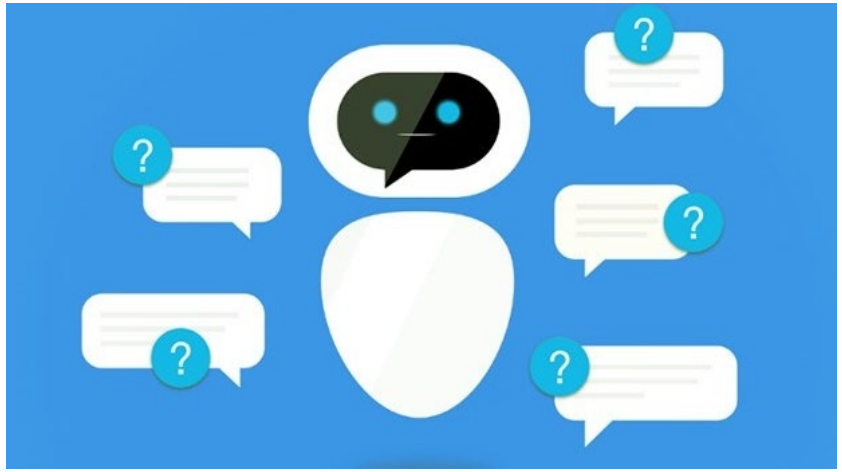
Exploring ChatGPT's Capabilities on Vulnerability Management

Peiyu Liu **Junming Liu** Lirong Fu Kangjie Lu Yifan Xia
Xuhong Zhang Wenzhi Chen Haiqin Weng Shouling Ji Wenhai Wang

