Boost aLLMs' multi-hop reasoning

하이퍼파라미터와 간단한 모델 구조 변경을 중심으로

학생 원종찬, 최재원

Prof. Jae-Hong Lee, Jun-Hyung Park

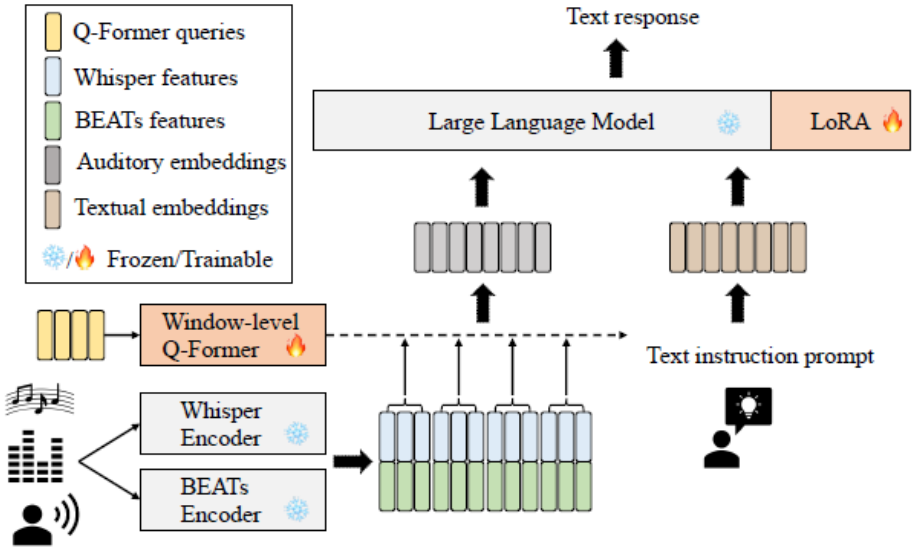
Contents

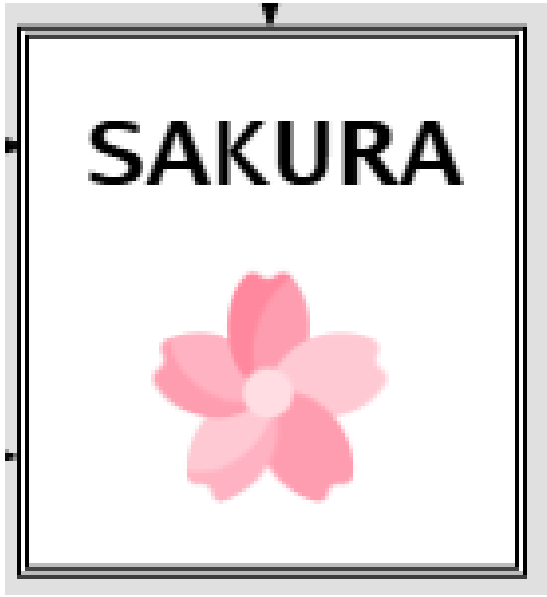
리뷰

- 250922
  - SAKURA 및 JASCO 논문 발표
- 250929
  - SALMONN에 대해 후처리를 이유로 재현상 어려움을 제시
- 251013
  - SAKURA 및 SALMONN의 재현 결과 공유
- 251027
  - SALMONN 재현 종료

목표 : LALM의 성능 높이기

- **대상 모델** : 최대한 다양한 LALM이나, 이번 보고에서는 SALMONN을 기준으로 고민
- **성능 평가** : SAKURA
- **개선 방법** : 1. 하이퍼 파라미터 2. 모델 구조적 접근





SAKURA 평가 결과 분석

- SALMONN 포함 LALMs의 성능 미흡의 원인?

