ICLR
The International Conference on Learning Representations (ICLR)
2024

Unveiling and Manipulating Prompt Influence in Large Language Models

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, Junlang Qian, Kezhi Mao

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE
(FRS) FUTURE RESILIENT SYSTEMS
(SEC) SINGAPORE-ETH CENTRE
CREATE
Campus for Research Excellence And Technological Enterprise

Prompt Influence

The formulation of prompts significantly shapes the textual responses of large language models (LLMs).

Comprehending the influence of individual input tokens in prompts, i.e., input saliency, can augment our insight into LLM interpretability and foster the development of refined prompting strategies to modulate LLM outputs.

Prompt Influence

Input-Output Example

Input: Tracy praises those lucky ___

LLM output: guys

Explanation Example

Target token: guys

Alternative token: any other token in the vocabulary, such as guy.

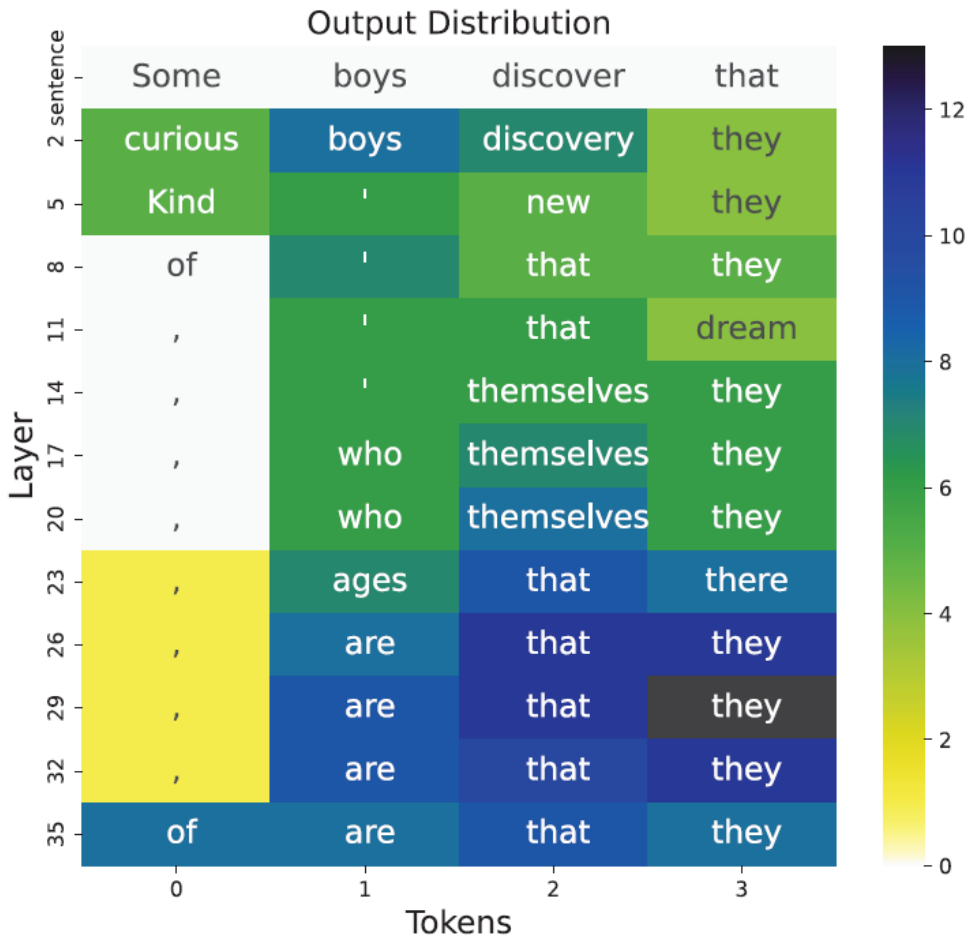
Contrastive explanation: Why the LLM generates “guys” instead of “guy”?

Token Distributions

Definition:

Token representations at every layer can be projected as evolving distributions over the vocabulary through the LM head.

Operation: Utilize the LM head to project the token hidden states at every layer into the embedding space.