LMs Have Been Widely Used in Diverse Domains



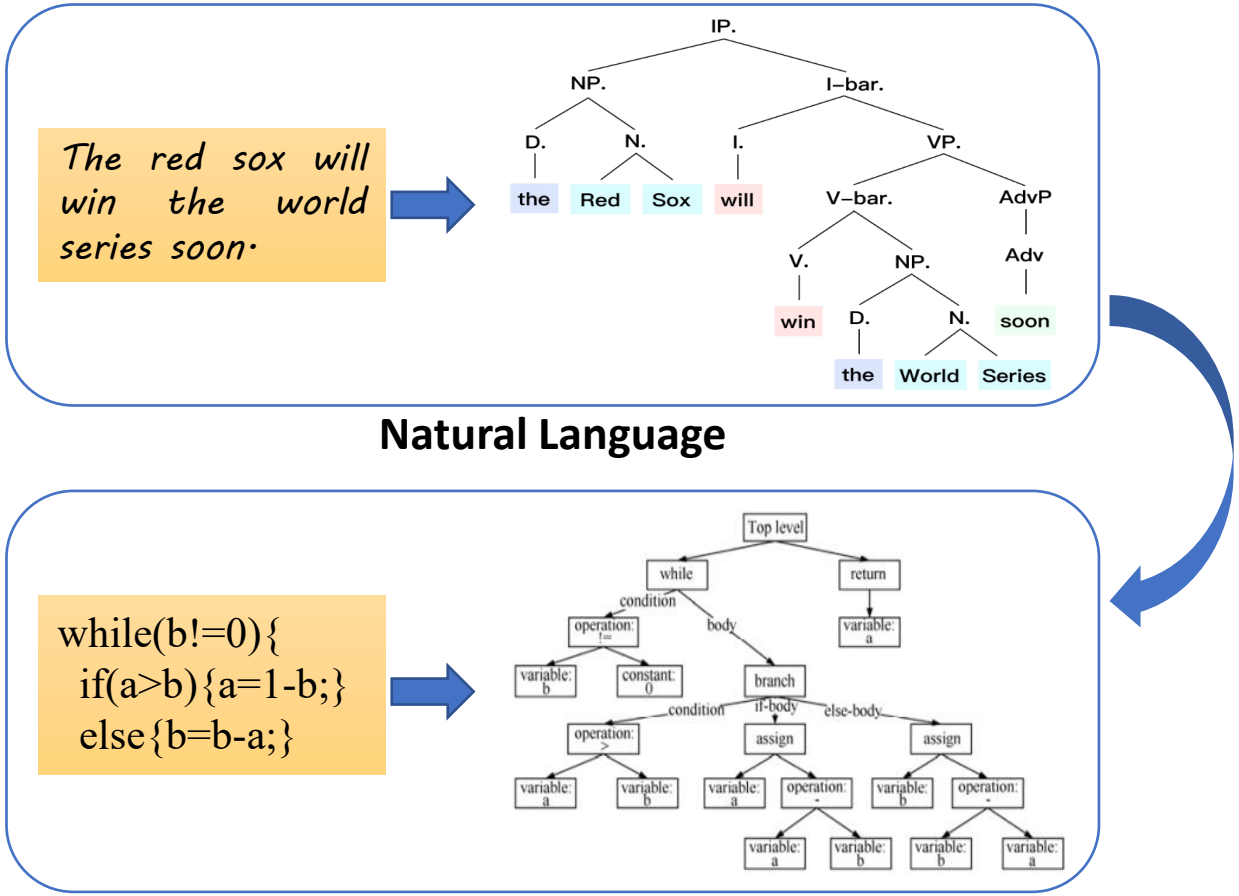
Question Answering

Data Augmentation

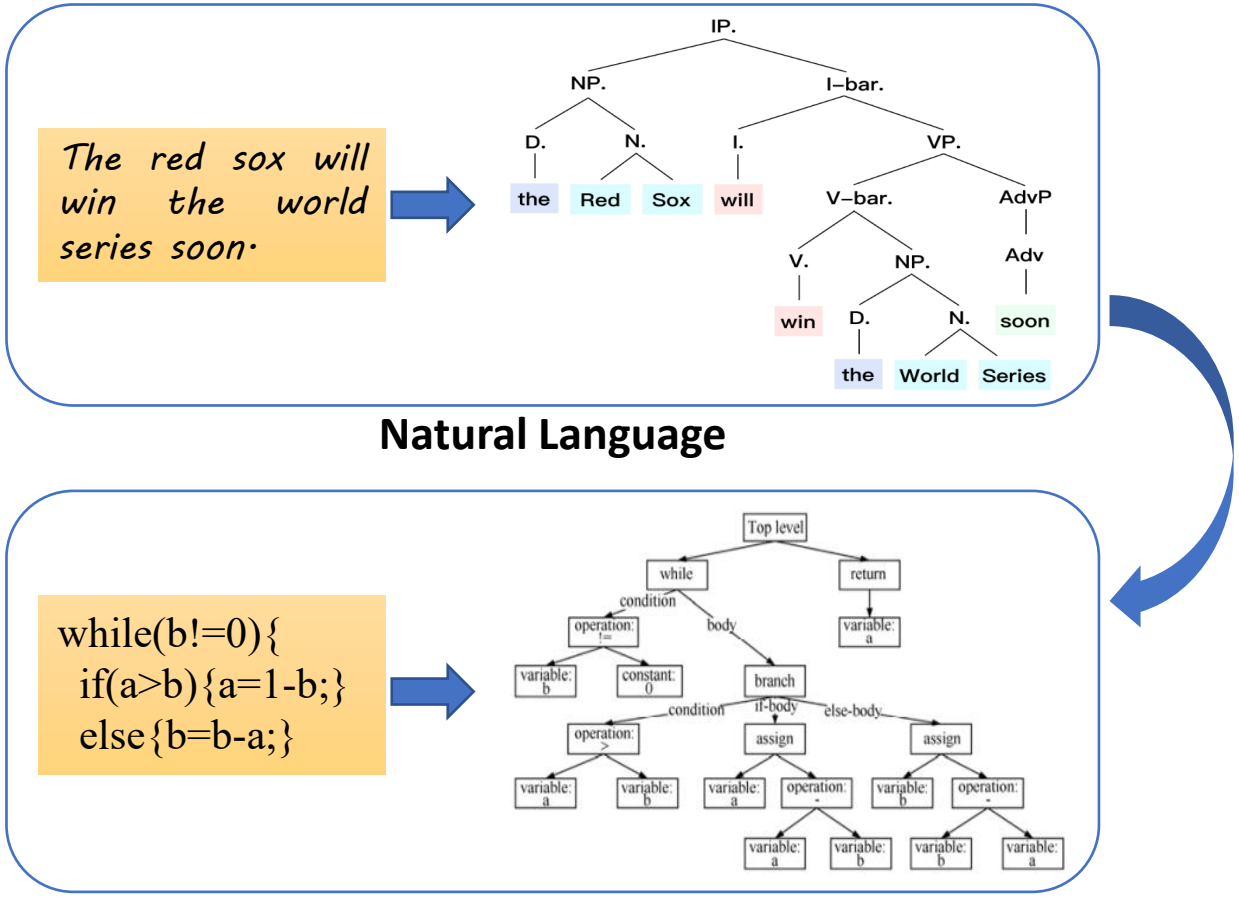
Medical Assistant

Education

Researchers Turn to Utilize ChatGPT for Code-related Analysis

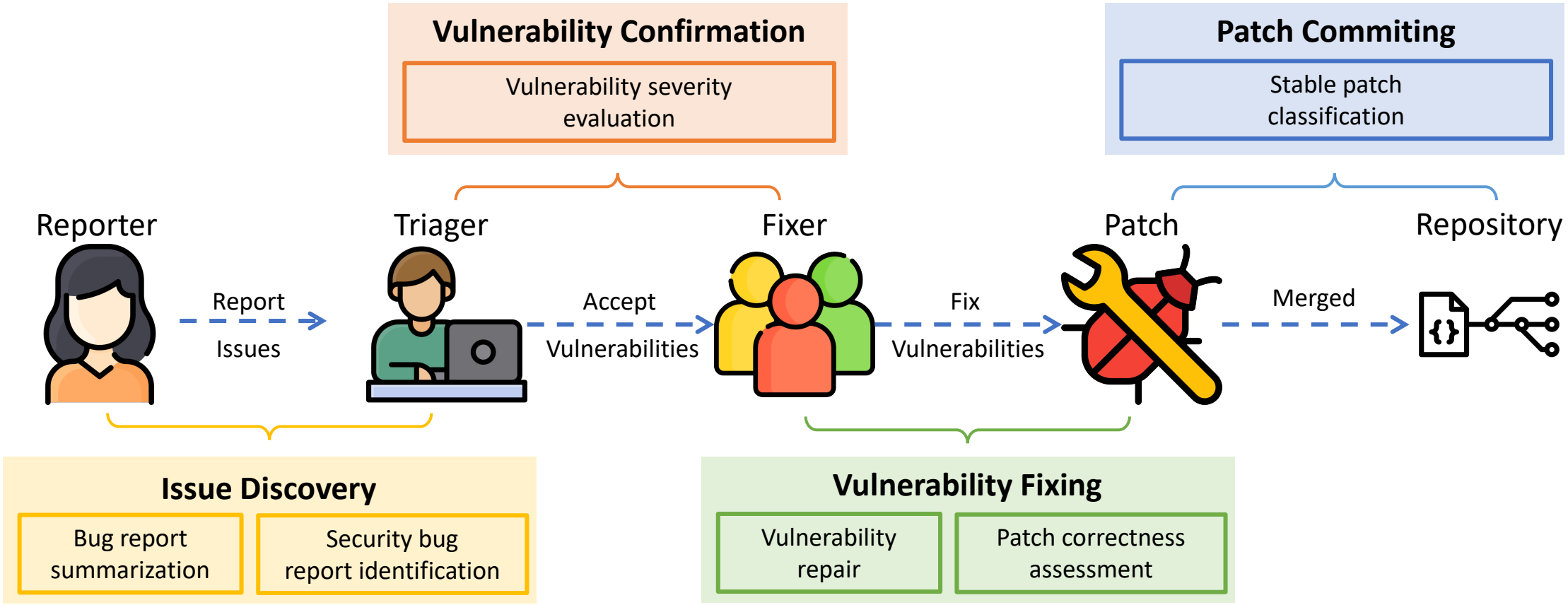


Researchers Turn to Utilize ChatGPT for Code-related Analysis



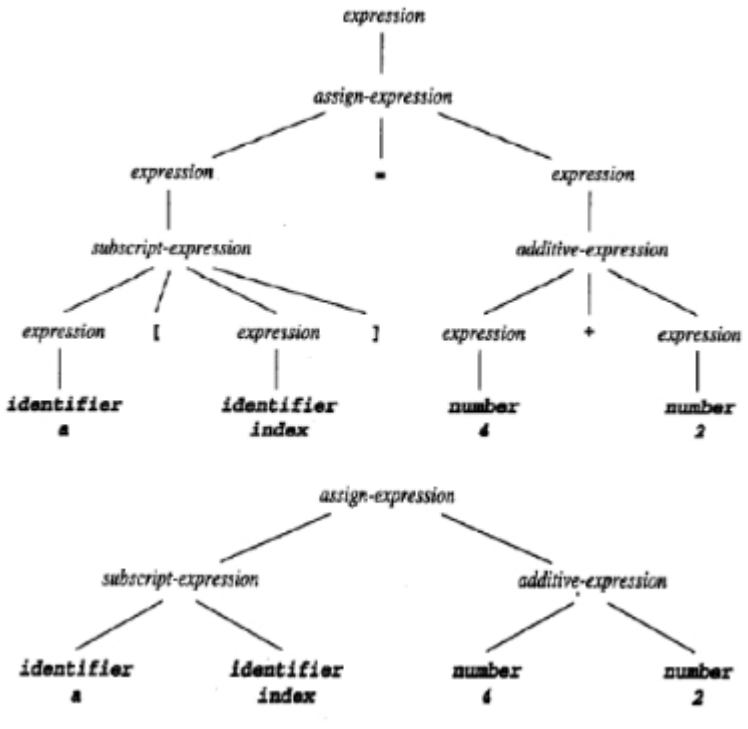
Prior works show that ChatGPT has the capabilities of processing foundational code analysis tasks, such as AST generation.

Software-Vulnerability Management

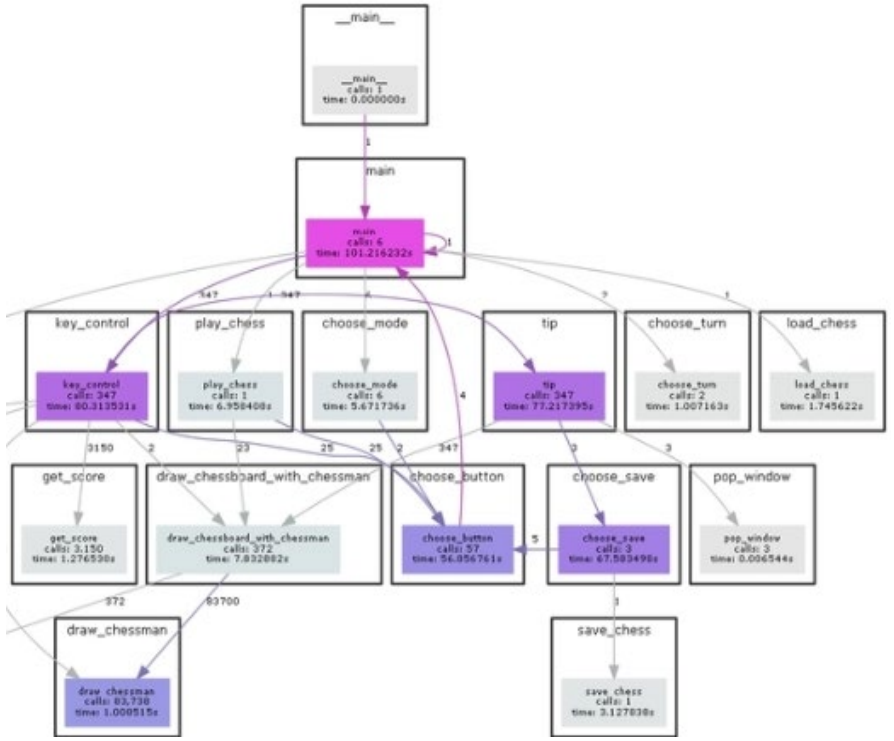


- Prior researches focus on several specific tasks rather than the entire lifecycle of vulnerability management.
- vulnerability management include a comprehensive process that consists of complex phases.

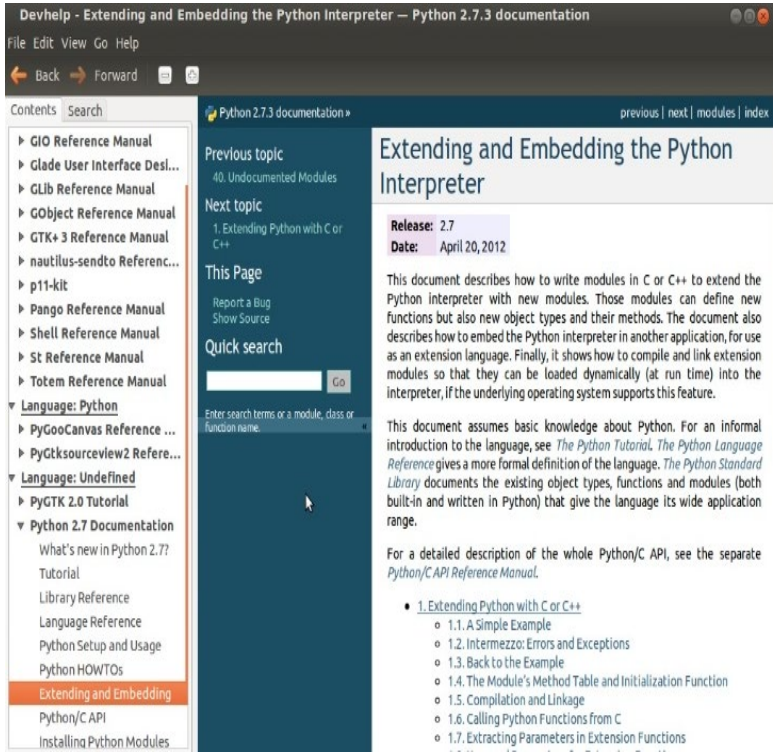
Software-Vulnerability Management



Code Syntax



Program Semantics



Software Documents

Vulnerability management tasks require a deep and all-encompassing understanding of code syntax, program semantics, and related documents.

Software-Vulnerability Management

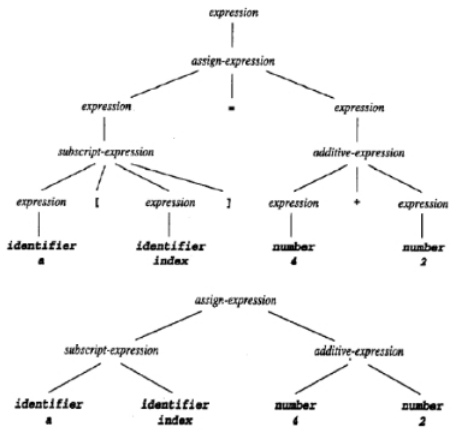
Vulnerability Confirmation

Patch Commiting

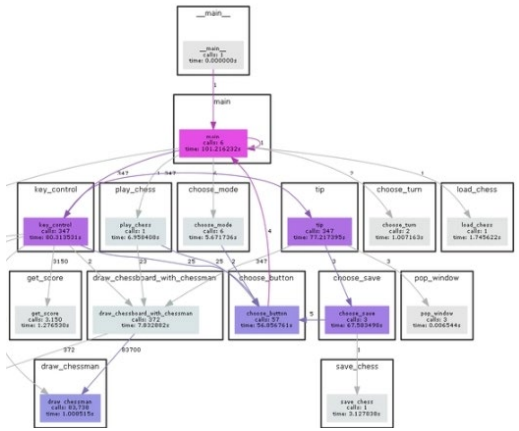
Reporter — Report Issues → Triager — Accept Vulnerabilities → Fixer — Fix Vulnerabilities → Patch — Merged → Repository

Issue Discovery

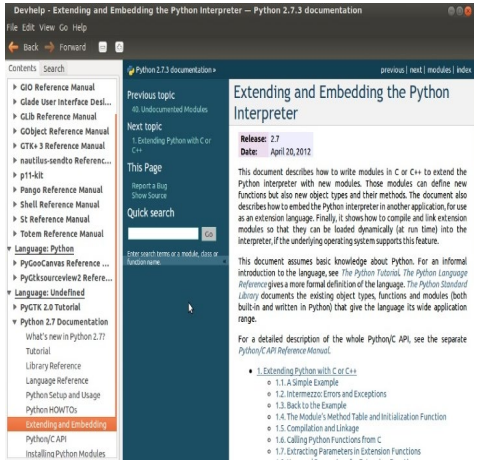
Vulnerability Fixing



Code Syntax



Program Semantics



Software Documents

Can ChatGPT directly assist software maintainers in diverse tasks during the whole vulnerability management process?