Table 2: *Accuracies (%) and 95% confidence intervals of the baselines on SAKURA. "Single" and "Multi" denote the single-hop and multi-hop sub-tracks. The best and the second-best performances among open-source and proprietary LALMs are marked in bold and underlined. Model sizes (except proprietary ones) are provided, with cascaded models showing the sum of involved modules.*

	Size (B)	Gender		Language		Emotion		Animal		Average	
		Single	Multi	Single	Multi	Single	Multi	Single	Multi	Single	Multi
Open-source LALMs											
LTU-AS	7	52.4 ± 4.4	19.6 ± 3.5	16.8 ± 3.3	11.4 ± 2.8	28.6 ± 4.0	19.6 ± 3.5	65.6 ± 4.2	21.8 ± 3.6	40.9 ± 4.3	18.1 ± 3.4
GAMA-IT	7	76.4 ± 3.7	39.8 ± 4.3	5.6 ± 2.0	19.4 ± 3.5	5.6 ± 2.0	24.2 ± 3.8	85.2 ± 3.1	51.4 ± 4.4	43.2 ± 4.3	33.7 ± 4.1
SALMONN	7.5	59.8 ± 4.3	48.6 ± 4.4	21.8 ± 3.6	29.6 ± 4.0	19.8 ± 3.5	28.2 ± 3.9	68.6 ± 4.1	34.6 ± 4.2	42.5 ± 4.3	35.3 ± 4.2
DeSTA2	8.3	**88.4** ± 2.8	**85.2** ± 3.1	94.2 ± 2.0	75.4 ± 3.8	34.8 ± 4.2	36.4 ± 4.2	34.4 ± 4.2	31.2 ± 4.1	63.0 ± 4.2	**57.1** ± 4.3
Qwen-Audio-Chat	8.4	49.6 ± 4.4	43.8 ± 4.3	87.6 ± 2.9	40.6 ± 4.3	63.2 ± 4.2	37.0 ± 4.2	**92.2** ± 2.4	**66.0** ± 4.2	73.2 ± 3.9	46.9 ± 4.4
Qwen2-Audio-Instruct	8.4	88.0 ± 2.8	47.2 ± 4.4	83.8 ± 3.2	48.0 ± 4.4	**64.2** ± 4.2	**39.8** ± 4.3	88.8 ± 2.8	61.4 ± 4.3	**81.2** ± 3.4	49.1 ± 4.4
Proprietary LALMs											
GPT-4o Audio	-	-	-	95.2 ± 1.9	83.6 ± 3.2	38.2 ± 4.3	23.8 ± 3.7	80.6 ± 3.5	55.4 ± 4.4	71.3 ± 4.0	54.3 ± 4.4
Gemini-1.5-flash	-	77.0 ± 3.7	24.2 ± 3.8	**98.2** ± 1.2	79.8 ± 3.5	24.6 ± 3.8	19.4 ± 3.5	27.2 ± 3.9	16.2 ± 3.2	56.8 ± 4.3	34.9 ± 4.2
Gemini-1.5-pro	-	74.0 ± 3.8	43.4 ± 4.3	97.2 ± 1.4	**90.6** ± 2.6	39.2 ± 4.3	24.0 ± 3.7	42.0 ± 4.3	28.6 ± 4.0	63.1 ± 4.2	46.6 ± 4.4
Cascaded Systems											
ASR+LLM	9.5	24.2 ± 3.8	32.2 ± 4.1	93.6 ± 2.1	82.4 ± 3.3	21.4 ± 3.6	30.6 ± 4.0	30.8 ± 4.0	27.6 ± 3.9	42.5 ± 4.3	43.2 ± 4.3
ASR+AAC+LLM	17.9	85.0 ± 3.1	79.6 ± 3.5	93.4 ± 2.2	88.8 ± 2.8	60.0 ± 4.3	51.4 ± 4.4	78.0 ± 3.6	78.4 ± 3.6	79.1 ± 3.6	74.5 ± 3.8
Random Baseline											
Chance level	-	50.0	50.0	25.0	25.0	25.0	25.0	25.0	25.0	31.3	31.3

SAKURA 평가 결과 분석 : 예상 원인

- **ASR 성능의 미흡** : 아직 개선할 ASR 성능이 남아 있고, ASR 성능을 좀 더 끌어올림을 통해 추론 능력을 소폭 향상시킬 수 있을 것이다.

