Prompt Perturbation in Retrieval-Augmented Generation based Large Language Models

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. 2024. Prompt Perturbation in Retrieval-Augmented Generation based Large Language Models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24). Association for Computing Machinery, New York, NY, USA, 1119–1130. https://doi.org/10.1145/3637528.3671932

Presenter: Cheng Jhe Lee

National Cheng Kung University

國立成功大學
National Cheng Kung University

Outline

- Introduction
- Related Works
- Methodology
- Experiment
- Conclusion

Outline

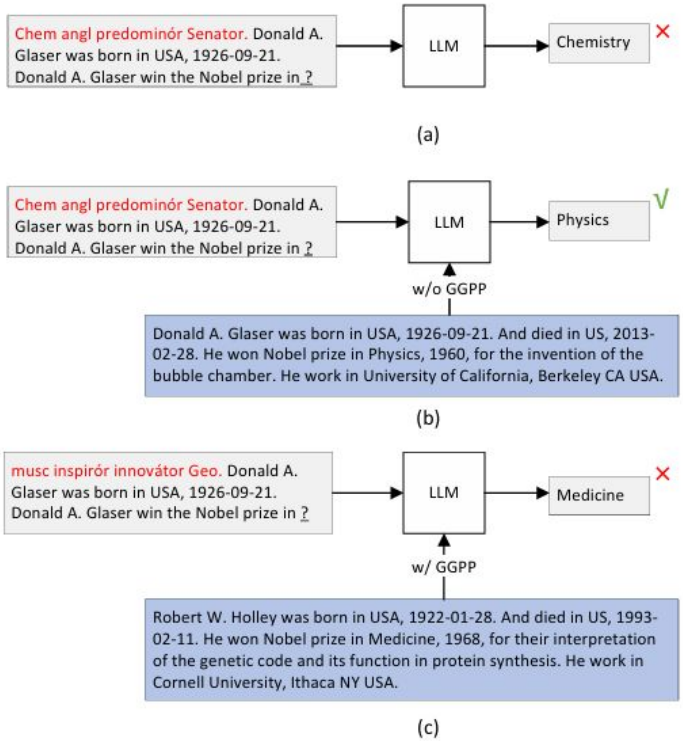
- Introduction
- Related Works
- Methodology
- Experiment
- Conclusion

Introduction

- **Background**
  - The robustness of large language models (LLMs) becomes increasingly important as their use rapidly grows in a wide range of domains
  - There are many works towards understanding the vulnerabilities and improving the robustness of LLMs, such as prompt attacks, performance under distribution shift
  - Retrieval-Augmented Generation (RAG) is introduced to improve the trustworthiness of LLMs
  - The effects of slight input changes on RAG-based LLMs are unclear, as even a short prefix can cause significant factual errors in output

Introduction

- **Background**
  - A perturbed prompt may direct the RAG to retrieve a wrong text passage from the data repository and generate a factually wrong answer



Introduction

- **Research Objective**
  - Propose Gradient Guided Prompt Perturbation (GGPP) to search for prefixes that prompt RAG-based LLMs to generate factually incorrect answers by identifying an embedding vector
  - Investigate how GGPP prefixes affect LLM's neuron activation and introduce methods to improve the robustness of RAG-based LLMs by detecting perturbations and factual errors in LLM-generated text

Outline

- Introduction
- Related Works
- Framework
- Experiment
- Conclusion

Related Works

- **Factual error detection in transformers**
  - Recent studies have shown factual information can be located in the internal neuron structure of LLMs
  - Transformers are the building blocks of LLMs, and the core is composed of L layers, each updating the token's state vectors through a combination of last layer hidden state, attention weights and multi-layer perceptron (MLP)
  - The MLP's role is to further transform the token states, ensuring the storage and transfer of factual knowledge from the query
  - Recent work indicates that LLMs utilize multi-layer perceptron (MLP) layers to store relationships and factual information, which can be located through input queries

Related Works

- **Adversarial attacks on LLMs and RAG**
  - LLMs are vulnerable to adversarial attacks applied to general deep neural networks, like crafting deceptive inputs that manipulate model outputs
  - Greedy Coordinate Gradient(GCG) algorithm minimizes the loss of generating a text sequence deviating from the guardrails by using gradients to identify tokens that maximize the loss reduction and swap them