Exploring ChatGPT's Capabilities on Vulnerability Management

RQ1: Does ChatGPT achieve capability on par with the SOTAs?

RQ2: How do prompt engineering methods impact ChatGPT's performance?

RQ3: What is the promising future direction to improve ChatGPT's performance on each task?

Evaluated Tasks, Baselines and Dataset

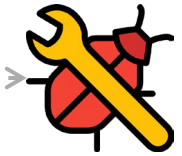
- **11 SOTA approaches are derived from the top venues.**
- **The test dataset used in this paper contains 70,346 samples (19,355,711 tokens).**

Issue Discovery

Bug report summarization	iTAPE (RNN- Attention)
	Farsec (Ensemble Learning)
Security bug report identification	DKG (Knowledge Graphs)
	CASMS (CNN-BiLSTM)

Vulnerability Confirmation

Vulnerability severity evaluation	DiffCVSS (BiLSTM- Attention)



Vulnerability Fixing

Vulnerability repair	LLMset
	ExtractFix (Static Analysis)
Patch correctness assessment	Quatrain (BiLSTM- Attention)
	Invalidator (CodeBERT)
	Panther (Ensemble Learning)

Patch Commiting

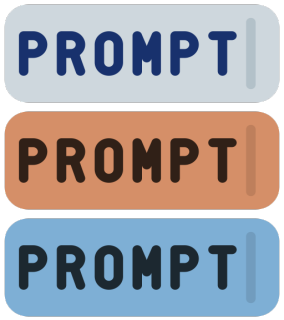
Stable patch classification	PatchNet (CNN-LSTM- Attention)

Evaluated Tasks, Baselines and Dataset

- **11 SOTA approaches are derived from the top venues.**
- **The test dataset used in this paper contains 70,346 samples (19,355,711 tokens).**

collect dataset
provided by the SOTA

evaluate ChatGPT's performance
with the same metrics used by each SOTA



investigate the influence of
prompt engineering methods

analyze ChatGPT's responses to
identify the bottlenecks

Prompt Templates

Name	Template
0-shot	**USER** <task description> <input>
1-shot	**USER** <task description> <demonstration example> <input>
few-shot	**USER** <task description> <demonstration example 1> <demonstration example 2> <demonstration example 3> <demonstration example 4> <input>
general-info	**SYSTEM** <role> <task description> <reinforce> **USER** <task description> <task confirmation> **ASSYSTANT** <task confirmation> **USER** <positive feedback> <input> <zero-CoT> <right>
expertise	**SYSTEM** <role> <task description> <expertise> <reinforce> **USER** <expertise> <task description> <task confirmation> **ASSYSTANT** <task confirmation> **USER** <positive feedback> <input> <zero-CoT> <right>
self-heuristic	**SYSTEM** <role> <task description> <reinforce> **USER** <knowledge> <task description> <task confirmation> **ASSYSTANT** <task confirmation> **USER** <positive feedback> <input> <zero-CoT> <right>

SYSTEM *You are Frederick, an AI expert in bug report analysis. Your task is to decide whether a given bug report is a security bug report (SBR) or non−security bug report (NBR). When analyzing the bug report, take into account that bug reports related to memory leak or null pointer problems should be seen as security bug report. Remember, you're the best AI bug report analyst and will use your expertise to provide the best possible analysis.*

USER *A security bug report is a bug report describing one or more vulnerabilities of a software. Besides, bug reports that directly mention "memory leak" or "null pointer" problems must be seen as security bug reports. I will give you a bug report and you will analyze it, step−by−step, to know whether or not it is a security bug report. Got it?*

ASSISTANT *Yes, I understand. I am Frederick, and I will analyze the bug report.*

USER *Great! Let's begin then :)*
For the bug report:
<bug report>
−−−−−−−−−
Is this bug report (A) a security bug report (SBR), or (B) a non−security bug report (NBR).
Answer: Let's think step-by-step to reach the right conclusion,

An example of the expertise prompt. After removing the bold pink text, the rest represents the general-info prompt.

Bug Report Summarization

summary a given bug report

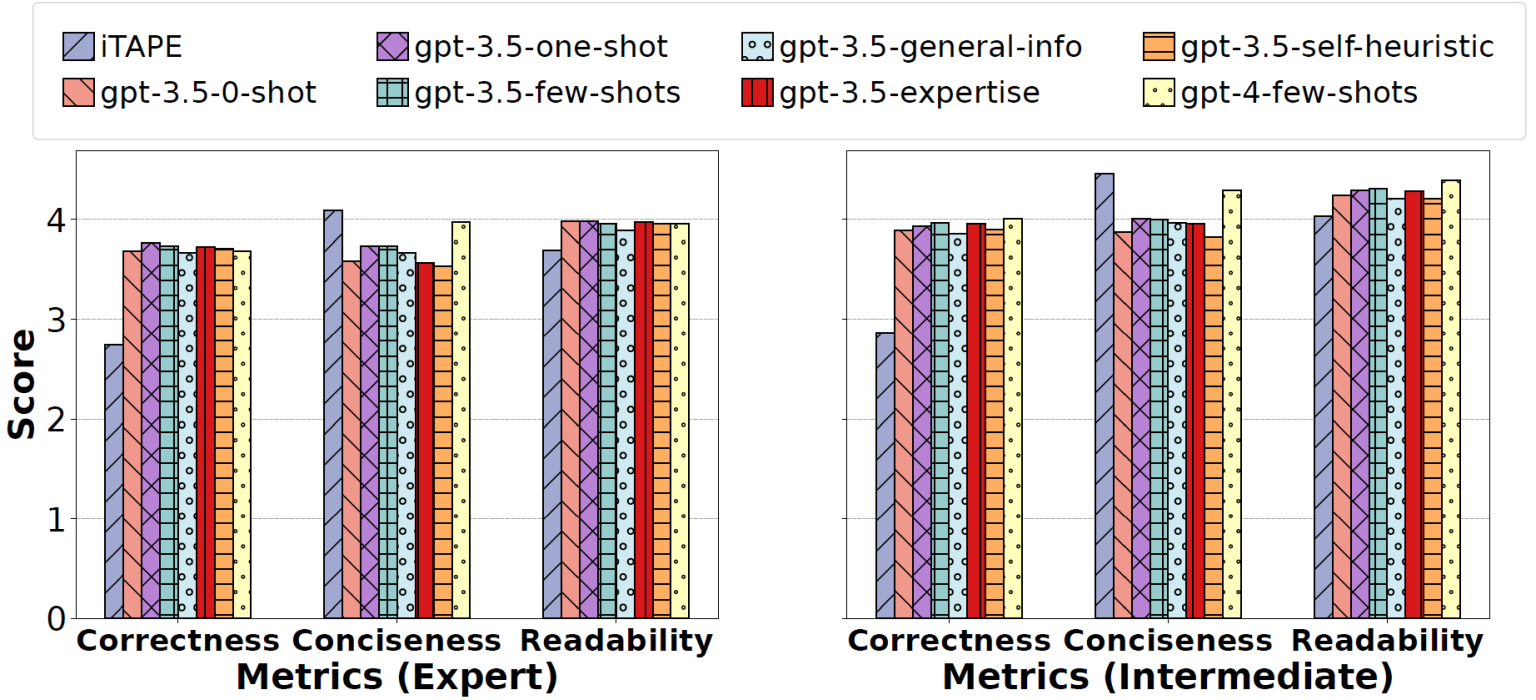
- **ChatGPT can obtain outstanding performance in this task.**

Approach	Prompt	Dataset	ROUGE-1			ROUGE-2			ROUGE-L		
			F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
iTAPE [18]	-	test	31.36	32.61	31.72	13.12	13.77	13.34	27.79	30.10	29.32
gpt-3.5	0-shot	probe-test	34.33	30.54	42.11	11.05	9.66	13.99	27.95	24.78	34.41
gpt-3.5	1-shot	probe-test	36.82	33.54	43.67	13.27	11.97	16.13	30.86	28.03	35.71
gpt-3.5	few-shot	probe-test	37.30	33.91	44.26	13.99	12.61	16.92	31.52	28.57	37.53
gpt-3.5	general-info	probe-test	32.37	28.23	41.12	10.73	9.25	14.10	26.55	23.10	33.83
gpt-3.5	expertise	probe-test	33.27	29.50	41.23	11.30	9.87	14.37	27.58	24.35	34.32
gpt-3.5	self-heuristic	probe-test	33.08	30.25	40.16	11.26	10.28	13.88	27.53	25.10	33.56
gpt-4	few-shot	probe-test	**40.38**	**39.07**	**44.35**	**15.86**	**15.26**	**17.69**	**34.30**	**33.12**	**37.75**
gpt-4	few-shot	test	39.17	37.52	43.45	14.34	13.58	16.35	33.23	31.77	36.92

The evaluation result on bug report summarization.