- **초기 정보의 손실** : 오디오 데이터, 입력 프롬프트의 길이가 길어지면서 초기 정보에 대한 attention이 걸리지 않는 문제가 처음 리뷰에서 언급됨

Multimodality

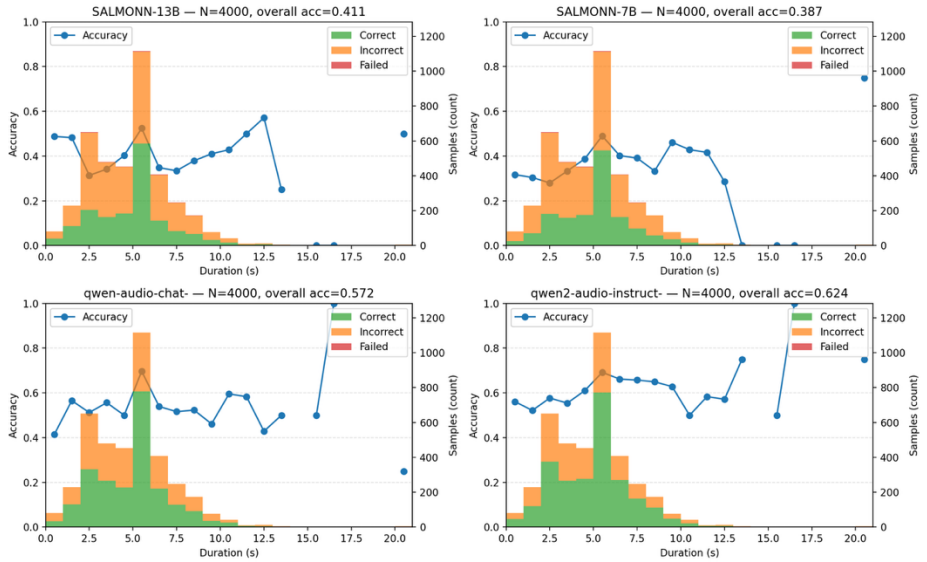
BIG Gap remains – Open Directions

- Encoding high resolution images
- Encoding long sequences
- Integrating domain knowledge