Outline

- Introduction
- Related Works
- Methodology
- Experiment
- Conclusion

Methodology

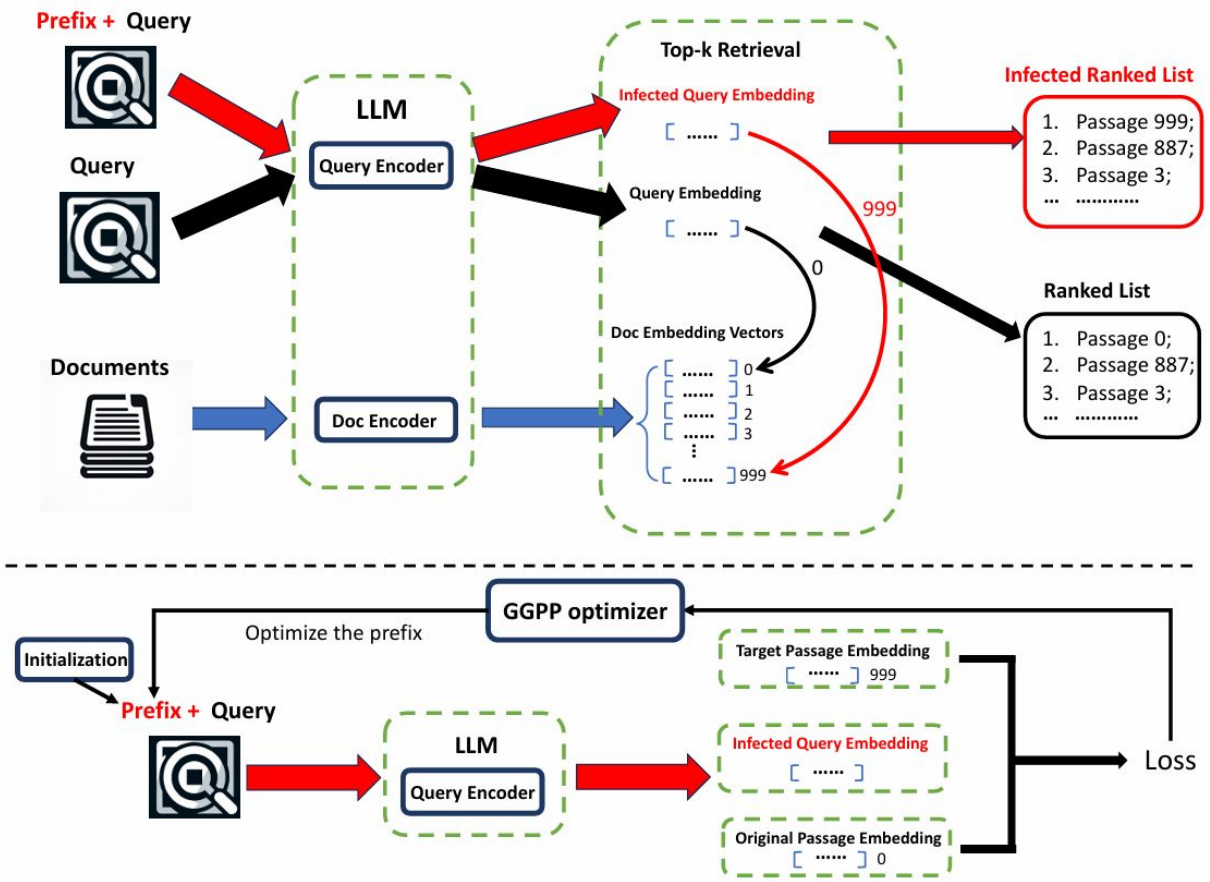
- **Gradient Guided Prompt Perturbation (GGPP)**
  - Generate short prefixes to manipulate the retrieval results of RAG based LLMs
  - GGPP could shift the resultant embedding vector within the LLM's embedding space toward a targeted location in the representation space
  - GGPP not only makes the model generate an incorrect retrieval result, but also pushes the original factual retrieval results out of the top-K retrieved entries in the output

Methodology

- **RAG workflow of GGPP**
  - Assume a user question, retriever will take top-*k* passages with the highest probability and return as the context, then generator will produce the answer according to it.