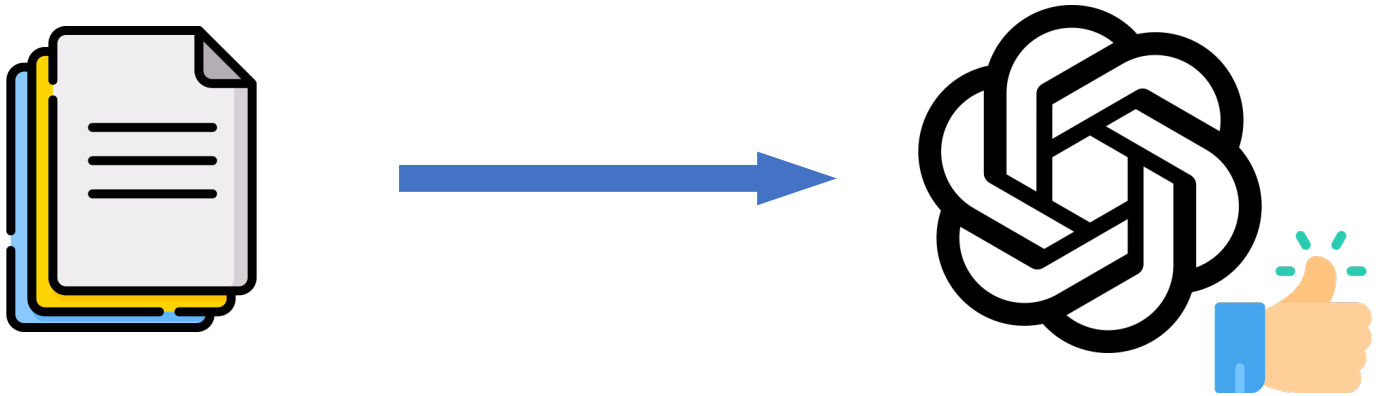
Bug Report Summarization

- ChatGPT can obtain outstanding performance in this task.
- **User Study: In real-world scenarios, ChatGPT has better correctness and readability.**



Bug Report Summarization

- ChatGPT can obtain outstanding performance in this task.
- In real-world scenarios, ChatGPT has better correctness and readability.
- **The results encourage software maintainers to use ChatGPT for bug report summarization and other vulnerability management tasks related to natural language processing.**



Security Bug Report Identification

whether a given bug report is security-related

Approach	Prompt	Dataset	R	FPR	P	F1	G
DKG [57]	-	test	0.70	0.02	**0.74**	**0.71**	0.81
CASMS [35]	-	test	0.73	0.28	-	-	0.72
Farsec [49]	-	test	0.57	0.16	0.40	0.43	0.64
gpt-3.5	0-shot	probe-test	0.35	0.02	0.21	0.27	0.52
gpt-3.5	1-shot	probe-test	0.76	0.09	0.12	0.21	0.83
gpt-3.5	few-shot	probe-test	0.88	0.06	0.21	0.34	0.91
gpt-3.5	general-info	probe-test	0.29	0.01	0.26	0.28	0.45
gpt-3.5	expertise	probe-test	0.71	0.01	0.57	0.63	0.82
gpt-3.5	self-heuristic	probe-test	0.29	**0.00**	0.56	0.38	0.45
gpt-4	expertise	probe-test	**0.94**	0.04	0.27	0.42	**0.95**
gpt-4	expertise	test	0.68	0.04	0.53	0.57	0.79

The evaluation result on security bug report identification.
R = Recall. P = Precision. FPR = False Positive Rate.
G = G-measure.

- **ChatGPT can outperform two baselines.**

Security Bug Report Identification

Approach	Prompt	Dataset	R	FPR	P	F1	G
DKG [57]	-	test	0.70	0.02	**0.74**	**0.71**	0.81
CASMS [35]	-	test	0.73	0.28	-	-	0.72
Farsec [49]	-	test	0.57	0.16	0.40	0.43	0.64
gpt-3.5	0-shot	probe-test	0.35	0.02	0.21	0.27	0.52
gpt-3.5	1-shot	probe-test	0.76	0.09	0.12	0.21	0.83
gpt-3.5	few-shot	probe-test	0.88	0.06	0.21	0.34	0.91
gpt-3.5	general-info	probe-test	0.29	0.01	0.26	0.28	0.45
gpt-3.5	expertise	probe-test	0.71	0.01	0.57	0.63	0.82
gpt-3.5	self-heuristic	probe-test	0.29	**0.00**	0.56	0.38	0.45
gpt-4	expertise	probe-test	**0.94**	0.04	0.27	0.42	**0.95**
gpt-4	expertise	test	0.68	0.04	0.53	0.57	0.79

The evaluation result on security bug report identification.
R = Recall. P = Precision. FPR = False Positive Rate.
G = G-measure.

- ChatGPT can outperform two baselines.
- **ChatGPT cannot obtain capability on par with DKG.**

Security Bug Report Identification

Approach	Prompt	Dataset	R	FPR	P	F1	G
DKG [57]	-	test	0.70	0.02	**0.74**	**0.71**	0.81
CASMS [35]	-	test	0.73	0.28	-	-	0.72
Farsec [49]	-	test	0.57	0.16	0.40	0.43	0.64
gpt-3.5	0-shot	probe-test	0.35	0.02	0.21	0.27	0.52
gpt-3.5	1-shot	probe-test	0.76	0.09	0.12	0.21	0.83
gpt-3.5	few-shot	probe-test	0.88	0.06	0.21	0.34	0.91
gpt-3.5	general-info	probe-test	0.29	0.01	0.26	0.28	0.45
gpt-3.5	expertise	probe-test	0.71	0.01	0.57	0.63	0.82
gpt-3.5	self-heuristic	probe-test	0.29	**0.00**	0.56	0.38	0.45
gpt-4	expertise	probe-test	**0.94**	0.04	0.27	0.42	**0.95**
gpt-4	expertise	test	0.68	0.04	0.53	0.57	0.79

The evaluation result on security bug report identification.
R = Recall. P = Precision. FPR = False Positive Rate.
G = G-measure.

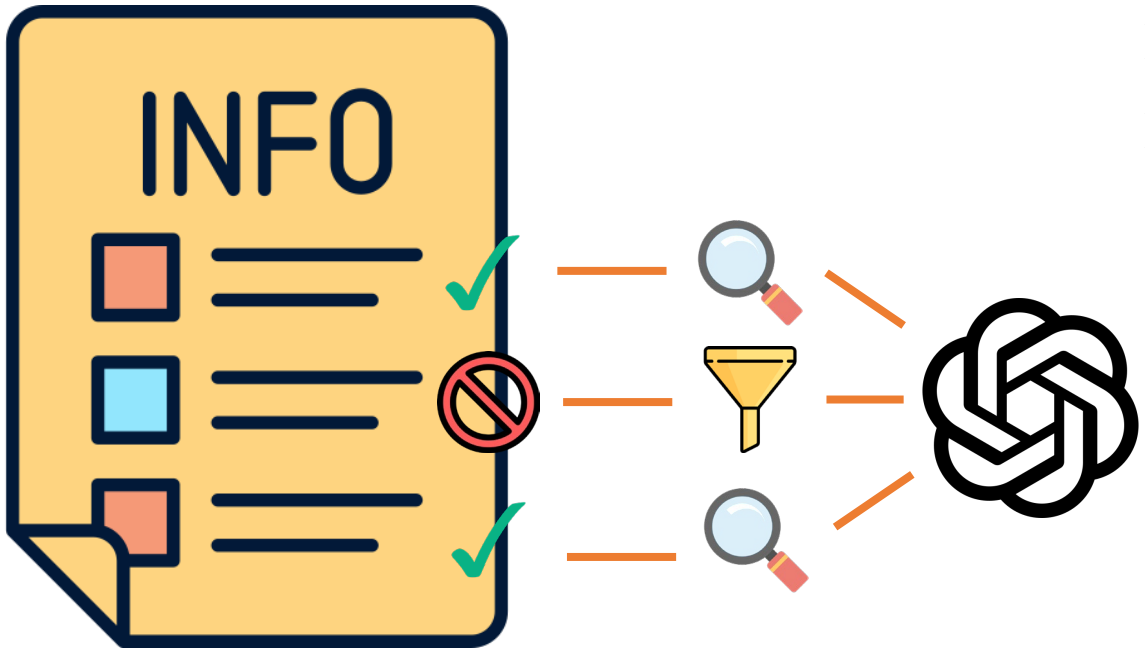
- ChatGPT can outperform two baselines.
- ChatGPT cannot obtain capability on par with DKG.
- **ChatGPT may learn some unrelated information from the labeled sample.**

  ChatGPT tends to mistakenly mark reports that contain unrelated words of the example report as security-related

```
USER  Decide whether a bug report is a security bug report (SBR) or non-security bug
      report (NBR).
      Bug report: Issue 836 : Memory Leak in about:memory      shekariyer
      Product Version: 0.2.149.27 (1583)      URLs (if applicable) : about:memory
      Other browsers tested: None      Safari 3: N/A Firefox 3: N/A IE 7: N/A
      1. Open a new tab and enter "about:memory" in the address bar 2. Monitor the
      memory usage of Chrome 3. Refresh the tab periodically (say 15 sec) and watch
      the memory grow. Memory should not grow just because refresh is invoked
      Memory usage should be constant
      Category: security bug report (SBR)
      ###
      Bug report: """<bug report>"""
      Category:
```

An example of the 1-shot prompt

Security Bug Report Identification



- ChatGPT can outperform two baselines.
- ChatGPT cannot obtain capability on par with DKG.
- ChatGPT may learn some unrelated information from the labeled sample.

when providing demonstration examples, how to make ChatGPT focus on helpful information rather than irrelevant content is an interesting question.