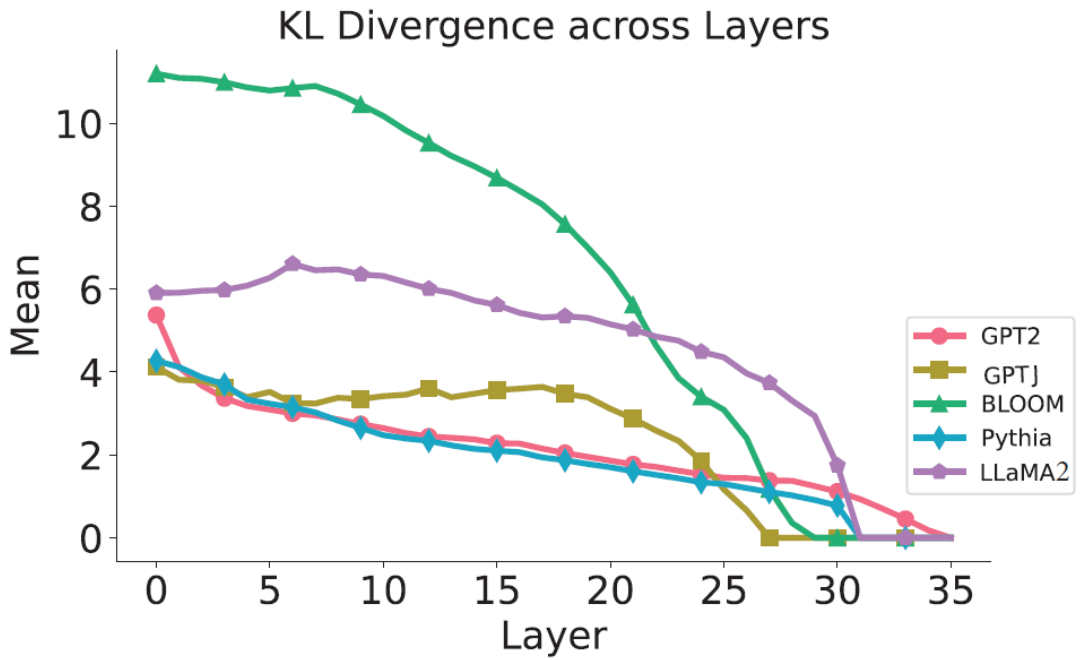


(a) GPT2-large

Token Distributions

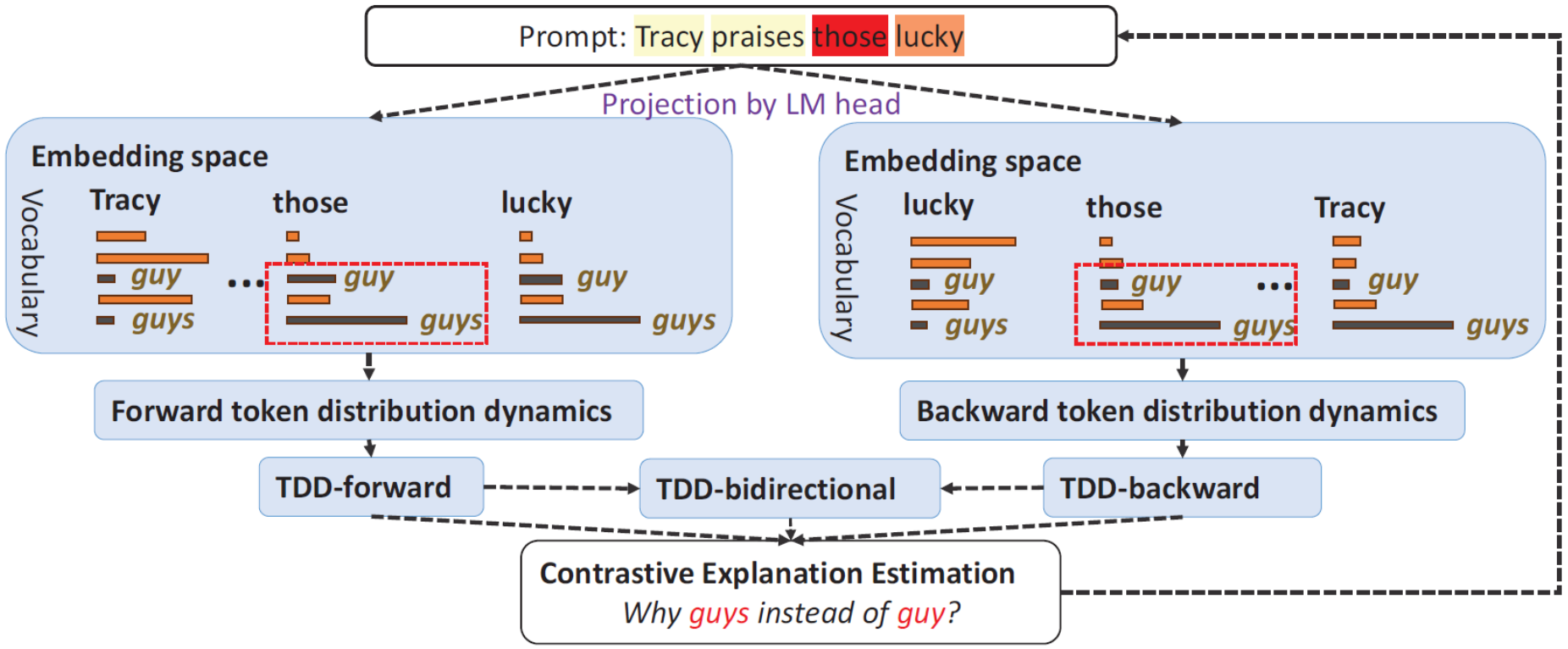
Properties:

