KAIST <CS453: Automated Software Testing> Final project

Knowledge Graph-guided Autoregressive Test Generation for Diversity-Based Prompt Testing

Team 3
Yohan Park (20190258)
Ihchae Ryu (20200216)
Sihyun Ahn (20200365)

Contents

1. Problem Definition

With the emergence of LLM-based applications, techniques to efficiently test LLM-applied software is becoming more and more critical. The core difficulties in LLM prompt testing includes the non-deterministic and uncontrollable outputs, time and cost expensiveness in iterative refinement, and the fact that problem-answer pair datasets are expensive and labor intensive to create.

To be specific, generating effective test inputs for prompt testing presents several challenges. Firstly, generated test inputs must be assigned a ground truth. Secondly, these inputs should exhibit diversity, as this diversity can potentially lead to the successful discovery of failures within the system under test. Finally, the test generation process itself needs to be generalizable and controllable, which ultimately assists developers in creating efficient prompt development workflows. The project aims to address "How to Generate Test Inputs of LLM Prompts?" effectively for LLM-based applications.

2. Baseline Paper

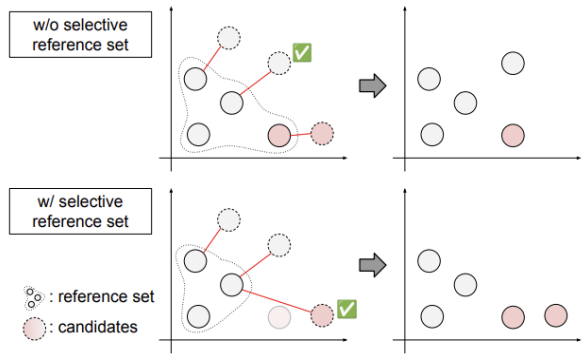
The project draws inspiration from two key baseline approaches:

2.1. Adaptive Testing for LLM-Based Applications: A Diversity-Based Approach[1]

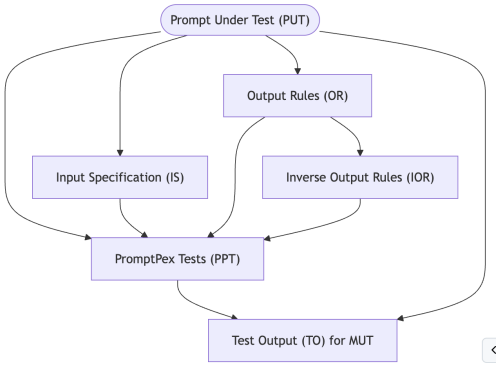
This framework is an Adaptive Random Testing (ART)-inspired approach for LLM application testing. Its core principles involve selecting tests based on a diversity-based distance score and comparing multiple distance metrics. This method selects and prioritizes tests from an existing pool, leading to increased failure discovery and output diversity.

2.2. PromptPex[2]

Prompt Pex focuses on rule-based LLM test input generation and evaluation. It operates by extracting rules that the Prompt Under Test(PUT) requires. Test inputs are then generated to adhere to these extracted rules. For evaluation, it assesses the compliance of the generated tests against the extracted Output Rules.