Security Bug Report Identification

Approach	Prompt	Dataset	R	FPR	P	F1	G
DKG [57]	-	test	0.70	0.02	**0.74**	**0.71**	0.81
CASMS [35]	-	test	0.73	0.28	-	-	0.72
Farsec [49]	-	test	0.57	0.16	0.40	0.43	0.64
gpt-3.5	0-shot	probe-test	0.35	0.02	0.21	0.27	0.52
gpt-3.5	1-shot	probe-test	0.76	0.09	0.12	0.21	0.83
gpt-3.5	few-shot	probe-test	0.88	0.06	0.21	0.34	0.91
gpt-3.5	general-info	probe-test	0.29	0.01	0.26	0.28	0.45
gpt-3.5	expertise	probe-test	0.71	0.01	0.57	0.63	0.82
gpt-3.5	self-heuristic	probe-test	0.29	**0.00**	0.56	0.38	0.45
gpt-4	expertise	probe-test	**0.94**	0.04	0.27	0.42	**0.95**
gpt-4	expertise	test	0.68	0.04	0.53	0.57	0.79

The evaluation result on security bug report identification.
R = Recall. P = Precision. FPR = False Positive Rate.
G = G-measure.

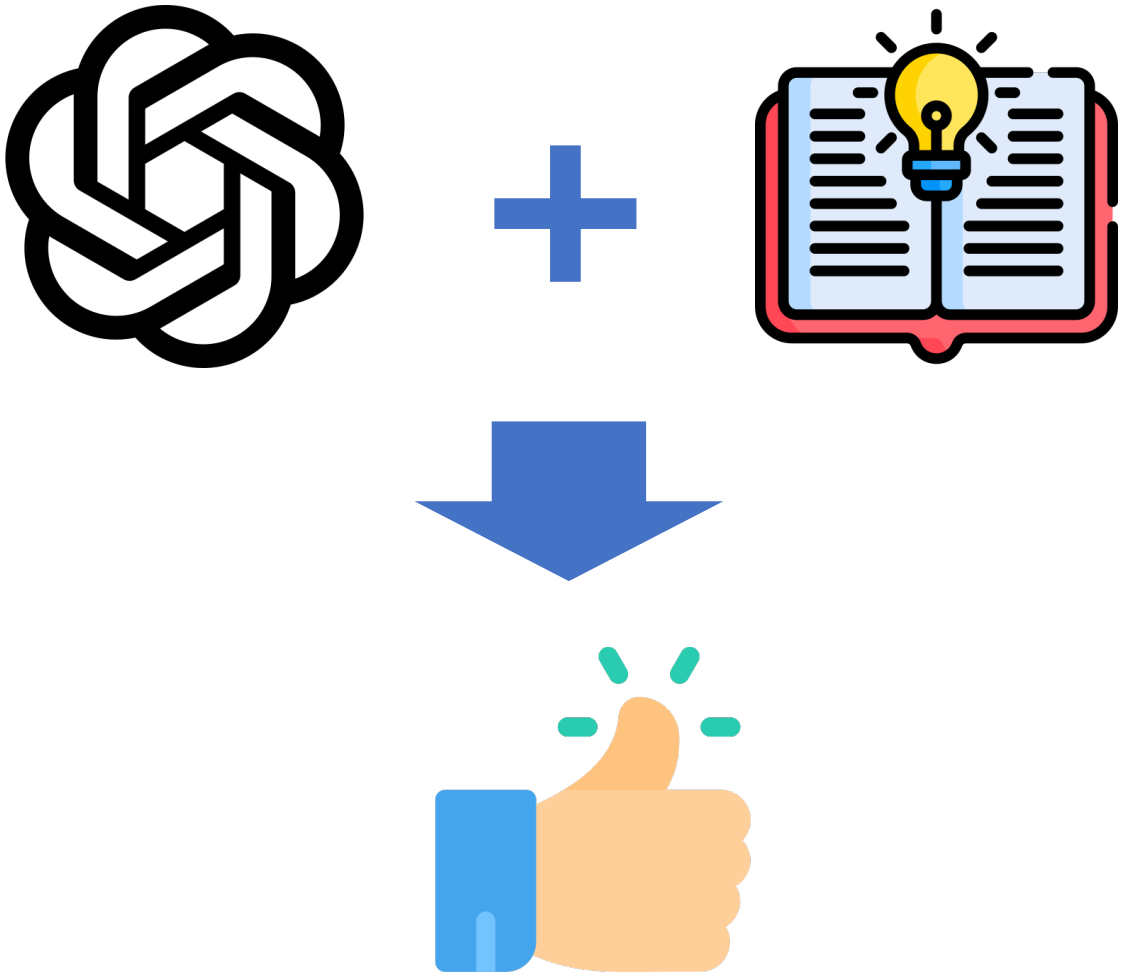
- ChatGPT can outperform two baselines.
- ChatGPT cannot obtain capability on par with DKG.
- ChatGPT may learn some unrelated information from the labeled sample.
- **ChatGPT has hallucinations in understanding what a security bug report is.**

  For instance, ChatGPT "thinks" memory leakage and null pointer dereference are not security-related.

"A security bug report is a bug report describing one or more vulnerabilities of a software. Besides, bug reports that directly mention "memory leak" or "null pointer" problems must be seen as security bug reports."

An example of domain knowledge in expertise prompt

Security Bug Report Identification



- ChatGPT can outperform two baselines.
- ChatGPT cannot obtain capability on par with DKG.
- ChatGPT may learn some unrelated information from the labeled sample.
- ChatGPT has hallucinations in understanding what a security bug report is.

Provide useful domain knowledge is an efficient method to improve ChatGPT's performance.

"A security bug report is a bug report describing one or more vulnerabilities of a software. Besides, bug reports that directly mention "memory leak" or "null pointer" problems must be seen as security bug reports."

An example of domain knowledge in expertise prompt

Vulnerability Severity Evaluation

map a function to the CVSS metrics based on its description

- **ChatGPT's performance is slightly inferior to the SOTA approach.**

Approach	Prompt	Dataset	AV						AC		PR		UI	
			Network		Adjacent		Physical		High		High		Required	
			R	P	R	P	R	P	R	P	R	P	R	P
DiffCVSS [48]	-	test	0.9242	**0.9384**	0.8750	0.9333	0.8852	0.9153	**0.9151**	**0.9238**	**0.9452**	0.9324	**0.9167**	**0.9296**
gpt-3.5	0-shot	probe-test	0.7143	0.5556	0	N/A	0	N/A	0.4286	0.6923	0.3684	0.5000	0	N/A
gpt-3.5	1-shot	probe-test	**1.0000**	0.2206	0	N/A	0.0909	**1.0000**	0.2857	1.0000	0.1053	**1.0000**	0.2667	0.3077
gpt-3.5	few-shot	probe-test	**1.0000**	0.4285	0.4444	0.6667	0.3636	0.4444	0.6190	0.6842	0.2632	0.3333	0.6667	0.2703
gpt-3.5	general-info	probe-test	0.7857	0.4783	0	N/A	0.1667	0.5000	0.8095	0.3269	0.7368	0.2188	0.4000	0.3000
gpt-3.5	expertise	probe-test	0.8571	0.5714	0.5000	0.6667	0.0833	**1.0000**	0.8095	0.2982	0.5263	0.3704	0.2667	0.2857
gpt-3.5	self-heuristic	probe-test	**1.0000**	*0.7368*	0.7500	**1.0000**	**1.0000**	0.9231	0.8095	0.5484	0.8421	0.6400	0.9333	0.5000
gpt-4	self-heuristic	probe-test	**1.0000**	*0.7368*	**1.0000**	**1.0000**	0.9167	0.9167	0.9048	0.6786	0.8947	0.7083	0.8667	0.7647
gpt-4	self-heuristic	test	0.9848	0.7738	0.9063	0.9355	0.9167	0.8333	0.7961	0.7321	0.8941	0.7917	0.7714	0.8852

The evaluation result on vulnerability severity evaluation. AV = Attack Vector. AC = Attack Complexity. PR = Privileges Required. UI = User Interaction. R = Recall. P = Precision.

Vulnerability Severity Evaluation

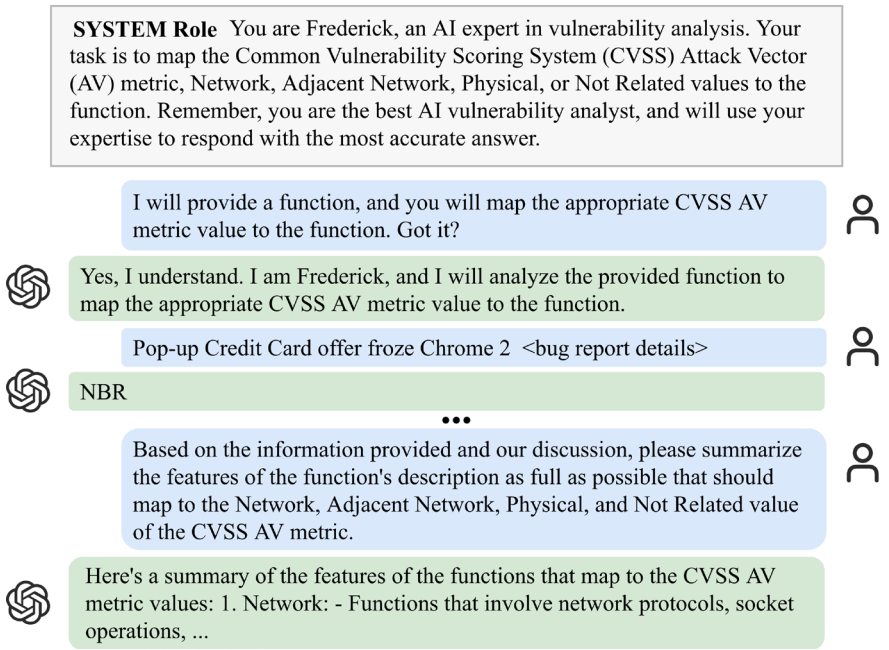
- ChatGPT's performance is slightly inferior to the SOTA approach.
- **Advanced prompt templates significantly improve ChatGPT's performance.**