- **Interpretability**: Each dimension signifies a specific token in the vocabulary and its logit value indicates its likelihood.
- **Convergence**: Intermediate layer token distributions monotonically converge towards the final layer's distribution.



Token Distribution Dynamics as Input Saliency

- The distribution shifts of each token are due to the introduction of input tokens.
- We can deduce a token's significance by observing how it affects token distribution changes.



TDD-Forward

TDD-forward capitalizes on token distribution dynamics throughout the forward progression of learning and prediction.

Most LLMs employ decoder-only structures, in which the $i-th$ token representation indicates the probability of any token in the vocabulary becoming the (i+1)$-th$ token.

Hence, given the first i tokens, the LLM's confidence to produce the target token w_t over the alternative token w_a can be computed as follows:

$$r_i = p_i(w_t) - p_i(w_a)$$

r_i represents the confidence of LLM generating w_t over w_a based on the first i tokens.

TDD-Forward

The transition from r_{i-1} to r_i can be roughly attributed to the introduction of i-th token w_i. Thus, the saliency of w_i can be approximated as:

$$c_i^{forward} = \begin{cases} r_i, & i = 1 \\ r_i - r_{i-1}, & i > 1 \end{cases}$$

TDD-Backward

For any given input, the process begins with the last token, progressively incorporating preceding tokens to evaluate the probability distribution of the ultimate prediction

$$r_i = p_{i,n}(w_t) - p_{i,n}(w_a)$$

$p_{i,n}$ is determined by projecting the final token based on input tokens from w_i to w_n and quantifies the probability of the $(n+1) - th$ token.

The transition from r_{i+1} to r_i can be roughly attributed to the introduction of i -th token w_i . Thus, the saliency of w_i can be approximated as:

$$c_i^{backward} = \begin{cases} r_i, & i = n \\ r_i - r_{i+1}, & i < n \end{cases}$$

TDD-Bidirectional

Drawing inspiration from the efficacy of bidirectional neural networks, which assimilate information from both directions, we also propose TDD-bidirectional.

$$c_i^{bidirectional} = c_i^{forward} + c_i^{backward}$$

Experiments

Dataset: 11 same datasets from BLiMP containing various linguistic phenomena across syntax, morphology, and semantics.

LLMs: GPT2-large, GPTJ-6B, BLOOM-7B, Pythia-6.9B, LLaMA2-7B

When presenting a prompt to LLMs, the goal is to ascertain the input saliency that prompts the LLM to produce the target token over the alternative.

Phenomenon	Prompt	Target token	Alternative token
Determiner-Noun Agreement (Morphology)	Joel complains about those ___	drivers	driver
Argument Structure (Sytax)	Amanda was respected by some ___	waitresses	picture
NPI Licensing (Semantics)	Even many birds can ___	really	ever

Experiments

Evaluation Metrics – Perturbation method. We replace K% of tokens, deemed most/least significant, with a meaningless space token to gauge its influence on LLMs' output. We quantify explanation faithfulness using two metrics: AOPC and Sufficiency.

AOPC: Initially, all input tokens are substituted with the meaningless space token. Tokens are then sequentially reintroduced (at 20% intervals), ranked from most to least significant. We compute the relative probability (AOPC) of the target token compared to the alternative token, determined by the softmax of their respective logits. A higher AOPC indicates a more precise explanation.

Sufficiency: Initially, all input tokens are retained. Tokens are subsequently removed, starting from the most to the least significant. We then report the relative probability (sufficiency) of the target token versus the alternative token. A lower sufficiency score signifies a more accurate explanation.

