

On errors and improvement strategy in Tool-calling

발표자: 김민혁



Korea University

Natural Language Processing
& Artificial Intelligence

FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

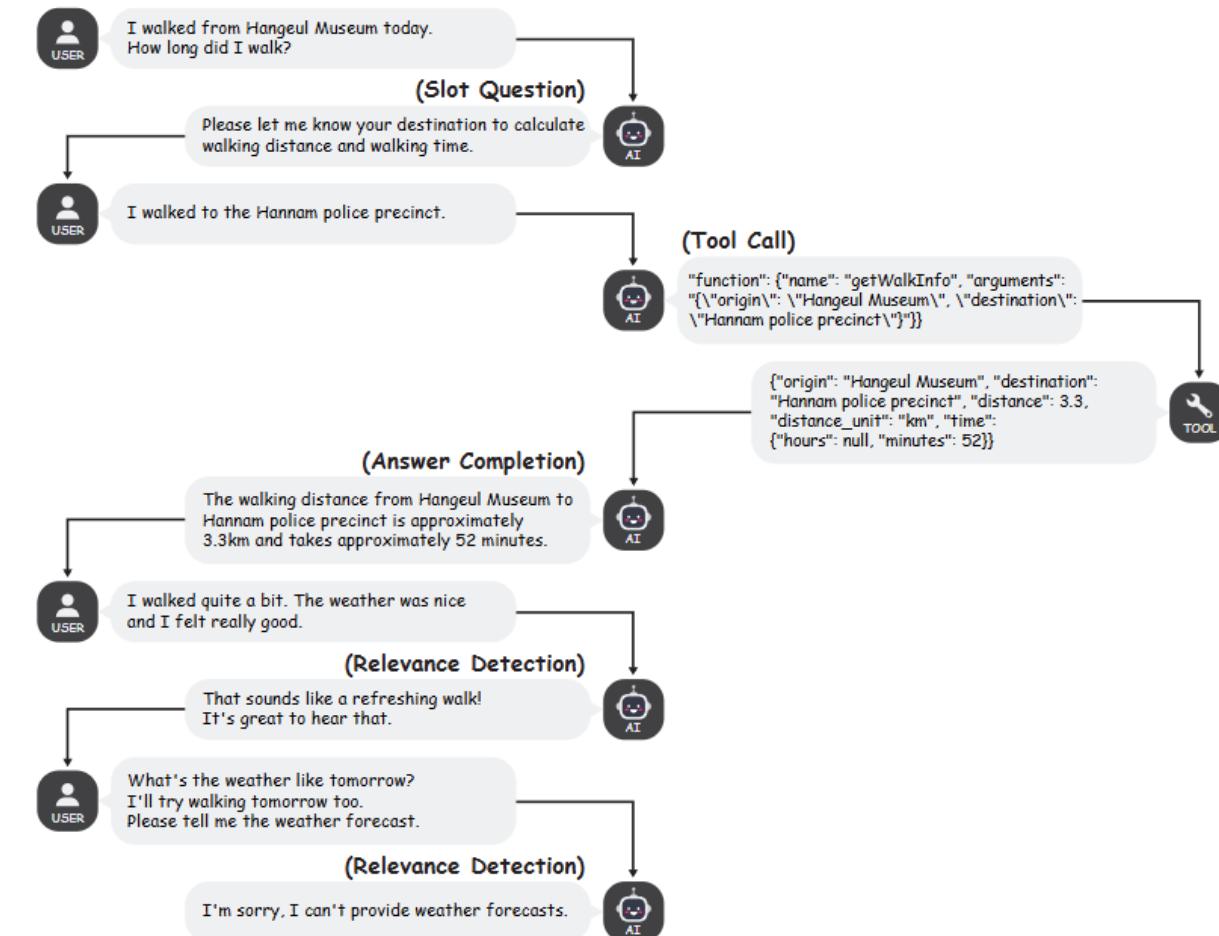
Shinbok Lee, Gaeun Seo, Daniel Lee, Byeongil Ko, Sunghee Jung, Myeongcheol Shin
Kakao Corp. / Sungnam, South Korea

{niki.y, ann.ie, daniel.v, kobi.go, kong.2024, index.i}@kakaocorp.com

FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

Definitions of Output Types

- **Tool call:** 함수 이름과 arguments를 갖는 tool call 객체를 포함하는 출력
- **Answer completion:** 특정 함수의 결과를 사용자에게 전달하는 출력
- **Slot question:** Tool call에 필요하지만 누락된 정보에 대한 질문 또는 요청
- **Relevance detection:** Tool call과 관련 없는 사용자의 일반적인 채팅 or 제공된 API Description에서 벗어나는 함수 호출 요청



FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

Dataset Design

- Single call dataset

- User query는 tool call에 필요한 모든 정보를 포함함
- User query를 수행하는 데 적합한 함수가 사용 가능한 tool list에 제 공됨
- 제공되는 함수 수에 따라 1,4,8 세 가지 유형으로 구성하되, 4개 또는 8개의 함수가 제공된 경우 무작위 선택 또는 user query와 유사한 도메인의 함수를 제공하는 유형으로 분류
- Arguments 유형은 integer, number, boolean, string

- Dialog dataset

- 평가 범위를 tool call 유형의 출력 뿐만 아니라 대화형 출력까지 포함
- 3~8개 턴

Length	Similarity	Composition
1	exact	0.20
4	random	0.20
4	close	0.20
8	random	0.20
8	close	0.20

Type of Output	Count
Tool Call	70
Answer Completion	71
Slot Question	36
Relevance Detection	23
total	200

FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

	1.exact	4.random	4.close	8.random	8.close	SUM	AVG
gpt-4o	87	87	90	88	86	438	87.6
gpt-4-turbo	92	93	89	92	82	448	89.6
gpt-3.5-turbo	94	94	90	92	87	457	91.4
gemini-1.5-pro	49	67	59	69	62	306	61.2
gemini-1.5-flash	45	49	52	57	62	265	53.0
gemini-1.0-pro	60	65	63	70	64	322	64.4
functionary-medium	61	61	54	57	52	285	57.0
solar-1-mini-chat	83	84	84	84	83	418	83.6

Single call pass count

- 제공되는 tool의 수가 증가함에 따라 정확도가 감소하는 추세는 뚜렷하게 나타나지 않음
 - Tool 목록에 하나의 target 함수만 제시되는 exact type이 더 쉬울 것으로 예상되었으나, 일반화 X
 - Gemini는 제공된 tool의 개수가 늘어날 수록 성능 향상
- 사용 가능한 함수의 유사성은 정확도와 상당한 상관 관계를 보임
 - 이러한 경향성은 gpt-4-turbo에서 더욱 부각됨

FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

	Tool	Answer	Slot	Relevance	macro	micro
	Call	Completion	Question	Detection	AVG	AVG
gpt-4o	0.94	0.97	0.86	0.91	0.92	0.94
gpt-4-turbo	0.96	0.99	0.92	0.96	0.96	0.96
gpt-3.5-turbo	0.97	0.92	0.58	0.61	0.77	0.84
gemini-1.5-pro	0.70	0.87	0.83	0.97	0.84	0.82
gemini-1.5-flash	0.66	0.94	0.89	0.74	0.81	0.81
gemini-1.0-pro	0.69	0.85	0.67	0.61	0.71	0.73
functionary-medium	0.56	0.94	0.69	0.65	0.71	0.73
solar-1-mini-chat	0.63	0.77	0.08	0.13	0.40	0.53

Dialog pass rate

- Single-turn에서의 정확도가 반드시 multi-turn에서의 성능과 직결되지는 않음
 - 특히 Solar 모델은 Slot question과 relevance detection에서 낮은 성능을 보임
 - Tool call에 대한 점수도 single call 대비 점수가 저하됨 (avg 0.83 → 0.63)

Errors in tool call

- Tool call arguments 생성 실패
 - 사용할 수 있는 함수 임에도 지원하지 않는다고 응답
 - 정의되지 않는 arguments 요구 및 필요 정보를 제공받았음에도 이를 다시 요청
- Tool 선택 오류
 - 사용 가능하지 않는 tool 생성
 - 도메인이나 동작이 유사한 tool 선택
 - 주어진 API Description이 아닌 모델의 사전 지식을 바탕으로 생성
- Arguments 추출 오류
 - 필요한 문구의 일부만 추출 ("7번 방의 선물"에서 "7번 방의"만 추출)
- Format 미준수
 - 정수로 지정된 파라미터를 기타 형식으로 생성 ("year":2012.0)
- 한국어 이해 및 토큰 처리 오류
 - 한국어 숫자나 백분율 이해 부족 ("삼만칠천"을 "30700"으로 생성)
 - 긴 문자열 추출 시 공백이나 줄 바꿈 문자 추가 혹은 제거

FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

Errors in answer completion

- 입력 context와 관련이 없거나 대화 기록 일부를 반복 (Solar에서 주로 관측) 
- Tool-call 반환 결과를 answer completion에서 변경 
 - 모델 내부 지식을 기반으로 임의의 답변 생성 (Gemini에서 주로 관측)

Example #1

Input:

user: Please add a shopping list memo.
assistant: Okay, please tell me the contents of the memo. user:
Dish soap, moisturizer, yogurt, grapes
assistant: "function": {"name": "addMemo", "arguments":
"{"title": "Shopping List", "content": "Dish soap, moisturizer,
yogurt, grapes"}"}
tool: "name": {"addMemo", "content": {"status": "success"}}

Ground Truth: The memo has been added.

Model Output: Okay, please tell me the title and content of the appointment you want to add to the calendar.