Approach	Prompt	Dataset	AV						AC		PR		UI	
			Network		Adjacent		Physical		High		High		Required	
			R	P	R	P	R	P	R	P	R	P	R	P
DiffCVSS [48]	-	test	0.9242	**0.9384**	0.8750	0.9333	0.8852	0.9153	**0.9151**	**0.9238**	**0.9452**	0.9324	**0.9167**	**0.9296**
gpt-3.5	0-shot	probe-test	0.7143	0.5556	0	N/A	0	N/A	0.4286	0.6923	0.3684	0.5000	0	N/A
gpt-3.5	1-shot	probe-test	**1.0000**	0.2206	0	N/A	0.0909	**1.0000**	0.2857	1.0000	0.1053	**1.0000**	0.2667	0.3077
gpt-3.5	few-shot	probe-test	**1.0000**	0.4285	0.4444	0.6667	0.3636	0.4444	0.6190	0.6842	0.2632	0.3333	0.6667	0.2703
gpt-3.5	general-info	probe-test	0.7857	0.4783	0	N/A	0.1667	0.5000	0.8095	0.3269	0.7368	0.2188	0.4000	0.3000
gpt-3.5	expertise	probe-test	0.8571	0.5714	0.5000	0.6667	0.0833	**1.0000**	0.8095	0.2982	0.5263	0.3704	0.2667	0.2857
gpt-3.5	self-heuristic	probe-test	**1.0000**	*0.7368*	0.7500	**1.0000**	**1.0000**	0.9231	0.8095	0.5484	0.8421	0.6400	0.9333	0.5000
gpt-4	self-heuristic	probe-test	**1.0000**	*0.7368*	**1.0000**	**1.0000**	0.9167	0.9167	0.9048	0.6786	0.8947	0.7083	0.8667	0.7647
gpt-4	self-heuristic	test	0.9848	0.7738	0.9063	0.9355	0.9167	0.8333	0.7961	0.7321	0.8941	0.7917	0.7714	0.8852

The evaluation result on vulnerability severity evaluation. AV = Attack Vector. AC = Attack Complexity. PR = Privileges Required. UI = User Interaction. R = Recall. P = Precision.

Vulnerability Severity Evaluation

- ChatGPT's performance is slightly inferior to the SOTA approach.
- Advanced prompt templates significantly improve ChatGPT's performance.
- **leveraging ChatGPT in a self-heuristic way to improve its performance for challenging tasks is an interesting future research direction.**



the process of extracting knowledge by ChatGPT

1 *Network: Functions that involve network communication, socket handling, or network device management. Examples: sock_register, udp4_hwcsum, ...*
2 *Adjacent Network: Functions that involve wireless communication, NFC, or Bluetooth. Examples: nfc_start_poll, lib80211_wep_encrypt, ...*
3 *Physical: Functions that involve hardware interaction, device management, or USB handling. Examples: usb_release_dev, snd_card_free, ...*
4 *Not Related: Functions that do not involve any network, adjacent network, or physical interactions, and are related to memory management, page allocation, or other internal system operations. Examples: do_set_mempolicy, do_page_mkwrite, ...*

The knowledge summarized by ChatGPT.

Vulnerability Severity Evaluation

- ChatGPT's performance is slightly inferior to the SOTA approach.
- Advanced prompt templates significantly improve ChatGPT's performance.
- **leveraging ChatGPT in a self-heuristic way to improve its performance for challenging tasks is an interesting future research direction.**

1. *Network: Functions that involve network communication, socket handling, or network device management. Examples: sock_register, udp4_hwcsum, ...*
2. *Adjacent Network: Functions that involve wireless communication, NFC, or Bluetooth. Examples: nfc_start_poll, lib80211_wep_encrypt, ...*
3. *Physical: Functions that involve hardware interaction, device management, or USB handling. Examples: usb_release_dev, snd_card_free, ...*
4. *Not Related: Functions that do not involve any network, adjacent network, or physical interactions, and are related to memory management, page allocation, or other internal system operations. Examples: do_set_mempolicy, do_page_mkwrite, ...*

The knowledge summarized by ChatGPT.

Approach	Prompt	Dataset	AV						AC		PR		UI	
			Network		Adjacent		Physical		High		High		Required	
			R	P	R	P	R	P	R	P	R	P	R	P
DiffCVSS [48]	-	test	0.9242	**0.9384**	0.8750	0.9333	0.8852	0.9153	**0.9151**	**0.9238**	**0.9452**	0.9324	**0.9167**	**0.9296**
gpt-3.5	0-shot	probe-test	0.7143	0.5556	0	N/A	0	N/A	0.4286	0.6923	0.3684	0.5000	0	N/A
gpt-3.5	1-shot	probe-test	**1.0000**	0.2206	0	N/A	0.0909	**1.0000**	0.2857	1.0000	0.1053	**1.0000**	0.2667	0.3077
gpt-3.5	few-shot	probe-test	**1.0000**	0.4285	0.4444	0.6667	0.3636	0.4444	0.6190	0.6842	0.2632	0.3333	0.6667	0.2703
gpt-3.5	general-info	probe-test	0.7857	0.4783	0	N/A	0.1667	0.5000	0.8095	0.3269	0.7368	0.2188	0.4000	0.3000
gpt-3.5	expertise	probe-test	0.8571	0.5714	0.5000	0.6667	0.0833	**1.0000**	0.8095	0.2982	0.5263	0.3704	0.2667	0.2857
gpt-3.5	self-heuristic	probe-test	**1.0000**	*0.7368*	0.7500	**1.0000**	**1.0000**	0.9231	0.8095	0.5484	0.8421	0.6400	0.9333	0.5000
gpt-4	self-heuristic	probe-test	**1.0000**	*0.7368*	**1.0000**	**1.0000**	0.9167	0.9167	0.9048	0.6786	0.8947	0.7083	0.8667	0.7647
gpt-4	self-heuristic	test	0.9848	0.7738	0.9063	0.9355	0.9167	0.8333	0.7961	0.7321	0.8941	0.7917	0.7714	0.8852

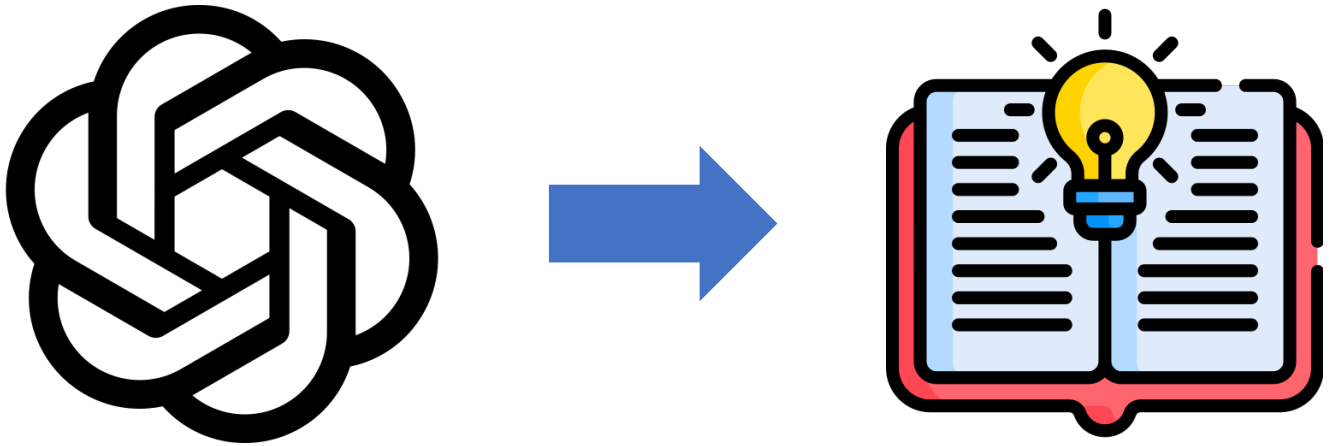
The evaluation result on vulnerability severity evaluation. AV = Attack Vector. AC = Attack Complexity. PR = Privileges Required. UI = User Interaction. R = Recall. P = Precision.

Vulnerability Severity Evaluation

- ChatGPT's performance is slightly inferior to the SOTA approach.
- Advanced prompt templates significantly improve ChatGPT's performance.
- ***leveraging ChatGPT in a self-heuristic way to improve its performance for challenging tasks is an interesting future research direction.***

1. *Network: Functions that involve network communication, socket handling, or network device management. Examples: sock_register, udp4_hwcsum, ...*
2. *Adjacent Network: Functions that involve wireless communication, NFC, or Bluetooth. Examples: nfc_start_poll, lib80211_wep_encrypt, ...*
3. *Physical: Functions that involve hardware interaction, device management, or USB handling. Examples: usb_release_dev, snd_card_free, ...*
4. *Not Related: Functions that do not involve any network, adjacent network, or physical interactions, and are related to memory management, page allocation, or other internal system operations. Examples: do_set_mempolicy, do_page_mkwrite, ...*

The knowledge summarized by ChatGPT.