Experiments

LLMs	GPT2		GPTJ		BLOOM		Pythia		LLaMA2	
	AOPC ↑	Suff ↓	AOPC ↑	Suff ↓	AOPC ↑	Suff ↓	AOPC ↑	Suff ↓	AOPC ↑	Suff ↓
Rollout	64.13	62.14	64.48	62.03	65.56	61.73	64.18	62.10	59.09	60.10
IG	63.72	61.03	64.02	62.11	62.80	62.33	63.29	60.88	59.95	57.11
Con-GN	63.69	61.12	63.73	61.89	63.86	61.43	63.76	60.84	59.79	57.78
Con-GI	64.71	60.53	64.89	60.32	65.18	60.84	64.59	59.94	59.73	57.19
TDD-forward	67.28	57.60	68.71	55.11	67.75	57.13	67.10	56.81	62.80	52.69
TDD-backward	**70.46**	**54.20**	70.61	54.08	70.20	**55.07**	**71.29**	**52.67**	63.71	53.22
TDD-bidirectional	69.95	55.22	**71.05**	**53.31**	**70.22**	55.39	70.58	53.38	**65.51**	**52.04**

Applications

Beyond merely providing interpretation, we elucidate how to harness TDD to manipulate prompts and control LLM outputs. We spotlight two key applications:

- Zero-shot toxic language suppression;
- Zero-shot sentiment steering.

In toxic language suppression, TDD identifies and neutralizes toxic triggers in prompts before they are fed into LLMs.

For sentiment modulation, TDD captures sentiment cues in prompts, adjusting their sentiment inclination to guide the sentiment of generated texts.

Zero-shot Toxic Language Suppression

Method	Toxicity↓	Severe Toxicity↓	Sexually explicit↓	Threat↓	Profanity↓	Identify attack↓	Fluency↓	Dist-1↑	Dist-2↑	Dist-3↑
GPT2	0.49	0.18	0.25	0.09	0.39	0.09	**23.06**	**0.83**	**0.78**	0.71
SP	0.47	0.14	0.24	0.05	0.36	0.09	23.23	0.83	0.78	0.71
ASP	0.41	0.12	0.18	0.05	0.31	0.08	23.84	0.83	0.78	0.71
WORDFILTER	0.36	0.11	0.14	0.09	0.25	0.07	24.39	0.83	0.78	0.71
FFNControl	0.26	0.09	0.15	**0.03**	0.22	0.05	27.24	0.83	0.78	0.71
Con-GI	0.26	0.09	0.13	0.05	0.21	0.05	23.30	0.79	0.75	0.70
Con-GN	0.24	0.08	0.11	0.04	0.19	0.05	23.35	0.80	0.77	**0.72**
TDD	**0.20**	**0.07**	**0.09**	0.04	**0.16**	**0.04**	23.10	0.81	0.77	0.71

Zero-shot Sentiment Steering

	Neutral → Negative					Neutral → Positive				
Method	Negative percent↑	Fluency↓	Dist-1↑	Dist-2↑	Dist-3 ↑	Positive percent↑	Fluency ↓	Dist-1↑	Dist-2↑	Dist-3 ↑
GPT2	0.48	**24.82**	**0.84**	**0.83**	**0.78**	0.52	**24.82**	**0.84**	**0.83**	**0.78**
SP	0.51	25.01	0.84	0.83	0.78	0.55	25.19	0.84	0.83	0.78
ASP	0.53	24.96	0.84	0.83	0.78	0.56	25.04	0.84	0.83	0.78
WordFILTER	0.52	25.26	0.84	0.83	0.78	0.57	25.07	0.84	0.83	0.78
FFNControl	0.82	24.84	0.84	0.83	0.78	0.68	28.45	0.84	0.83	0.78
Con-GI	0.85	25.12	0.81	0.80	0.76	0.75	25.86	0.82	0.81	0.77
Con-GN	0.84	25.02	0.81	0.80	0.76	0.70	25.94	0.83	0.83	0.78
TDD	**0.87**	25.55	0.82	0.81	0.76	**0.78**	26.27	0.82	0.82	0.76

Conclusions

- We introduce a novel and efficient TDD framework to unveil and manipulate prompt influence in LLMs. This approach harnesses distribution dynamics to gauge token significance. Comprehensive tests reveal that TDD outperforms existing baselines in elucidating prompts' effects on LLM outputs.

- Furthermore, we highlight two practical applications of TDD: zero-shot toxic language mitigation and sentiment direction. By precisely pinpointing toxic or sentiment indicators in prompts, TDD can adeptly steer LLMs to produce desired outputs.