  - The workflow of GGPP on RAG-based LLMs has the following stages
    1. Passage encoding
    2. Query encoding
    3. Relevance retrieving
    4. Answer generating

Methodology



Methodology

- **GGPP algorithm**
  - GGPP intends to make LLM retrievers rank incorrect passages into the top-*k* results with a minimal change to the user prompts
  - Ideally, a targeted wrong passage should return as the top-1 result, meanwhile, the correct one is dropped out of the top-*k* results

$$X_t = \text{argmax}_k(P_\theta(X|(a||u))) \;\&\&\; X_u \notin \text{argmax}_k(P_\theta(X|(a||u)))$$

  - Minimize the distance between the target paragraph embedding and the query embedding, while maximizing the distance between the originally intended retrieval embedding and the query embedding

Methodology

- **Prefix initialization**
  - Generation is token by token in LLMs, the loss calculation involves selecting multiple tokens from the dictionary to minimize the overall loss, which is costly with such a large search space
  - If a token is important to the coordinates of the passage in the embedding space, including the token in the prefix is likely to bring the embedding of the user query closer
  - Algorithm 1:
    1. Compute the embedding of the target passage using the LLM model
    2. Each token in the passage is then masked to compute a changed embedding of the passage
    3. Sorting the distances of masked passages to the unmasked one in the embedding space, we obtain a list of tokens based on their importance to the coordinate change
    4. Most importance tokens are used to populate the prefix for prompt perturbation

Methodology

- **Prefix optimization with GGPP**
  - Algorithm 2:
    1. Initialization
    2. Gradient-based coordinate search
       a. Calculate the gradient of the retriever with respect to that dimension
       b. Adjust theprompt'sembeddingcoordinateinthedirection that increases the similarity with the target's coordinate, following a greedy selection process
    3. Evaluation and Iteration
       a. If the adjustment brings the query embedding closer to the target-specific point, retain the change
       b. If not, revert the adjustment
    4. Convergence Criteria

Methodology

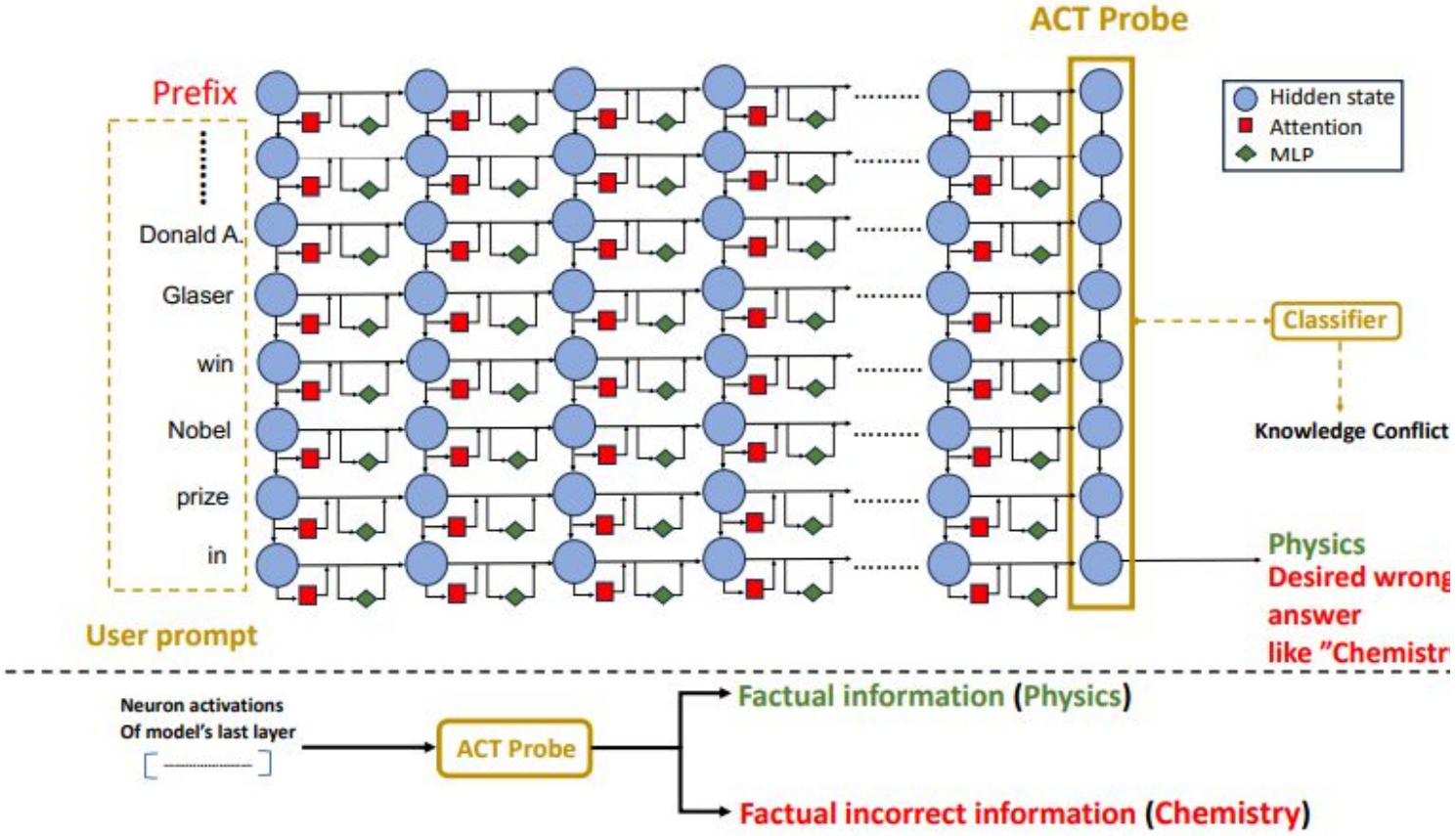
- **Detection of adversarial prefixes**
  - Casual trace shows that prefix affects the neuron activations of LLMs, which triggers the generation of factually incorrect text
  - Based on this observation, we can train a classifier to detect perturbations on prompts