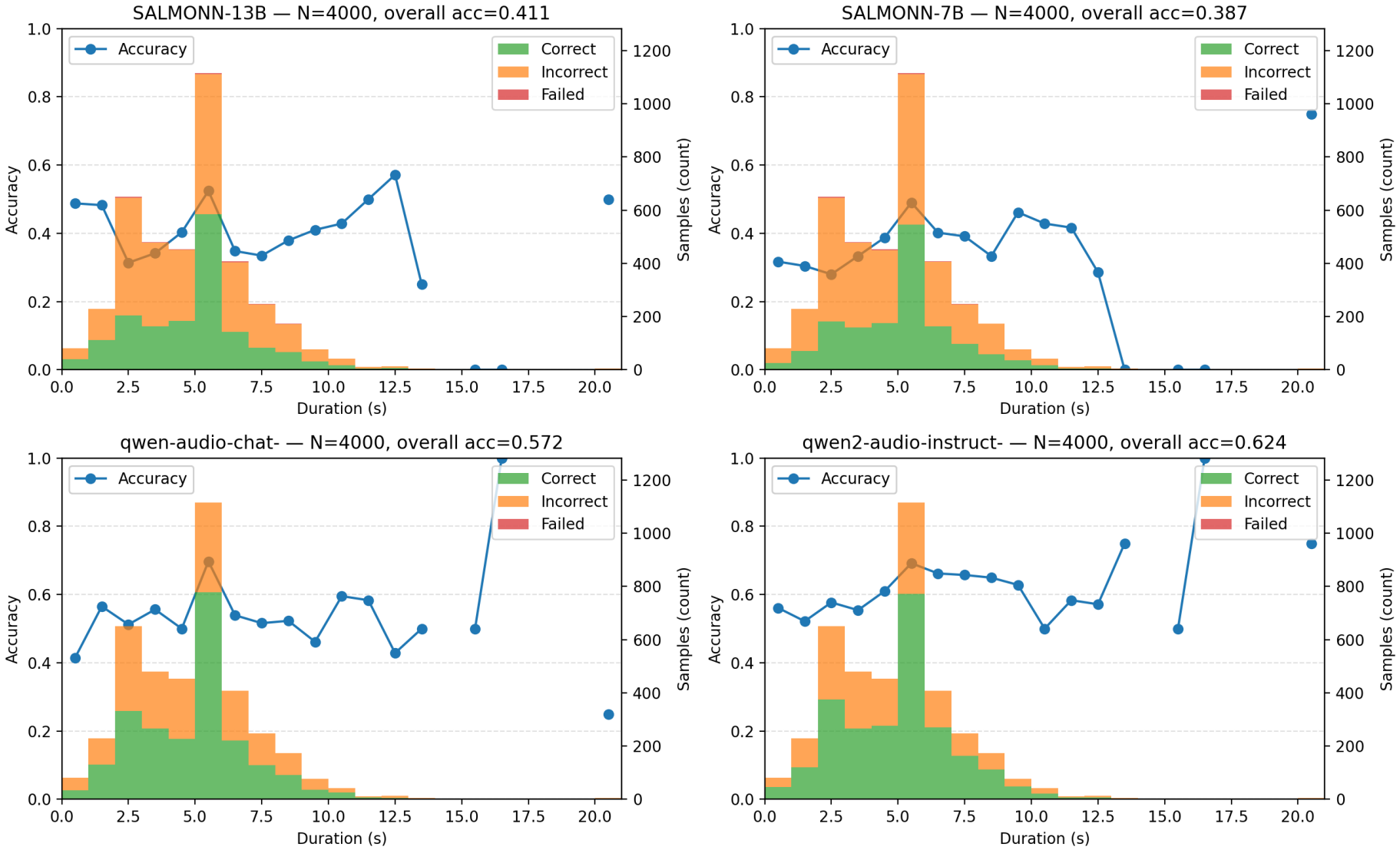
SAKURA 평가 결과 분석 : 예상 원인 검증

- **ASR 성능의 미흡** : 아직 개선할 ASR 성능이 남아 있고, ASR 성능을 좀 더 끌어올림을 통해 추론 능력을 소폭 향상시킬 수 있을 것이다.**(검증 못함, 그러나 ASR + LLM 구조 아이디어)**

- **초기 정보의 손실** : 오디오 데이터, 입력 프롬프트의 길이가 길어지면서 초기 정보에 대한 attention이 걸리지 않는 문제가 처음 리뷰에서 언급됨 **(검증 과정 :입력 오디오 길이별 acc 분석)**