Vulnerability Repair

fix the vulnerable code snippet

- **ChatGPT can fix 10/12 vulnerabilities with a high valid repair rate.**

Approach	Prompt	Dataset	# Gen	# Vld	# Vuln	# Fn	# Fn & Vuln	# Fn & Safe	% Vld Repair	# Fixed
ExtractFix [24]	-	test	-	-	-	-	-	-	-	**10**
LLMset [37]	0-shot	test	3,300	674	234	388	252	159	23.6	5
LLMset [37]	expertise	test	3,300	1254	726	926	705	221	17.6	8
gpt-3.5	0-shot	probe-test	350	329	23	166	5	161	48.9	5
gpt-3.5	1-shot	probe-test	350	326	8	176	7	169	51.8	5
gpt-3.5	few-shot	probe-test	350	337	7	145	4	141	41.8	6
gpt-3.5	general-info	probe-test	350	204	4	118	4	114	55.9	4
gpt-3.5 (Orig.)	expertise	probe-test	350	138	40	78	39	39	28.3	5
gpt-3.5	expertise	probe-test	350	259	40	227	39	188	72.6	7
gpt-3.5	self-heuristic	probe-test	350	253	7	153	7	146	57.7	6
gpt-4	expertise	probe-test	350	292	2	290	2	288	**98.6**	7
gpt-4	expertise	test	600	377	20	370	20	350	92.8	**10**

The evaluation result on vulnerability repair. Gen = Generated. Vld = compilable. Vuln = Vulnerable. Fn = Functional. Safe = Not Vulnerable. Fixed = Fixed Vulnerabilities.

Vulnerability Repair

- **Failed cases: insufficient vulnerability-related context provided**

  In particular, EF08 involves a shift range error, but the context does not include information about the shift variable, leading ChatGPT to "guess" the variable name and thereby failing to generate the correct repair code.

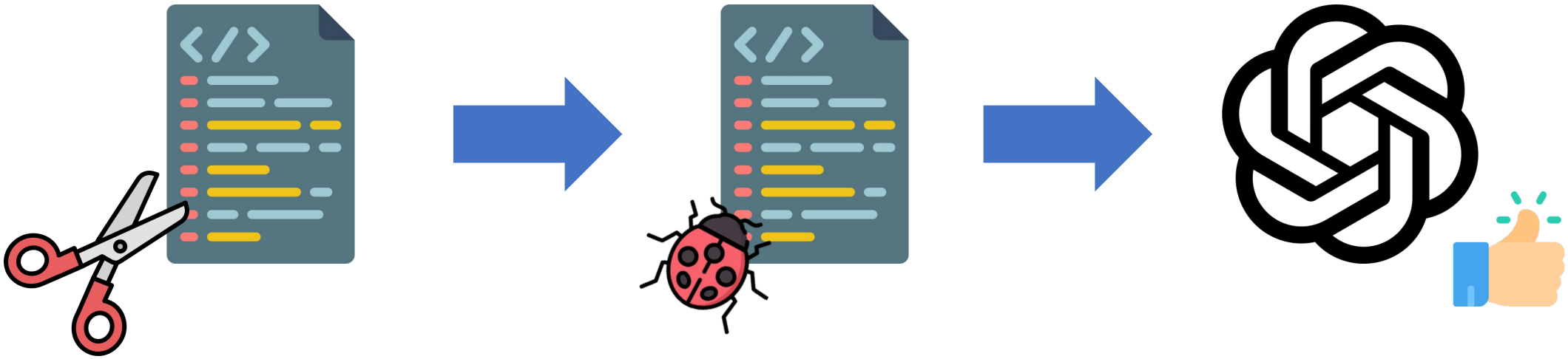
Approach	Prompt	EF01 CVE-2016-5321	EF02_01 CVE-2014-8128	EF02_02 CVE-2014-8128	EF07 CVE-2016-10094	EF08 CVE-2017-7601	EF09 CVE-2016-3623	EF10 CVE-2017-7595	EF15 CVE-2016-1838	EF17 CVE-2012-5134	EF18 CVE-2017-5969	EF20 CVE-2018-19664	EF22 CVE-2012-2806
ExtractFix [24]	-	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓
LLMset [37]	0-shot	33/49	0/2	0/81	-	42/135	4/4	4/65	-	53/58	0/13	0/198	0/69
LLMset [37]	expertise	14/117	23/124	0/205	-	46/78	96/190	11/37	3/98	24/33	0/120	4/171	0/81
gpt-4	expertise	31/38	50/50	-	4/6	0/5	50/50	32/34	2/4	47/50	37/43	47/47	50/50
gpt-4/LLMs/EF	-	✓/✓/✓	✓/✓/✓	✗/✗/✗	✓/✗/✓	✗/✓/✓	✓/✓/✓	✓/✓/✓	✓/✓/✓	✓/✓/✓	✓/✗/✓	✓/✓/✗	✓/✗/✓

The evaluation result on vulnerability repair for each CVE. The results are presented as '# Fn & Safe'/'# Vld'. Orig = Using the original code grafting method designed for LLMset.

Vulnerability Repair

- **Failed cases: insufficient vulnerability-related context provided**

  To improve ChatGPT's vulnerability repairing capability in real-world applications, it could be effective to apply more advanced program slicing methods to provide specific vulnerability-related context.



Patch Correctness Assessment

whether a patch correctly fixes a bug

- **ChatGPT performs comparably to the SOTA approaches.**

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Invalidator [31]	-	test	0.813	0.900	0.789	0.540	0.675	0.844
gpt-3.5	0-shot	probe-test	0.568	0.758	0.415	0.510	0.610	0.586
gpt-3.5	1-shot	probe-test	0.581	**0.970**	0.268	0.516	0.674	0.619
gpt-3.5	few-shot	probe-test	0.595	0.576	0.610	0.543	0.559	0.593
gpt-3.5	general-info	probe-test	0.608	0.576	0.634	0.559	0.567	0.605
gpt-3.5	expertise	probe-test	0.621	0.545	0.683	0.581	0.563	0.614
gpt-3.5	self-heuristic	probe-test	0.730	0.758	0.707	0.676	0.714	0.732
gpt-4	self-heuristic	probe-test	0.757	0.667	**0.829**	**0.759**	0.710	0.748
gpt-4	self-heuristic	test	**0.849**	0.933	0.826	0.596	**0.727**	**0.880**

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Panther [44]	-	test	0.745	0.811	0.670	0.738	0.773	0.818
gpt-3.5	0-shot	probe-test	0.710	0.963	0.381	0.669	0.789	0.672
gpt-3.5	1-shot	probe-test	0.642	0.972	0.214	0.616	0.754	0.593
gpt-3.5	few-shot	probe-test	0.653	**0.981**	0.226	0.622	0.762	0.603
gpt-3.5	general-info	probe-test	0.720	0.844	0.560	0.713	0.773	0.702
gpt-3.5	expertise	probe-test	0.715	0.771	0.643	0.737	0.753	0.707
gpt-3.5	self-heuristic	probe-test	0.730	0.844	0.583	0.724	0.780	0.714
gpt-4	self-heuristic	probe-test	**0.870**	0.899	**0.833**	**0.875**	**0.887**	**0.866**
gpt-4	self-heuristic	test	0.813	0.829	0.794	0.821	0.825	0.811

The evaluation result on patch correctness assessment.

Patch Correctness Assessment

- **ChatGPT performs comparably to the SOTA approaches.**

  Invalidator and Panther: only contain the code of patches.

  Quatrain: only contains the description of patches.

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Quatrain [46]	-	test	0.775	0.786	0.773	0.371	0.504	**0.858**
gpt-3.5	0-shot	probe-test	0.617	0.577	0.625	0.246	0.345	0.601
gpt-3.5	1-shot	probe-test	0.682	0.479	0.725	0.270	0.345	0.602
gpt-3.5	few-shot	probe-test	0.720	0.493	0.768	0.311	0.381	0.631
gpt-3.5	general-info	probe-test	0.797	0.359	0.889	0.408	0.382	0.624
gpt-3.5	expertise	probe-test	0.761	0.479	0.821	0.362	0.412	0.650
gpt-3.5	self-heuristic	probe-test	**0.837**	0.366	**0.937**	**0.553**	0.441	0.652
gpt-4	self-heuristic	probe-test	0.789	0.275	0.898	0.364	0.313	0.587
gpt-3.5	desc-code	probe-test	0.725	0.697	0.731	0.355	0.470	0.714
gpt-3.5	code-only	probe-test	0.564	0.817	0.510	0.261	0.396	0.663
gpt-4	desc-code	probe-test	0.700	**0.915**	0.655	0.360	0.517	0.785
gpt-4	code-only	probe-test	0.816	0.901	0.798	0.487	**0.632**	0.850
gpt-4	code-only	test	0.819	0.868	0.811	0.439	0.583	0.840

The evaluation result on patch correctness assessment (compared with Quatrain).

Patch Correctness Assessment

- **The code of patches plays an important role in this task.**

  We manually collect the corresponding code for each patch and provide the code and description simultaneously in the desc-code prompt.

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Quatrain [46]	-	test	0.775	0.786	0.773	0.371	0.504	**0.858**
gpt-3.5	0-shot	probe-test	0.617	0.577	0.625	0.246	0.345	0.601
gpt-3.5	1-shot	probe-test	0.682	0.479	0.725	0.270	0.345	0.602
gpt-3.5	few-shot	probe-test	0.720	0.493	0.768	0.311	0.381	0.631
gpt-3.5	general-info	probe-test	0.797	0.359	0.889	0.408	0.382	0.624
gpt-3.5	expertise	probe-test	0.761	0.479	0.821	0.362	0.412	0.650
gpt-3.5	self-heuristic	probe-test	**0.837**	0.366	**0.937**	**0.553**	0.441	0.652
gpt-4	self-heuristic	probe-test	0.789	0.275	0.898	0.364	0.313	0.587
gpt-3.5	desc-code	probe-test	0.725	0.697	0.731	0.355	0.470	0.714
gpt-3.5	code-only	probe-test	0.564	0.817	0.510	0.261	0.396	0.663
gpt-4	desc-code	probe-test	0.700	**0.915**	0.655	0.360	0.517	0.785
gpt-4	code-only	probe-test	0.816	0.901	0.798	0.487	**0.632**	0.850
gpt-4	code-only	test	0.819	0.868	0.811	0.439	0.583	0.840

The evaluation result on patch correctness assessment (compared with Quatrain).