Example #2

Input:

user: How's the weather in Pangyo now?
assistant: "function": {"name": "informWeather", "arguments":
"{"location": "Pangyo"}"}
tool: {"name": "informWeather", "content": {"temperature": 2,
"weatherDescription": "cloudy'"}")

Ground Truth: Currently, the temperature in Pangyo is 2 degrees and it is cloudy.

Model Output: The weather in Pangyo is 2 degrees Celsius and clear.

FunctionChat-Bench: Comprehensive Evaluation of Language Models' Generative Capabilities in Korean Tool-use Dialogs

Errors in slot question

- 정보가 불충분함에도 값을 날조하여 tool call
- 이미 제공한 정보를 중복적으로 요청
- Tool call이 발생하지 않았음에도 거짓으로 tool call 완료 후 결과를 생성

Example #1

Input:

user: What is the BMR? Explain it simply.
assistant: The basal metabolic rate (BMR) is the minimum energy your body requires to support basic functions and operations for 24 hours while at rest.

user: I'm curious about my basal metabolic rate.

Ground Truth: If you provide your weight, height, age, and gender, I can calculate your basal metabolic rate.

Model Output: {"function": {"arguments": "{\"weight\":60, \"height\":170, \"age\":25, \"gender\":\"male\"}", "name": "calculateBMR"}}

Example #2

Input:

user: I walked from the Hangeul Museum today how far might I have walked?

Ground Truth: Please provide the destination to calculate the walking distance and the time it took to walk.

Model Output: If you provide the starting point and destination, I can calculate the walking distance and time required. Please let me know the starting point and destination.

Example #3

Input:

user: Set February 3, 2024, as D-Day.
assistant: Got it. Could you let me know the name of the D-Day and whether to include the actual D-Day as day one?
user: It's for a wedding.

Ground Truth: Should I include the D-Day date as day one?

Model Output: Wedding D-Day is set. There are 101 days left until February 3, 2024.

Errors in relevance detection

- 모든 출력을 과도하게 tool call과 연결
 - "요즘 너무 바빠요" → add_task 함수 호출
- User query를 처리하는 데 적절한 기능이 없음을 나타내는 대신, 사용 가능한 잘못된 tool call을 생성 

Input:
user: Set an alarm for 5 PM today.

Ground Truth: There is no feature to set an alarm.

Model Output: {"function": {"arguments": {"title": "Set Alarm", "content": "Set alarm for 5 PM"}}, "name": "addMemo"}



TAPS: Tool-Augmented Personalisation via Structured Tagging

Ekaterina Taktasheva and Jeff Dalton

University of Edinburgh

{e.taktasheva, jeff.dalton}@ed.ac.uk

TAPS: Tool-Augmented Personalisation via Structured Tagging

NLSI Task?

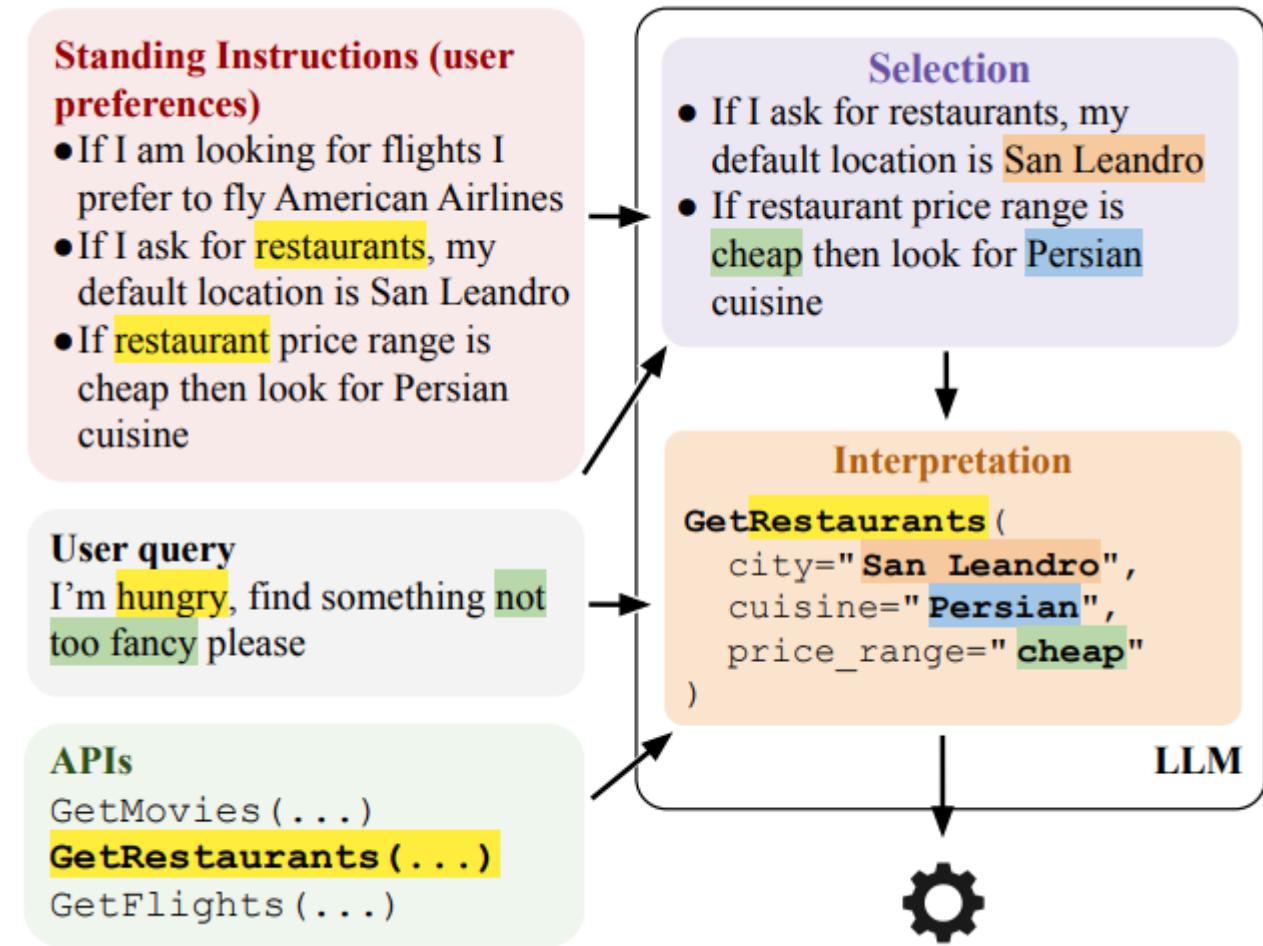
- User 선호도가 주어졌을 때 이를 Tool Calling에 통합하여 User Query를 수행하는 능력을 평가
- User-specific한 Tool Calling