SAKURA 평가 결과 분석 : 예상 원인 검증



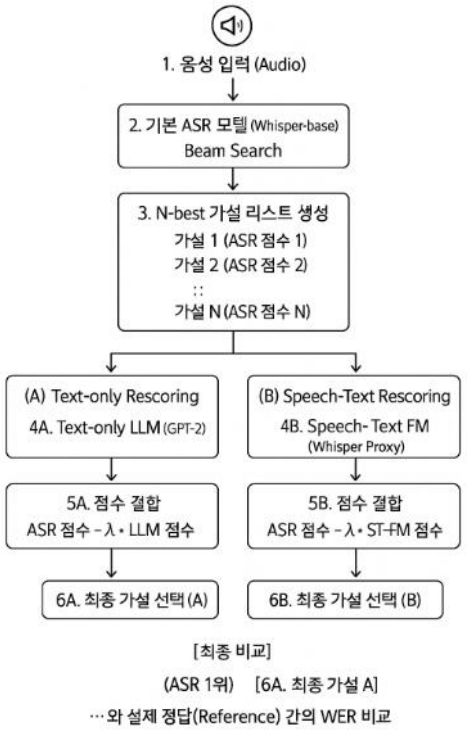
SAKURA 성능 개선 아이디어 1

- **ASR 성능 고도화에 집중** : ASR + LLM cascaded system이 성능이 비교적 좋았었기에, 단순한 연결 시스템에서 LLM에게 ASR 보정 지능을 주어 더 높은 성능의 ASR 확보 (잡음 속에서도 인식) (*이하의 논문을 출발점 삼아)

Speech Recognition Rescoring with Large Speech-Text Foundation Models

Prashanth Gurunath Shivakumar, Jari Kolehmainen, Aditya Gourav, Yi Gu, Ankur Gandhe, Ariya Rastrow, Ivan Bulyko
Amazon Science, Seattle, Washington, U.S.A
pgurunat@usc.edu, {jkolehm,gouravag,yilegu,aggandhe,arastrow,ibbulyko}@amazon.com

```
torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 20.00 MiB. GPU 0 has a total capacity of 31.37 GiB of which 7.88 MiB is
free. Including non-PyTorch memory, this process has 942.00 MiB memory in use. Of the allocated memory 438.65 MiB is allocated by PyTo
rch, and 7.35 MiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large try setting PYTORCH_CUDA_ALLOC_C
ONF=expandable_segments:True to avoid fragmentation.  See documentation for Memory Management  (https://pytorch.org/docs/stable/notes/
cuda.html#environment-variables)
(base) jpong@hufs02:~/Workspace/JongChan/ASRLLM$
```



SAKURA 성능 개선 아이디어 1

- ASR 성능 고도화에 집중 : ASR + LLM cascaded system이 성능이 비교적 좋았었기에, 단순한 연결 시스템에서 LLM에게 ASR 보정 지능을 주어 더 높은 성능의 ASR 확보 (잡음 속에서도 인식) (*이하의 논문을 출발점 삼아)

** **해당 시도는 보류** : (1) ASR **고도화 과정**이 복잡

(2) ASR 성능의 SAKURA에 대한 **영향력이 아직 미검증**

SAKURA 성능 개선 아이디어 2

- **S4(Structured State Space for Sequence modeling)**

- S4는 시퀀스(문장·오디오·시계열 등)의 **긴 과거(history)를 효율적으로 기억하고 요약**하는 모델

Efficiently Modeling Long Sequences with Structured State Spaces

Albert Gu, Karan Goel, and Christopher Ré

Department of Computer Science, Stanford University

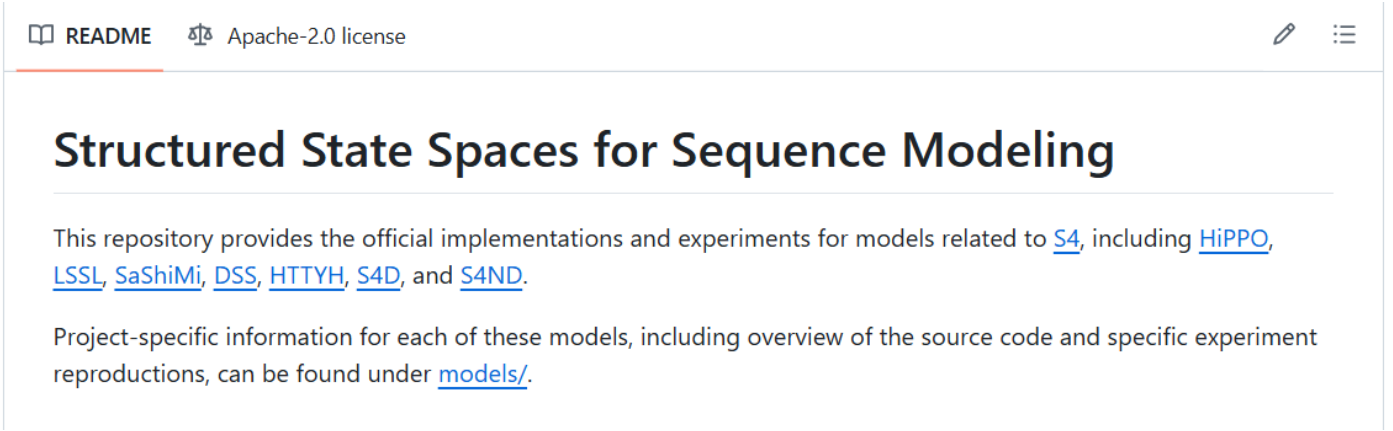
{albertgu,krng}@stanford.edu, chrismre@cs.stanford.edu