Patch Correctness Assessment

- **The code of patches plays an important role in this task.**

  We manually collect the corresponding code for each patch and provide both the code and description in the desc-code prompt.

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Quatrain [46]	-	test	0.775	0.786	0.773	0.371	0.504	**0.858**
gpt-3.5	0-shot	probe-test	0.617	0.577	0.625	0.246	0.345	0.601
gpt-3.5	1-shot	probe-test	0.682	0.479	0.725	0.270	0.345	0.602
gpt-3.5	few-shot	probe-test	0.720	0.493	0.768	0.311	0.381	0.631
gpt-3.5	general-info	probe-test	0.797	0.359	0.889	0.408	0.382	0.624
gpt-3.5	expertise	probe-test	0.761	0.479	0.821	0.362	0.412	0.650
gpt-3.5	self-heuristic	probe-test	**0.837**	0.366	**0.937**	**0.553**	0.441	0.652
gpt-4	self-heuristic	probe-test	0.789	0.275	0.898	0.364	0.313	0.587
gpt-3.5	desc-code	probe-test	0.725	0.697	0.731	0.355	0.470	0.714
gpt-3.5	code-only	probe-test	0.564	0.817	0.510	0.261	0.396	0.663
gpt-4	desc-code	probe-test	0.700	**0.915**	0.655	0.360	0.517	0.785
gpt-4	code-only	probe-test	0.816	0.901	0.798	0.487	**0.632**	0.850
gpt-4	code-only	test	0.819	0.868	0.811	0.439	0.583	0.840

The evaluation result on patch correctness assessment (compared with Quatrain).

Patch Correctness Assessment

- The code of patches plays an important role in this task.
- **Providing patch descriptions even negatively affects this task.**

  When the code and description are provided simultaneously, ChatGPT tends to analyze whether the code changes "match" the description rather than the correctness of the patch.

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Quatrain [46]	-	test	0.775	0.786	0.773	0.371	0.504	**0.858**
gpt-3.5	0-shot	probe-test	0.617	0.577	0.625	0.246	0.345	0.601
gpt-3.5	1-shot	probe-test	0.682	0.479	0.725	0.270	0.345	0.602
gpt-3.5	few-shot	probe-test	0.720	0.493	0.768	0.311	0.381	0.631
gpt-3.5	general-info	probe-test	0.797	0.359	0.889	0.408	0.382	0.624
gpt-3.5	expertise	probe-test	0.761	0.479	0.821	0.362	0.412	0.650
gpt-3.5	self-heuristic	probe-test	**0.837**	0.366	**0.937**	**0.553**	0.441	0.652
gpt-4	self-heuristic	probe-test	0.789	0.275	0.898	0.364	0.313	0.587
gpt-3.5	desc-code	probe-test	0.725	0.697	0.731	0.355	0.470	0.714
gpt-3.5	code-only	probe-test	0.564	0.817	0.510	0.261	0.396	0.663
gpt-4	desc-code	probe-test	0.700	**0.915**	0.655	0.360	0.517	0.785
gpt-4	code-only	probe-test	0.816	0.901	0.798	0.487	**0.632**	0.850
gpt-4	code-only	test	0.819	0.868	0.811	0.439	0.583	0.840

The evaluation result on patch correctness assessment (compared with Quatrain).

Patch Correctness Assessment

- The code of patches plays an important role in this task.
- **Providing patch descriptions even negatively affects this task.**

More information is not always better. Guiding ChatGPT to leverage the information in the prompt in a suitable way is an interesting research direction.

Approach	Prompt	Dataset	Accuracy	+Recall	-Recall	Precision	F1	AUC
Quatrain [46]	-	test	0.775	0.786	0.773	0.371	0.504	**0.858**
gpt-3.5	0-shot	probe-test	0.617	0.577	0.625	0.246	0.345	0.601
gpt-3.5	1-shot	probe-test	0.682	0.479	0.725	0.270	0.345	0.602
gpt-3.5	few-shot	probe-test	0.720	0.493	0.768	0.311	0.381	0.631
gpt-3.5	general-info	probe-test	0.797	0.359	0.889	0.408	0.382	0.624
gpt-3.5	expertise	probe-test	0.761	0.479	0.821	0.362	0.412	0.650
gpt-3.5	self-heuristic	probe-test	**0.837**	0.366	**0.937**	**0.553**	0.441	0.652
gpt-4	self-heuristic	probe-test	0.789	0.275	0.898	0.364	0.313	0.587
gpt-3.5	desc-code	probe-test	0.725	0.697	0.731	0.355	0.470	0.714
gpt-3.5	code-only	probe-test	0.564	0.817	0.510	0.261	0.396	0.663
gpt-4	desc-code	probe-test	0.700	**0.915**	0.655	0.360	0.517	0.785
gpt-4	code-only	probe-test	0.816	0.901	0.798	0.487	**0.632**	0.850
gpt-4	code-only	test	0.819	0.868	0.811	0.439	0.583	0.840

The evaluation result on patch correctness assessment (compared with Quatrain).

Stable Patch Classification

whether a given patch is a stable patch

Approach	Prompt	Dataset	ACC	P	R	F1	AUC
PatchNet [25]	-	test	**0.862**	**0.839**	0.907	**0.871**	**0.860**
gpt-3.5	0-shot	probe-test	0.566	0.564	0.995	0.720	0.508
gpt-3.5	1-shot	probe-test	0.555	0.558	0.986	0.713	0.496
gpt-3.5	few-shot	probe-test	0.557	0.561	0.964	0.709	0.501
gpt-3.5	general-info	probe-test	0.568	0.565	**0.996**	0.721	0.510
gpt-3.5	expertise	probe-test	0.762	0.761	0.837	0.798	0.752
gpt-3.5	self-heuristic	probe-test	0.646	0.631	0.884	0.737	0.614
gpt-4	expertise	probe-test	0.736	0.694	0.945	0.800	0.708
gpt-4	expertise	test	0.733	0.679	0.950	0.792	0.716

The evaluation result on stable patch classification.
ACC = Accuracy. P = Precision. R = Recall.

- **ChatGPT performs slightly worse than the SOTA.**

Stable Patch Classification

Approach	Prompt	Dataset	ACC	P	R	F1	AUC
PatchNet [25]	-	test	**0.862**	**0.839**	0.907	**0.871**	**0.860**
gpt-3.5	0-shot	probe-test	0.566	0.564	0.995	0.720	0.508
gpt-3.5	1-shot	probe-test	0.555	0.558	0.986	0.713	0.496
gpt-3.5	few-shot	probe-test	0.557	0.561	0.964	0.709	0.501
gpt-3.5	general-info	probe-test	0.568	0.565	**0.996**	0.721	0.510
gpt-3.5	expertise	probe-test	0.762	0.761	0.837	0.798	0.752
gpt-3.5	self-heuristic	probe-test	0.646	0.631	0.884	0.737	0.614
gpt-4	expertise	probe-test	0.736	0.694	0.945	0.800	0.708
gpt-4	expertise	test	0.733	0.679	0.950	0.792	0.716

The evaluation result on stable patch classification.
ACC = Accuracy. P = Precision. R = Recall.

- ChatGPT performs slightly worse than the SOTA.
- **When using the 0-shot and 1-shot prompts, ChatGPT tends to report all patches as stable ones.**

  ChatGPT does not understand what a stable patch is. It tends to report all patches as stable ones. Thus, the precision scores are close to 0.5 while recall scores are close to 1.

Stable Patch Classification

Approach	Prompt	Dataset	ACC	P	R	F1	AUC
PatchNet [25]	-	test	**0.862**	**0.839**	0.907	**0.871**	**0.860**
gpt-3.5	0-shot	probe-test	0.566	0.564	0.995	0.720	0.508
gpt-3.5	1-shot	probe-test	0.555	0.558	0.986	0.713	0.496
gpt-3.5	few-shot	probe-test	0.557	0.561	0.964	0.709	0.501
gpt-3.5	general-info	probe-test	0.568	0.565	**0.996**	0.721	0.510
gpt-3.5	expertise	probe-test	0.762	0.761	0.837	0.798	0.752
gpt-3.5	self-heuristic	probe-test	0.646	0.631	0.884	0.737	0.614
gpt-4	expertise	probe-test	0.736	0.694	0.945	0.800	0.708
gpt-4	expertise	test	0.733	0.679	0.950	0.792	0.716

The evaluation result on stable patch classification.
ACC = Accuracy. P = Precision. R = Recall.

- ChatGPT performs slightly worse than the SOTA.
- When using the 0-shot and 1-shot prompts, ChatGPT tends to report all patches as stable ones.
- **Providing the definition of stable patch significantly improves ChatGPT's performance.**

> *"fixing a problem that causes a build error, an oops, a hang, data corruption, a real security issue, or some 'oh, that's not good' issue"*

Summary

- We conduct the first large-scale evaluation to explore the capabilities of ChatGPT on vulnerability management.
- We compare ChatGPT with 11 SOTA approaches on 6 vulnerability management tasks by using a large-scale dataset.
- Our findings demonstrate that ChatGPT has excellent capabilities when processing several vulnerability management tasks.
- We also reveal the difficulties ChatGPT encountered and shed light on future research to explore better ways to leverage ChatGPT in vulnerability management tasks.

Thank you for listening!

浙江大学
Zhejiang University

杭州電子科技大学
HANGZHOU DIANZI UNIVERSITY

UNIVERSITY OF MINNESOTA
Driven to Discover℠

Exploring ChatGPT's Capabilities on Vulnerability Management

Junming Liu
jmliu@zju.edu.cn

Prompt templates & code: https://github.com/Jamrot/ChatGPT-Vulnerability-Management