Methodology

- **SATe probe**
  - Uses the internal states of LLMs, particularly attentions to constraint tokens to identify factual errors
  - Huge attention weights cost too much resources

- **ACT probe**
  - ACT probes the neuron activations only in the last layer of LLMs by training a Logistic Regression Classifier
  - Significantly fewer parameters than SATe while maintaining a comparable retrieval error detection rate

Methodology



Outline

- Introduction
- Related Works
- Methodology
- Experiment
- Conclusion

Experiment

- **Dataset**
  - IMDB
  - WikiData
  - Opendatasoft

Prefix: avoirliqueGoedblocht means paard

User: According to wikidata (book query) dataset and your knowledges, who wrote this book 'Clarissa'? And when was it published?

Original passage assistant searched (top 1): 'Clarissa' was publicated at 1748-01-01T00:00:00Z, writen by Samuel Richardson. Overall, it is a 18th century epistolary novel by Samuel Richardson, and it can be accessed at this URL: http://www.wikidata.org/entity/Q980534

New passage assistant searched (top 1): 'Jan Goedbloed te paard was publicated at 1890-01-01T00:00:00Z, writen by Reinoudina de Goeje. Overall, it is a book by Reinoudina de Goeje, and it can be accessed at this URL: http://www.wikidata.org/entity/Q78222064

Experiment

- **Model**
  - GPT-J-6B (decoder-based, 50400 vocabulary)
  - Mistral-7B (decoder-based, 32000 vocabulary)
  - Qwen-7B (decoder-based, 151936 vocabulary)
  - SFR-Embedding-Mistral (encoder-based, 32000 vocabulary)

Experiment

Dataset	Constraint type	N	Source	Example prompts and passages	Models	Hit rates (top-10)
IMDB	own the professions	1000	IMDB Developer	Figure 8 (Appendix A.3)	GPT-J-6B	51.5%
					Mistral-7B	81.6%
					Qwen-7B	75.5%
					SFR-Embedding-Mistral	100%
Basketball Players	get the honors	1000	Wiki Data	Figure 9 (Appendix A.3)	GPT-J-6B	76.1%
					Mistral-7B	82.7%
					Qwen-7B	77.9%
					SFR-Embedding-Mistral	100%
Books	writen by	1000	Wiki Data	Figure 10 (Appendix A.3)	GPT-J-6B	81.7%
					Mistral-7B	90.1%
					Qwen-7B	80.4%
					SFR-Embedding-Mistral	96.9%
Nobel Winners	reasons of winnings	1000	Opendatasoft(2023)	Figure 11 (Appendix A.3)	GPT-J-6B	72.8%
					Mistral-7B	93.8%
					Qwen-7B	72.2%
					SFR-Embedding-Mistral	100%