SAKURA 성능 개선 아이디어 2

- A. **Whisper는 고정(frozen) → Whipser 출력에 S4적용**
- B. **비음성 전용 인코더를 S4로 교체**
- C. **S4를 LLM에 적용**
- 구체적 구현 방법에 대해 S4 깃헙 레포를 통해 조사중

SAKURA 성능 개선 아이디어 2

- A. **Whisper는 고정(frozen) → Whipser 출력에 S4적용**
- B. **비음성 전용 인코더를 S4로 교체**
- C. **S4를 LLM에 적용**
- 구체적 구현 방법에 대해 S4 깃헙 레포를 통해 조사중

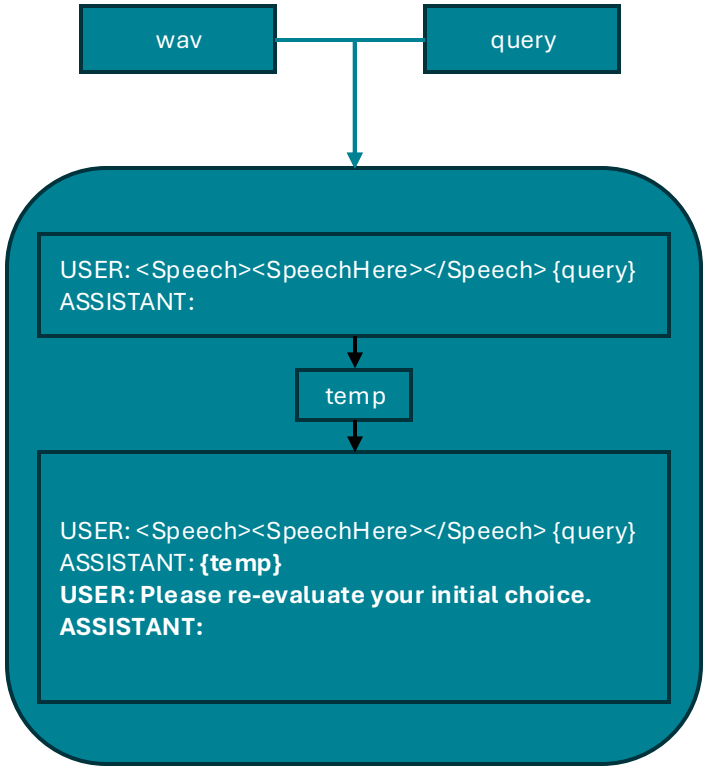


앞으로 할 일

- **입력 길이에 따른 성능저하** 여부 및 **S4 적용 가능성에 대한 기술적 검토**
- SAKURA에 대한 **저성능 원인을 밝힐 수 있는 실험** 설계

exp1: chatting templated re-evaluate | x -> x

- 정답
  - (c) Give them space to cool off before engaging further.
- 초기 출력
  - (b) Provide emotional support and avoid dismissing their sadness.
- 기존 프롬프트 + "다시 평가해봐"
  - USER: <Speech><SpeechHere></Speech> 질문 프롬프트
  - ASSISTANT: 초기 출력
  - USER: Please **re-evaluate** your initial choice.
  - ASSISTANT:
- 최종 출력
  - (a) Smile and engage positively in the conversation.