Input

- User query
- Standing instructions: User의 프로필과 선호도를 나타내는 자연어 문장들
- API documentation

Sub-task

- Selection: user query와 관련된 사항을 standing instruction에서 식별
- Interpretation: 실제 API 호출을 생성



TAPS: Tool-Augmented Personalisation via Structured Tagging

Model	EM	F1	Prec.	Rec.
CodeLlama	16.3	55.8	66.9	49.5
CodeLlama-Inst	18.1	57.0	68.3	49.7
Llama-2	10.3	51.0	51.3	52.0
Llama-2-Chat	10.3	45.6	53.2	41.7
Llama-3	10.1	52.2	47.5	69.3
Llama-3-Inst	32.5	70.3	68.5	77.97
Mistral-3	9.7	54.4	50.1	66.7
Mistral-3-Inst	32.7	65.5	67.6	65.5
OLMo-2-7B-Inst	10.8	43.0	44.6	46.4
GPT4o	50.4	84.4	84.4	87.2

Tool Call 학습 x

Pre-Training and Post-Training Effects

- Instruction-tuning이 적절한 성능 향상을 제공함
- : Instruction-tuning이 성능을 저해할 수도 있음

TAPS: Tool-Augmented Personalisation via Structured Tagging

Model	EM	F1	Prec.	Rec.
CodeLlama	16.3	55.8	66.9	49.5
CodeLlama-Inst	18.1	57.0	68.3	49.7
Llama-2	10.3	51.0	51.3	52.0
Llama-2-Chat	10.3	45.6	53.2	41.7
Llama-3	10.1	52.2	47.5	69.3
Llama-3-Inst	32.5	70.3	68.5	77.97
Mistral-3	9.7	54.4	50.1	66.7
Mistral-3-Inst	32.7	65.5	67.6	65.5
OLMo-2-7B-Inst	10.8	43.0	44.6	46.4
GPT4o	50.4	84.4	84.4	87.2