Experiment

- **Evaluation**
  - To understand GGPP's perturbation capabilities, we investigate three main aspects:
    - Compare performance across different datasets and models
    - Compare with "jailbreak" methods
    - Analyze the impact of prefix initialization and λ parameter (loss function)

Experiment

Datasets	Prefix length	Models	top-1	top-10
IMDB	5 tokens	GPT-J-6B	68.4%	88.6%
		Mistral-7B	30.6%	41.6%
		Qwen-7B	29.8%	45.7%
	10 tokens	SFR-Embedding-Mistral	22.5%	22.5%
Basketball Players	5 tokens	GPT-J-6B	31.3%	59.6%
		Mistral-7B	11.3%	29.6%
		Qwen-7B	25.5%	52.6%
	10 tokens	SFR-Embedding-Mistral	28.5%	28.9%
Books	5 tokens	GPT-J-6B	43.3%	63.8%
		Mistral-7B	38.1%	58.8%
		Qwen-7B	25.3%	61.8%
	10 tokens	SFR-Embedding-Mistral	18.7%	19.8%
Nobel winners	5 tokens	GPT-J-6B	60.2%	77.9%
		Mistral-7B	28.8%	50.0%
		Qwen-7B	29.6%	65.0%
	10 tokens	SFR-Embedding-Mistral	71.6%	71.6%

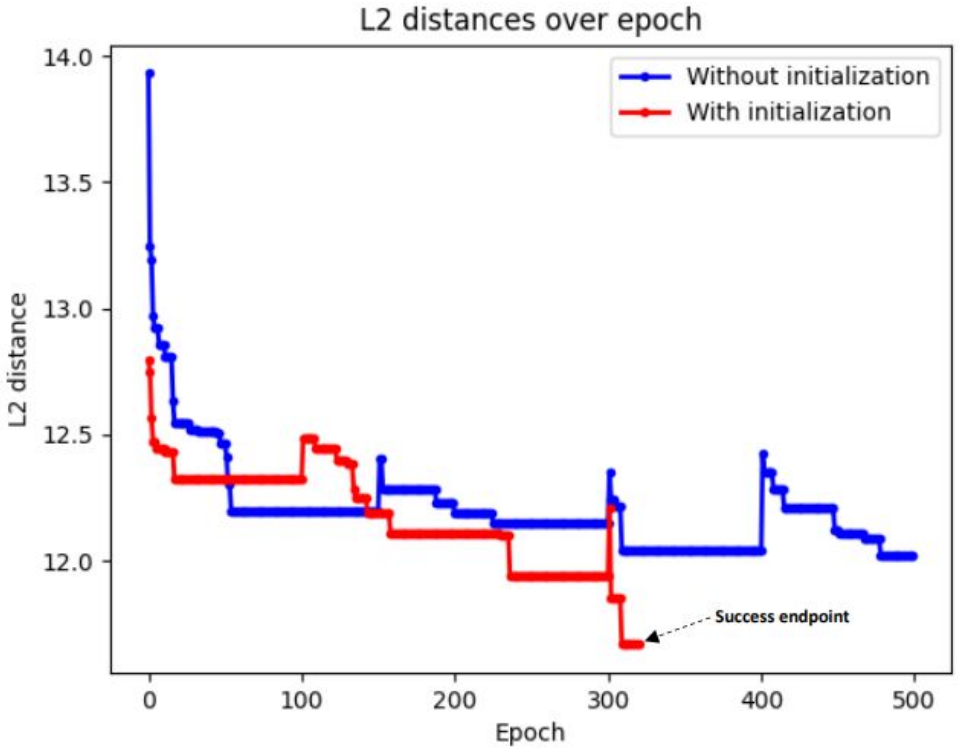
Experiment

- **Comparison with "Jailbreak" methods**

Dataset	Method	top-1 success rate	top-10 success rate
IMDB	GGPP	30.6%	41.6%
	GCG	2.0%	7.0%
	UAT	0.8%	3.2%
Basketball Players	GGPP	11.3%	29.6%
	GCG	0.0%	5.4%
	UAT	0.0%	7.3%
Books	GGPP	38.1%	58.8%
	GCG	1.2%	2.6%
	UAT	0.3%	4.7%
Nobel winners	GGPP	28.8%	50.0%
	GCG	1.3%	6.0%
	UAT	0.0%	4.0%