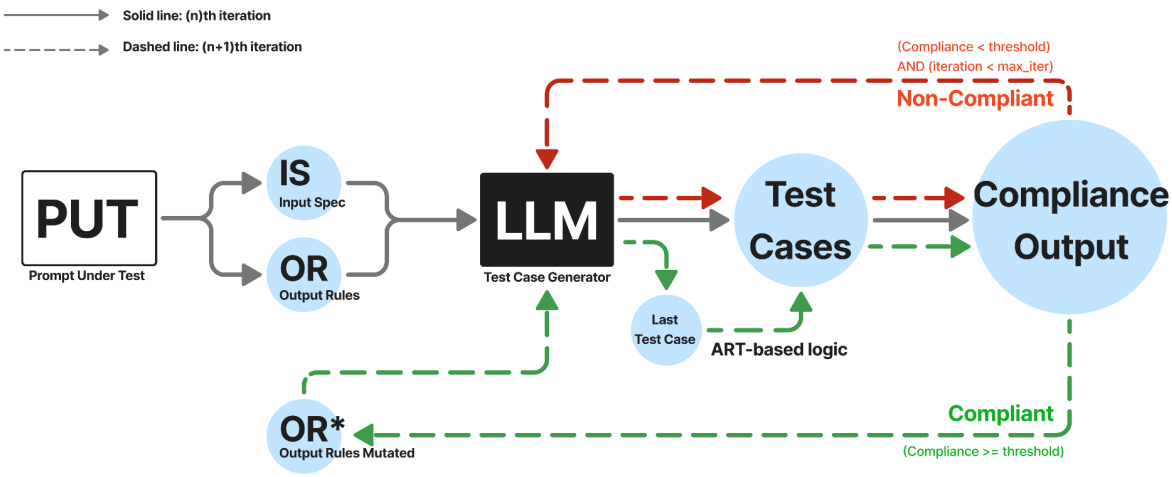
[Fig 1] 2.1 Baseline Approach



[Fig 2] 2.2 Baseline Approach

3. Methodology

The project proposes a high-level approach involving an autoregressive test generation pipeline guided by a knowledge graph. The behavior of test generation is conditional on the result of the previous time step. To enhance diversity, the methodology incorporates Adaptive Random Testing (ART) logic and Output Rule (OR) mutation. The overview of the autoregressive test generation tool is described in [Fig 3].

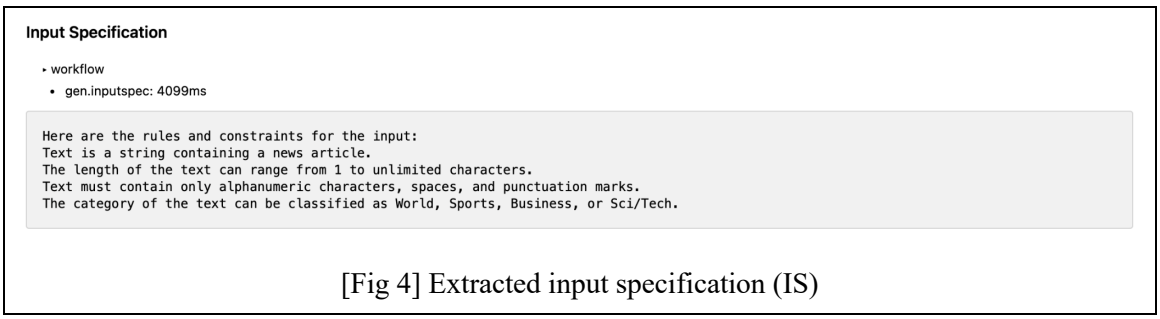


[Fig 3] Overall structure of Autoregressive Test Generation Tool

The methodology is broken down into three main components:

3.1. Rule Extraction & Mutation

1) Extract Rules: This phase involves extracting Input Specifications, Output Rules (ORs), and Inverse Rules (IRs).



[Fig 4] Extracted input specification (IS)

Output Rules

- workflow
  - gen.outputrules: 12325ms

rule
The output response generated by the chatbot must be a single category from the set {World, Sports, Business, Sci/Tech}.
The output response must be determined by the presence of specific keywords or phrases indicative of the respective categories in the provided text "{{text}}".
If the provided text contains keywords or phrases exclusively related to international relations, politics, and global issues, then the output response must be "World".
If the provided text contains mentions of athletes, sports teams, scores, or other competitive events, then the output response must be "Sports".
If the provided text discusses corporate news, financial data, stock prices, mergers, acquisitions, or economic performance indicators, then the output response must be "Business".
If the provided text explores scientific discoveries, technological advancements, innovative research, or astronomical phenomena, then the output response must be "Sci/Tech".