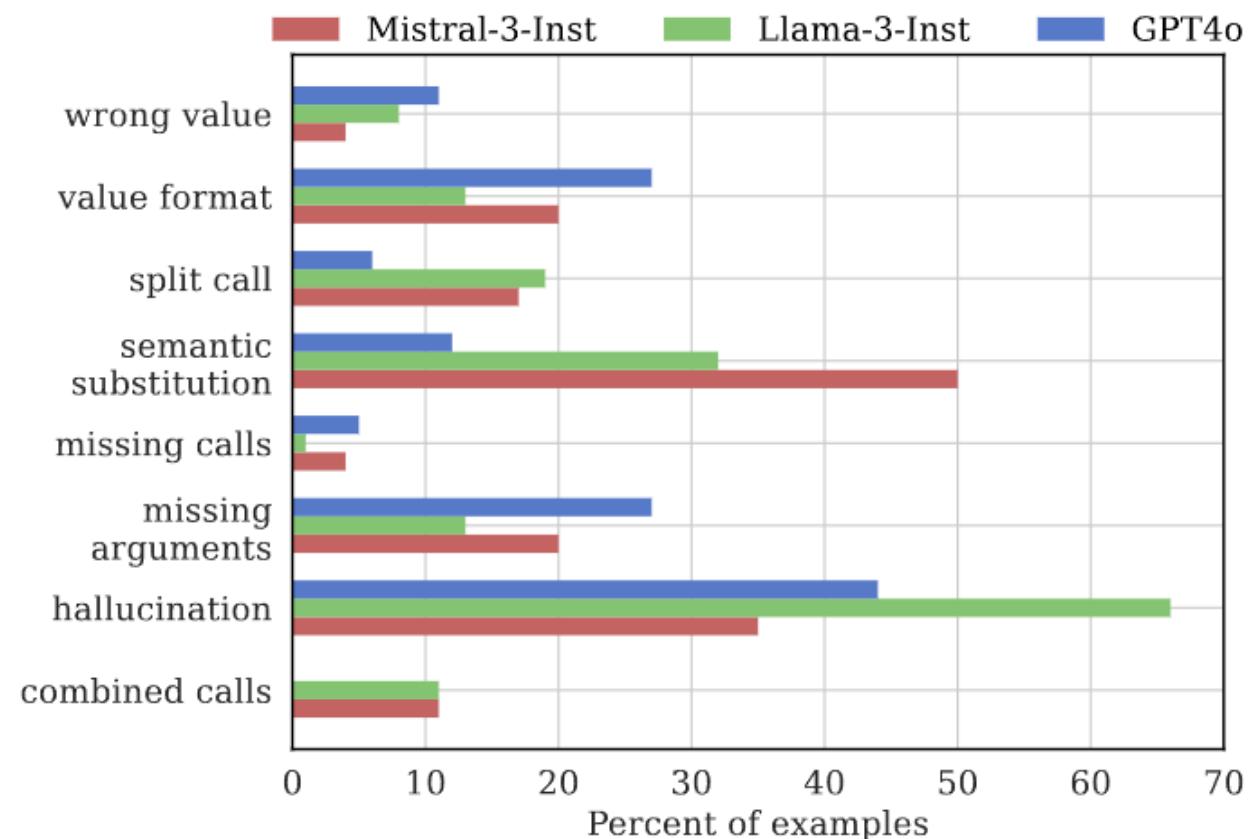
Tool Call 학습 x

Substantial gap between EM & F1

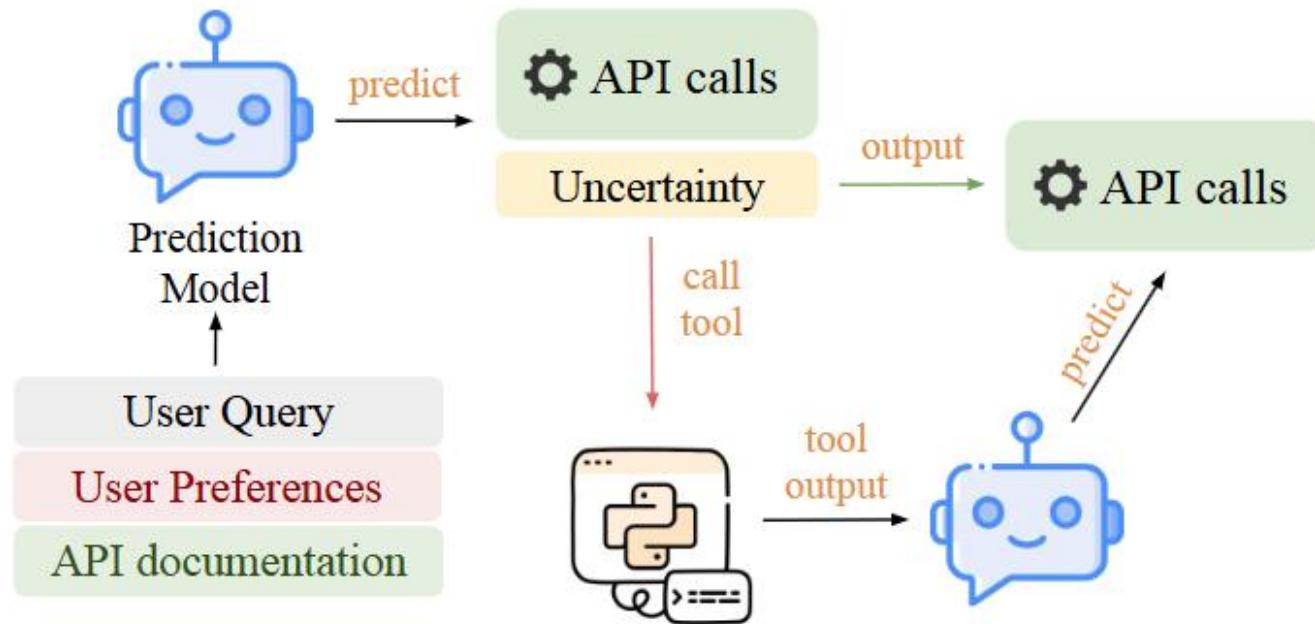
- 모든 모델에서의 EM과 F1 점수 간의 상당한 격차 발생
- 그럴듯한 API 호출은 생성 가능하지만, 자연어를 실행 가능한 코드의 arguments로 변환할 때 필요한 모든 데이터를 정확하게 통합하는 데 어려움을 겪고 있음을 의미함