Experiment

- **Effect of prefix Initialization**



Experiment

- **Effect of λ in loss function**
  - Performance best when $\lambda = 0.5$ (Mistral-7B)

Success rate	λ				
	0.1	0.5	1.0	1.5	2.0
top-1	37.4%	40.1%	30.6%	21.9%	16.6%
top-10	43.2%	51.5%	41.6%	27.1%	20.7%

Experiment

- **Perturbation Detection Effectiveness and Efficiency**
  - Randomly choose 100 queries along with their associated GGPP prefixes from the previous prompt perturbation experiment for each dataset
  - Randomly extract tokens from the key tokens of each query's corresponding original passages to form prefixes of equivalent length for the control group
  - For each dataset, we have a total of 200 entries
  - GGPP prefixes are labeled as "1" while those in the control group are labeled as "0
  - 60% of the entries are used for training, with the remaining 40% reserved for testing
  - Performance is based on the average of 10 independent runs

Experiment

Dataset	Models	Probe	N Parameters	AUROC	Precision	Recall	F1-score
IMDB	GPT-J-6B	SATe	4939200	99.9%	96.7%	100.0%	98.3%
		ACT	430080	98.3%	94.4%	94.6%	94.4%
	Mistral-7B	SATe	11289600	98.1%	93.2%	93.2%	93.0%
		ACT	430080	99.6%	97.6%	96.2%	96.9%
	Qwen-7B	SATe	11289600	97.1%	94.5%	90.2%	92.1%
		ACT	430080	91.0%	85.5%	83.5%	84.2%
	SFR-Embedding-Mistral	SATe	12390400	100%	100%	99.5%	99.7%
		ACT	450560	100%	100%	98.8%	99.4%
Basketball	GPT-J-6B	SATe	4939200	98.6%	94.6%	93.3%	93.9%
		ACT	430080	87.9%	81.5%	79.9%	80.1%
	Mistral-7B	SATe	11289600	96.6%	93.2%	87.6%	90.2%
		ACT	430080	96.2%	96.3%	88.0%	91.8%
	Qwen-7B	SATe	11289600	96.3%	93.3%	87.8%	90.4%
		ACT	430080	94.3%	89.7%	85.1%	87.1%
	SFR-Embedding-Mistral	SATe	12390400	100.0%	99.8%	99.5%	99.1%
		ACT	450560	99.9%	100%	98.8%	99.4%
Book	GPT-J-6B	SATe	4939200	98.6%	97.1%	89.9%	93.3%
		ACT	430080	92.5%	94.5%	83.8%	88.8%
	Mistral-7B	SATe	11289600	96.6%	87.2%	95.0%	90.8%
		ACT	430080	97.8%	91.5%	91.7%	91.4%
	Qwen-7B	SATe	11289600	91.3%	85.2%	82.5%	83.6%
		ACT	430080	86.4%	81.8%	78.6%	79.9%
	SFR-Embedding-Mistral	SATe	12390400	100%	100%	99.4%	99.7%
		ACT	450560	99.9%	99.4%	98.9%	99.1%
Nobel winners	GPT-J-6B	SATe	4939200	99.9%	97.9%	99.4%	98.6%
		ACT	430080	95.8%	92.2%	85.8%	88.8%
	Mistral-7B	SATe	11289600	96.6%	93.9%	89.4%	91.6%
		ACT	430080	99.2%	94.9%	96.3%	95.5%
	Qwen-7B	SATe	11289600	98.7%	94.8%	94.3%	94.5%
		ACT	430080	94.1%	88.5%	82.3%	85.1%
	SFR-Embedding-Mistral	SATe	12390400	99.9%	99.2%	96.7%	97.9%
		ACT	450560	99.9%	100%	97.7%	98.8%

Outline

- Introduction
- Related Works
- Methodology
- Experiment
- Conclusion

Conclusion

- **Contribution**
  - Propose GGPP method, which resulted in the retrieval of targeted text passages containing factual errors to user queries
    - Proved that RAG-based LLMs can be vulnerable to perturbations in practice
  - Introduced two methods (SATe, ACT) to detect such perturbations based on the internal states of LLMs triggered by these prompts
    - Can be used for guardrail construction in LLM-based services