[Fig 5] Extracted output rules (ORs)

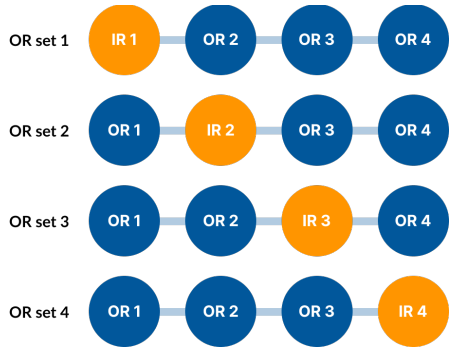
Inverse Output Rules

- workflow
  - gen.inverseoutputrules.single: 2486ms
  - gen.inverseoutputrules.single: 1714ms
  - gen.inverseoutputrules.single: 2496ms
  - gen.inverseoutputrules.single: 1431ms
  - gen.inverseoutputrules.single: 1915ms
  - gen.inverseoutputrules.single: 1907ms

generate inverse output rule
Any output that is a complex phrase combining categories or containing additional irrelevant information is acceptable.
There can exist a response where the exact opposite keywords or phrases are absent from the input text "{{text}}".
The generated inverse rule is: The output response cannot be "World" regardless of the presence of any keywords or phrases related to science, technology, or medicine in the provided text.
The output cannot contain any reference to professional or amateur competitions, scores, results, or related terminology.
The output cannot be described by mentioning a non-business-related topic such as arts, culture, entertainment, sports, or personal experiences.
The output response may not be "Sci/Tech" when discussing historical events, social issues, philosophies, or creative works.

[Fig 6] Extracted inverse output rules (IRs)

2) Mutate Rules: The system attempts diverse sets of ORs until a faulty test input is discovered. This is achieved by composing diverse OR sets through mutating one of the ORs into its corresponding Inverse Rule (IR). Configuration of OR set with mutation one of the elements is described in [Fig 7].



[Fig 7] Mutation of OR set

3.2. ART-based Test Generation

Input generation behavior utilizes ART during the phase where the generated input yields correct behavior. The feature space for this process is based on SBERT embedding.

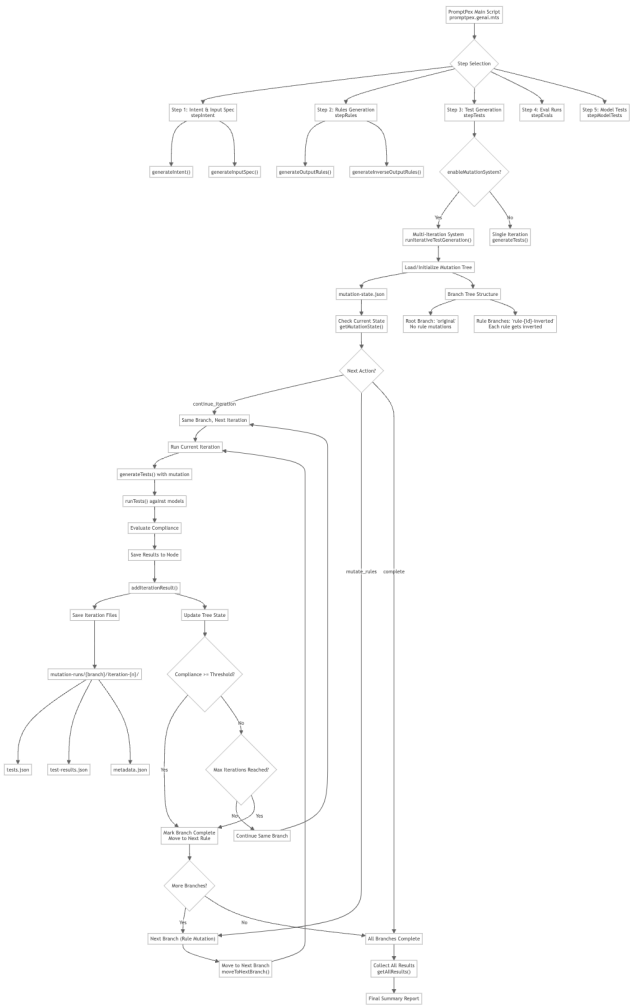
1) Semantic Distance Calculation: The project uses the logic of Sentence BERT (SBERT) to embed each prompt, prioritizing candidates that are semantically different. SBERT is a modification of BERT that employs a Siamese network to produce fixed-size sentence embeddings. These embeddings are optimized for semantic similarity and related downstream tasks.
2) SBERT over NCD: Based on numerous tests, SBERT was found to more effectively detect semantic differences compared to NCD.