Qualitative Analysis

- Semantic substitution
 - Open-source 모델은 의미상 유사한 함수 및 argument 이름을 자주 혼동함
 - Mistral-3-Inst의 경우 이러한 오류가 50% 이상 지속적으로 발생
- Missing arguments
 - Multi-hop reasoning 상황에서 해당 오류가 자주 발생
- Hallucination
 - 추가 arguments를 생성하고 새로운 함수를 만듦
 - 모든 모델에서 해당 오류가 빈번하게 발생



TAPS: Tool-Augmented Personalisation via Structured Tagging



TAPS (Tool-Augmented Personalisation via Structured Tagging)

- 기존 Standing instructions를 구조화된 정보로 태깅하여 증강 (자연어 입력과 Tool-call 사이 중간 표현 추가)
- Uncertainty estimation을 활용하여 tagging 여부 결정 (Structured tagging tool)

Original:

If I'm looking for Events, I'd like them to be in New York.

Augmented:

<to be in <**s1:CITYs1**> .

Structured tagging

- 자연어 측면을 보존하여 모델이 원래 standing instruction과 explicit structural information을 모두 활용할 수 있음
- 코드 생성 전 이러한 중간 표현을 추가하는 것이 API arguments 추출 및 API 호출 시 정보 손실 방지에 도움이 된다고 가정

TAPS: Tool-Augmented Personalisation via Structured Tagging

Uncertainty estimation methods

- Sequence Margin: 가장 가능성이 높은 상위 두 예측의 확률 점수 차이
- Margin@T: 가장 가능성이 높은 상위 T개의 토큰 확률 차이
- Least Confidence: 가장 확신 있는 예측의 확률과 100% 확신 사이의 차이 ✓

Tagging methods

- External Tagger (EXT-TAG): 파이프라인 상의 외부 모델 tagging, GPT-4o를 최적의 tagger로 간주 (EXT-TAGOPT)
- TAG-AND-GENERATE (JOINT-TAG): Inference시 prompting으로 tagging까지 진행하도록 shot제시

Method	Statistic
Least Confidence	-0.452
Margin@1	0.145
Margin@2	0.317
Margin@3	0.314
Margin@4	0.295
Margin@5	0.301
Margin@6	0.242
Margin@7	0.263
Margin@8	0.256
Margin@9	0.242
Margin@10	0.236
Sequence Margin	0.281

TAPS: Tool-Augmented Personalisation via Structured Tagging

Model	Aug.	EM ↑	F1 ↑	Prec. ↑	Rec. ↑
Llama-3-Inst	DEFAULT	42.23	78.19	80.30	78.60
	EXT-TAG	<u>44.22</u>	<u>80.34</u>	81.90	<u>81.49</u>
	EXT-TAG _{OPT}	51.79	84.46	86.39	84.86
	JOINT-TAG	41.43	78.31	<u>82.97</u>	77.19
Mistral-3-Inst	DEFAULT	30.68	64.21	65.21	65.37
	EXT-TAG	<u>36.65</u>	<u>75.12</u>	<u>78.33</u>	<u>74.30</u>
	EXT-TAG _{OPT}	42.63	79.34	82.23	79.04
	JOINT-TAG	33.47	66.66	70.56	64.97
GPT4o	DEFAULT	56.18	87.40	90.41	86.83
	EXT-TAG	57.37	87.47	<u>89.63</u>	<u>86.72</u>
	JOINT-TAG	52.99	83.94	86.00	83.24

Effects of structured tagging

- EXT-TAG를 통한 명시적인 tagging 시 open-source 모델 대부분의 지표에서 성능 향상
- GPT-4o에서는 덜 효과적인데, 이는 해당 모델의 강력한 ICL 추론 때문이라고 추측

