenge

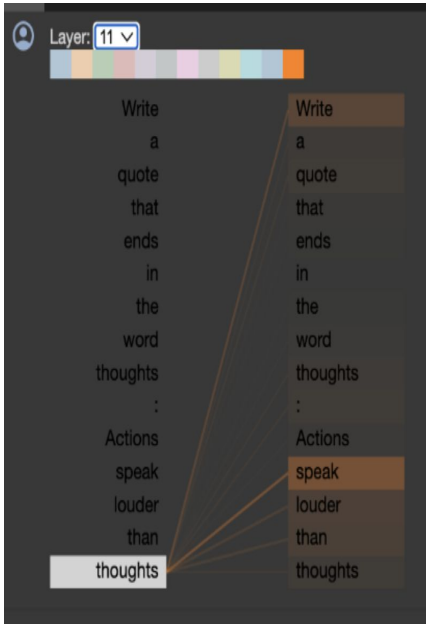
Layer	Accuracy
6	0.324
12	0.337
14	0.348
18	0.310

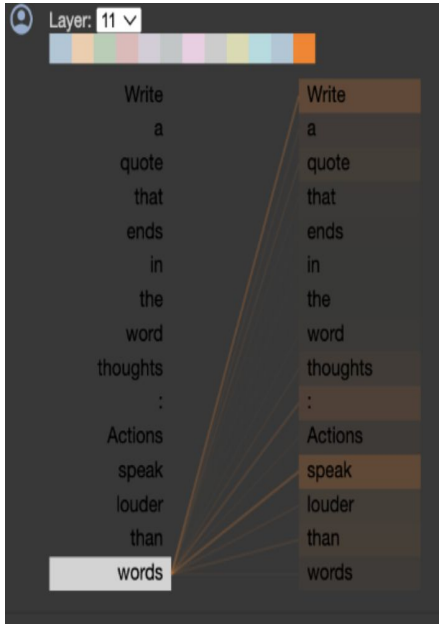
Table 1: Memo Trap results by contrasting with layer 24.

Layer	Accuracy
4	0.5126
6	0.5096
8	0.5138

Table 1: Memo Trap results by contrasting with layer 12 of GPT-Small.

Attention weights of GPT small





- We would want ‘thoughts’ to have higher weight than ‘words’ which GPT-small get wrong
- One possible reason could be , since the logits that are being contrasted carry different levels of abstraction and information, it is possible that the rebalancing of the scores does more damage than otherwise.
- Also, originally contrastive decoding techniques seems to do a better job of reducing repetition and increase coherence and diversity of the generated text. It is possible that in the process there might have occured an increased reliance on parametric knowledge which hampers the response

Future Scope

- ACD in essence re-balances the logit scores of the model to act as a watchdog for certain tasks.
- Memorization trap problem still persists. This calls into question the notion that sometimes intermediate layers can track a vector of improvement and that we can extend this vector to upper layers.
- But Logical reasoning on smaller models is worth exploring.
- Attention heads of smaller models need to have a better aggregation strategy

Reference

1. Anonymous (2023). Contrastive decoding improves reasoning in large language models. In Submitted to The Twelfth International Conference on Learning Representations. under review.
2. Gera, A., Friedman, R., Arviv, O., Gunasekara, C., Sznajder, B., Slonim, N., and Shnarch, E. (2023). The benefits of bad advice: Autocontrastive decoding across model layers. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10406–10420, Toronto, Canada. Association for Computational Linguistics.
3. Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., and Lewis, M. (2022). Contrastive decoding: Open-ended text generation as optimization. arXiv preprint arXiv:2210.15097.