3.3. Knowledge Graph Loop

The Knowledge Graph (KG, implemented as `PromptPexMutationTree`) loop guides the generation process. In this context, a "Branch"(implemented as `PromptPexMutationBranch`) refers to a set of rules used for generating tests and evaluating compliance. A "Node"(implemented as `PromptPexMutationNode`) represents a single time step of test generation and compliance evaluation. The system also utilizes threshold parameters (compliance threshold, max iterations per branch) for detailed behavior.

1) If Compliant: If a test is compliant, the system needs to generate more edge cases. It then branches out to new mutated rules and generates test cases using them.
2) If Non-Compliant: If a test is non-compliant, it indicates that the system is exploring a faulty region. In this scenario, the system generates test cases again using the current rules. This KG-guided loop is crucial for the implementation of the autoregressive generation process.

The specific implementation flow of the KG loop is described with [Fig 8].



[Fig 8] Full Flow of Knowledge Graph Loop

4. Result

In order to compare the efficiency of our pipeline, we tested with the same prompt set with PromptPex. The specification and source of each prompt is described in Table 1.

Name of Prompt	Prompt Description	Source
speech-tag	Determine the part of speech for a given word within a sentence using a predefined set of tags. The task may return tags such as noun, verb, adjective among others, or return "Unknown" or "CantAnswer" if the word cannot be classified.	Modified from an example used by Schnabel et. al. [3]
text-to-p	Format a paragraph of text into HTML by splitting	format_text prompt

	it into sentences and wrapping each sentence with paragraph tags, while enhancing key words and phrases with additional HTML tags.	from GhostWriter [4]
shakespeare	Assist users in creating text that mimics the Shakespearean style of writing, including the use of archaic language and stylistic elements typical of the period.	Azure AI Studio Prompt Catalog [5]
sentence	Rewrite a sentence to improve its readability and make it more conversational while preserving its original meaning. This includes simplifying complex phrases and enhancing engagement through fluid structure.	The Big Prompt Library [6]
extract-names	Extract model names from machine learning paper abstracts, returning a structured array of identified model names or "NA" if none are found.	Information extraction prompt from Prompt Hub [7]
elements	Extract important entities from a text, including company names, people names, specific topics, and general themes, presented in a structured list format.	OpenAI documentation [8]
classify	Classify a news article into one of several predefined categories such as World, Sports, Business, or Sci/Tech, based on its content and context.	Prompt used in a tutorial [9]
art-prompt	Create detailed prompts based on user descriptions for generating AI images, focusing on key characteristics, timing, lighting, and the desired emotional impact of the image in a concise single paragraph.	The Big Prompt Library [10]

[Table 1] Description of the prompts used in the evaluation with their sources.

Below, Table 2 shows the percentage of test non-compliance for different prompts on each model. Higher percentage of test non-compliance means more edge cases and failing tests were generated, representing better results.

Prompts	qwen2.5:3b	
	KGA	PPex
speech-tag	**35%**	2%
text-to-p	58%	**72%**

shakespeare	**28%**	5%
sentence	**60%**	21%
extract-names	20%	**31%**
elements	40%	**62%**
classify	**20%**	12%
art-prompt	**47%**	17%
Average	**39%**	28%

[Table 2] Test non-compliance results for the tests generated by Knowledge Graph-guided Autoregression (KGA) vs the baseline PromptPex (PPex) on different models. More test non-compliance (winner shown in bold) represents more challenging tests and hence are better tests.

In Table 3, we compare the non-compliant test percentage between test generation based on original OR set and average of mutated OR sets.