TAPS: Tool-Augmented Personalisation via Structured Tagging

Model	Aug.	EM ↑	F1 ↑	Prec. ↑	Rec. ↑
Llama-3-Inst	DEFAULT	42.23	78.19	80.30	78.60
	EXT-TAG	<u>44.22</u>	<u>80.34</u>	81.90	<u>81.49</u>
	EXT-TAG _{OPT}	51.79	84.46	86.39	84.86
	JOINT-TAG	41.43	78.31	<u>82.97</u>	77.19
Mistral-3-Inst	DEFAULT	30.68	64.21	65.21	65.37
	EXT-TAG	<u>36.65</u>	<u>75.12</u>	<u>78.33</u>	<u>74.30</u>
	EXT-TAG _{OPT}	42.63	79.34	82.23	79.04
	JOINT-TAG	33.47	66.66	70.56	64.97
GPT4o	DEFAULT	<u>56.18</u>	<u>87.40</u>	90.41	86.83
	EXT-TAG	57.37	87.47	<u>89.63</u>	<u>86.72</u>
	JOINT-TAG	52.99	83.94	86.00	83.24

Internal reasoning does not boost the interpretational abilities of LLMs

- Tool-call 이전 tag를 할 수 있도록 prompting하는 것은 항상 효과적이지 않음
- Recall의 감소는 이러한 접근 방식이 오히려 일부 정보의 손실을 초래할 수 있음을 의미함

TAPS: Tool-Augmented Personalisation via Structured Tagging

Tool detection effects

- Oracle prediction
 - Tool detection의 이론적 상한으로써, tagging 유/무 둘 모두의 성능을 측정 후 선택
 - 최적의 tagger가 사용될 때 모든 모델 및 성능 지표에서 가장 높은 성능을 보임
 - Tool detection에서 uncertainty estimation을 사용하는 것은 효과적이지만, 최적이 아님을 시사
- TAPS
 - 무조건적인 tagging 보다는, 선택적 tagging이 효과적임
 - 동일한 tagging 모델을 사용한 상황에서는 항상 EXT-TAG 보다 우수한 성능을 보임

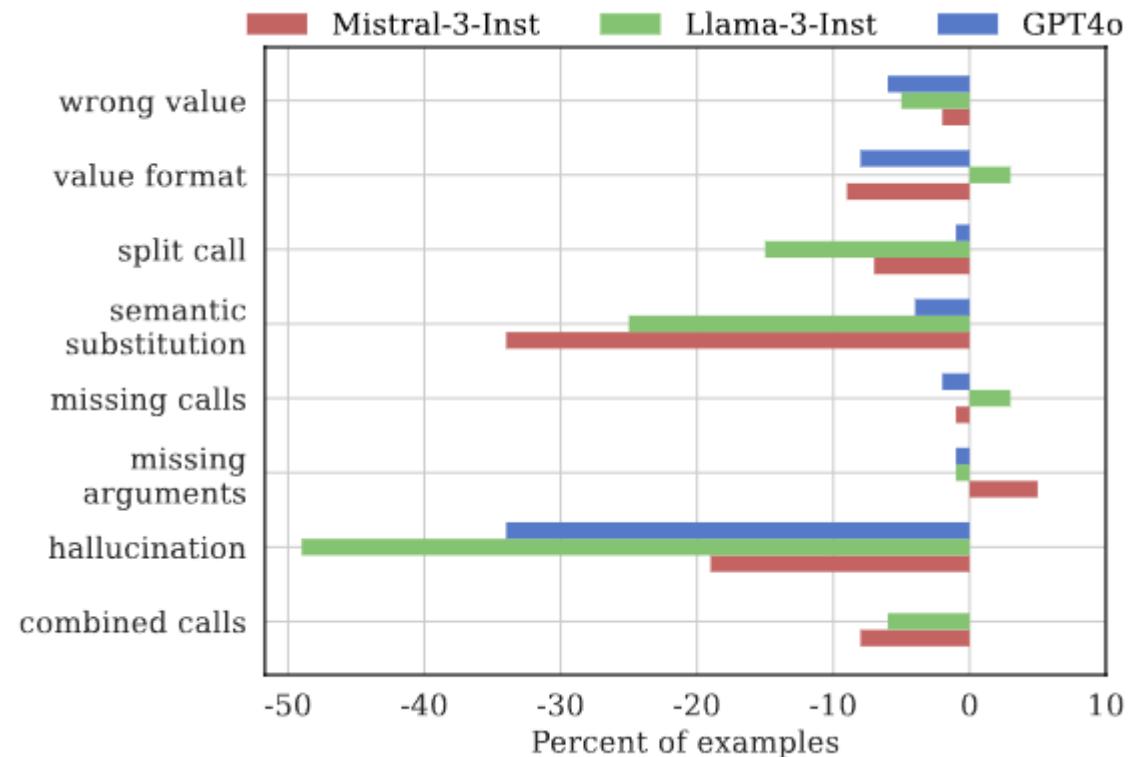
Model	Config	EM ↑	F1 ↑	Prec. ↑	Rec. ↑
Llama-3-Inst	DEFAULT	41.76	78.26	82.96	76.80
	EXT-TAG	47.55	82.28	84.89	81.88
	TAPS	51.18	83.94	87.20	82.95
	TAPS-ORACLE	59.07	88.42	91.60	87.11
	EXT-TAG _{OPT}	51.23	84.51	87.23	83.86
	TAPS _{OPT}	53.04	85.64	88.67	84.56
Mistral-3-Inst	EXT-TAG _{OPT}	59.85	89.65	92.82	88.10
	DEFAULT	35.74	69.11	70.64	69.83
	EXT-TAG	40.29	76.20	79.53	75.57
	TAPS	42.40	76.76	79.74	76.32
	TAPS-ORACLE	48.82	81.79	84.29	81.33
	TAPS-ORACLE _{OPT}	49.85	83.19	86.19	82.36
GPT4o	EXT-TAG _{OPT}	42.35	78.55	82.63	77.24
	DEFAULT	56.32	86.99	89.25	86.91
	EXT-TAG	55.54	86.49	88.78	85.65
	TAPS	58.63	87.86	90.03	87.21
TAPS-ORACLE	EXT-TAG _{OPT}	65.88	91.46	93.57	90.49
	TAPS-ORACLE	65.88	91.46	93.57	90.49