Prompts	qwen2.5:3b	
	original OR set	mutated OR sets (avg)
speech-tag	16.7%	**26.7%**
text-to-p	12.5%	**33.3%**
shakespeare	**33%**	28.5%
sentence	25%	**40%**
extract-names	**33.3%**	11.2%
elements	**50%**	37.5%
classify	0%	**25%**
art-prompt	**50%**	30%
Average	27.6%	**29%**

[Table 3] Test non-compliance results for tests generated from the original OR set, where all entries are OR, and average test non-compliance results for the mutated OR sets, where one of the entries is mutated into IR, by Knowledge Graph-guided Autoregression (KGA) on different models. More test non-compliance (winner shown in bold) represents more challenging tests and hence are better tests.

After all mutation iterations are finished, the generated test cases are categorized based on which rules among the OR set they have violated. For instance, [Fig 9] and [Fig 10] show examples of how the generated test cases are categorized for each prompt's KGA loop.

✅ Compliant Tests (16)

Branch	Model	Scenario	Type	Input	Output	Compliance	rules_compliant
original	qwen2.5:3b		Normal	Upcoming athlete aims for Olympic medal ...	This news article should be categorized under the **Sports**...	✅	100
original	qwen2.5:3b		Normal	An emerging AI system showcases its capa...	This news article would best fit into the "Science/Technolog...	✅	70
original	qwen2.5:3b		Normal		Certainly! Please provide me with the text of the news artic...	✅	0
original	qwen2.5:3b		Normal	international affairs	I'm sorry for the misunderstanding, but to classify the news...	✅	0
original	qwen2.5:3b		Normal	Upcoming athlete aims for Olympic medal ...	Based on the content provided in the news article, it would ...	✅	0
original	qwen2.5:3b		Normal	Researchers from MIT announce a breakthr...	This news article should be classified under the category **...	✅	100
original	qwen2.5:3b		Normal	corporation conducted merger in tech mar...	Based on the content of the provided sentence "corporation c...	✅	100
rule-rule_1-inverted	qwen2.5:3b		Normal	market activity in a mixed context preve...	Based on the provided sentence "market activity in a mixed c...	✅	100
rule-rule_1-inverted	qwen2.5:3b		Normal	but lacks other core sports-related term...	Based solely on the limited information provided in your req...	✅	50
rule-rule_2-inverted	qwen2.5:3b		Normal	internet security, data protection techn...	Based on the content of "internet security" and "data protec...	✅	0
rule-rule_2-inverted	qwen2.5:3b		Normal	fitness tips for professional athletes	Based on the provided title "Fitness Tips for Professional A...	✅	100
rule-rule_3-inverted	qwen2.5:3b		Normal	{{input_text}} discusses recent advancem...	Based on the content described in the input text, which disc...	✅	100
rule-rule_4-inverted	qwen2.5:3b		Normal	input':"Global geopolitical event impact...	Based solely on the content provided in the input news artic...	✅	80
rule-rule_4-inverted	qwen2.5:3b		Normal	input':"International collaborations in ...	Based on the given snippet "International collaborations in ...	✅	100
rule-rule_5-inverted	qwen2.5:3b		Normal	input 'international affairs are driving...	Based on the content of the input sentence "international af...	✅	100
rule-rule_5-inverted	qwen2.5:3b		Normal	input 'global trade unions convene'	This headline "Global trade unions convene" falls under the ...	✅	0

[Fig 9] Example of All Compliant Test Cases of a KGA Loop

❌ Non-Compliant Tests (20)

Violated Rules [1, 2, 3, 4, 5] (2 tests)

Branch	Model	Scenario	Type	Input	Output	Compliance	Violated Rules	rules_compliant
original	qwen2.5:3b		Normal	international relations etc.	Based solely on the instruction to classify this news articl...	?	[1, 2, 3, 4, 5]	0
rule-rule_1-inverted	qwen2.5:3b		Normal	input 'An analysis on cyber security and...	Based on the content provided in the news article excerpt "A...	❌	[1, 2, 3, 4, 5]	0

Violated Rules [1, 2] (2 tests)

Branch	Model	Scenario	Type	Input	Output	Compliance	Violated Rules	rules_compliant
original	qwen2.5:3b		Normal	athlete participated in Olympics	Based on the content provided ("athlete participated in Olym...	?	[1, 2]	0
rule-rule_3-inverted	qwen2.5:3b		Normal	{{input_text}} is a string containing co...	Based on the content provided, which discusses geopolitical ...	?	[1, 2]	0

Violated Rules [1] (1 tests)

Branch	Model	Scenario	Type	Input	Output	Compliance	Violated Rules	rules_compliant
rule-rule_2-inverted	qwen2.5:3b		Normal	politics, global affairs, international ...	To classify the given news article with the provided terms (...	?	[1]	0

[Fig 10] Example of Non-Compliant Test Cases of a KGA Loop, Categorized based on Violated Rules

5. Conclusion

5.1. Limitations of the project

We tried to show the improvement made compared to the baseline paper 'PromptPex', by testing within the same environments provided in the paper. However, due to the limited performance of team members' laptops, we were unable to run large-parameter local model(gemma2:9b). On the other hand, models with very small parameters(llama3.2:1b) did not produce meaningful outputs, making them unsuitable for testing. In addition, we couldn't use API-based model(gpt-4o-mini) that did not offer a predictable or reasonable cost structure.

5.2. Contribution

Acknowledging the project's limitations, we argue our contribution in mainly two ways. First, with the knowledge graph-guided loop system, we propose a meaningful approach in generating diverse failure test suites. The comparison of non-compliant test case percentage demonstrates the clear potential of diverse test case generation based on specification mutations. Second, categorization of failing test cases based on rule violation patterns leads to a systematic analysis of faulty regions, instead of simple compliant/non-compliant binary results of the baseline paper.

5.3. Future Works

Testing our system with better performing language models would be the next step, followed by repetitive iteration to ensure quality with a large number of test generations. Analysis of clustered faulty regions would be the next step.

Reference

[1] Yoon, Juyeon, Robert Feldt, and Shin Yoo. "Adaptive testing for LLM-based applications: A diversity-based approach." 2025 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 2025.

[2] Sharma, Reshabh K., et al. "PromptPex: Automatic Test Generation for Language Model Prompts." arXiv preprint arXiv:2503.05070 (2025).

[3] Tobias Schnabel and Jennifer Neville. Symbolic prompt program search: A structure-aware approach to efficient compile-time prompt optimization. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 670–686, 2024.

[4] Ghostwriter: Augmenting collaborative human-ai writing experiences through personalization and agency. https://arxiv.org/abs/2402.08855, 2024.

[5] Azure ai studio prompt catalog. https://ai.azure.com/explore/prompts/shakespeare_writing_assistant/version/0.0.1/registry/azureml?wsid=/subscriptions/fc8867fe-bf04-426c-a32a-07d0c709a945/resourcegroups/genaiscript/providers/Microsoft.MachineLearningServices/workspaces/genaiscript&tid=512451b2-ca3c-4016-b97c10bd8c704cfc&promptType=promptSamples&promptSharedInHub=false, 2023.

[6] GitHub - openai/evals: Evals is a framework for evaluating LLMs and LLM systems, and an open-source registry of benchmarks. — github.com. https://github.com/openai/evals. [Accessed 13-12-2024].

[7] Prompt examples from the website. https://www.promptingguide.ai/prompts/information-extraction/extract-models, 2023.

[8] Openai documentation: Best practices for prompt engineering with the openai api. https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api, 2023.

[9] How to use llama2 tutorial for text classification. https://pupuweb.com/how-use-llama-2-text-classification-tasks/, 2023.

[10] The big prompt library. https://github.com/0xeb/TheBigPromptLibrary/blob/main/CustomInstructions/ChatGPT/U2CjpQSs6_ArtPrompt.md, 2023.