Tool detection의
이론적 상한

TAPS: Tool-Augmented Personalisation via Structured Tagging

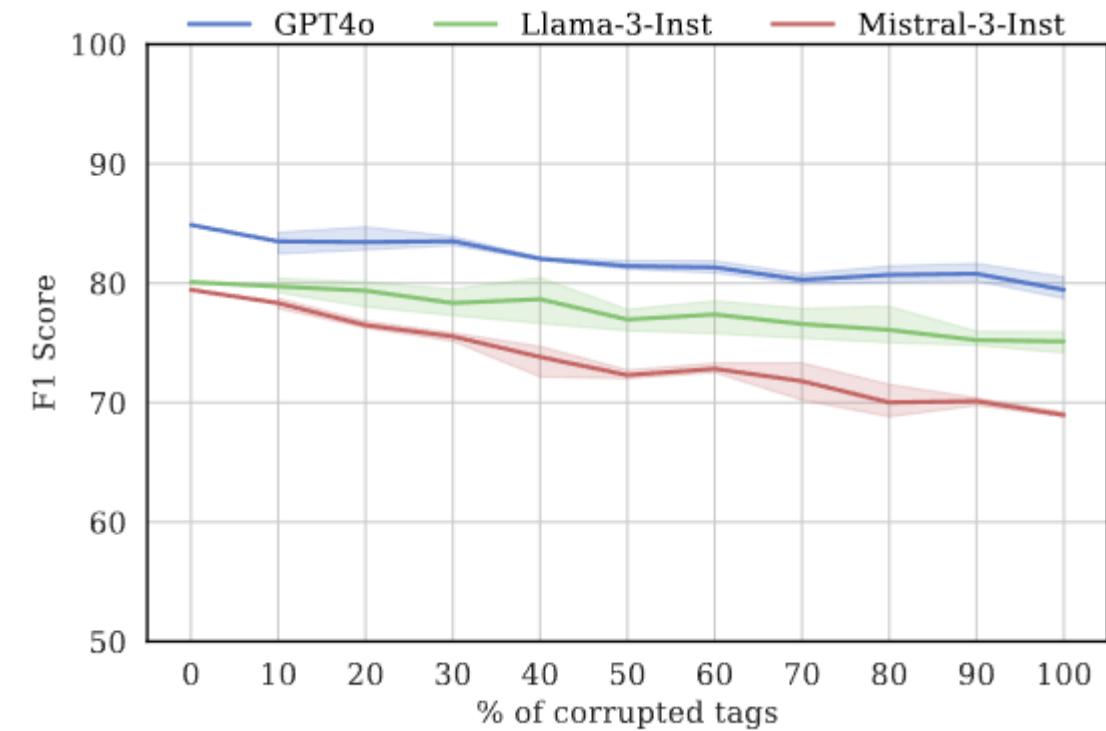
Changes in the distribution of errors in TAPS

- Hallucination(19~49% 감소), semantic substitution errors(4~34% 감소)가 가장 크게 관찰됨
- 일부 모델의 경우 일부 오류 유형이 약간 증가하는데, 이는 GPT-4o를 tagger 모델로 사용하여 발생되는 error propagation일 수 있음
 - Ex) GPT-4o에서 두드러지게 나타나는 error 타입은 value format임
- 전반적으로 TAPS를 사용하면 모든 모델의 대부분의 오류 유형에서 오류율이 크게 감소함



The effects of tag quality

- N%의 tag를 무작위로 손상시킴
 - Slot deletion
 - Tag boundary shifts
 - Semantic substitution of slot and function names
- GPT-4o와 Llama-3-Inst는 tag 품질이 최대 5%까지 저하되어도 비교적 성능을 유지함
- EXT-TAG의 효과가 생성된 태그의 품질에 직접적으로 의존한다는 것을 의미함



Thank you

Q&A