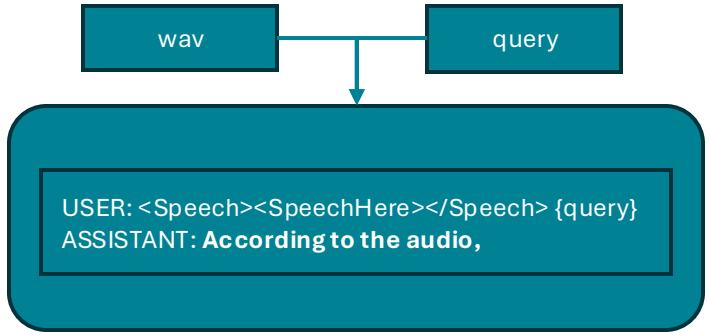


exp1: chatting templated re-evaluate – Code


```python
results = self.model.generate(
    samples,
    self.config.generate,
    prompts=prompts,
    skip_special_tokens=True,
)
print("=" * 50)
print("answers:", samples["text"])
print("Initial results:", results)
new_prompts = [f"{p} {r}\nUSER: Please re-evaluate your initial choice.\nASSISTANT:" for p, r in zip(prompts, results)]
print("new prompts:", new_prompts)
results = self.model.generate(
    samples,
    self.config.generate,
    prompts=new_prompts,
)
print("Revised results:", results)
```

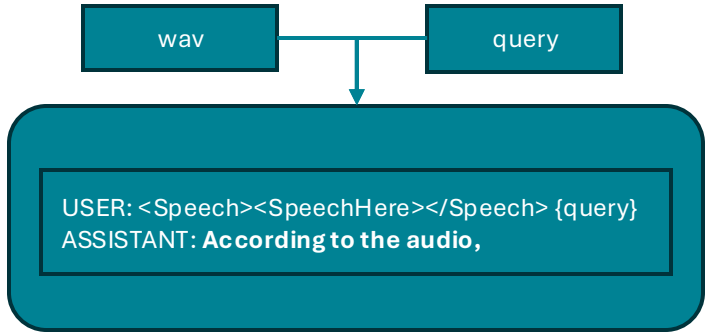
exp1-2: set initial tokens | x -> x

- 정답
  - (d) Stop making excuses; this is entirely your fault!
- 초기 출력
  - (b) This is the best day ever, I feel on top of the world!
- 기존 프롬프트 + 초기 응답 토큰 설정
  - USER: <Speech><SpeechHere></Speech> 질문 프롬프트
  - ASSISTANT: According to the audio,
- 최종 출력
  - the sentence that people feeling the same emotion are most likely to say is (c) I'm really scared about what might happen next.



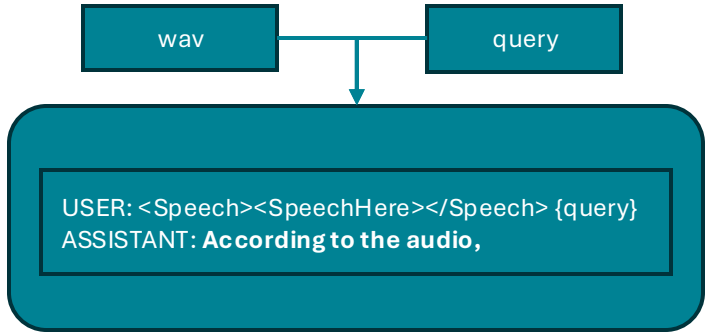
exp1-2: set initial tokens | x -> o

- 정답
  - (d) How dare they treat me so unfairly like this!
- 초기 출력
  - (c) This is terrifying, I can’t stop thinking about it.
- 기존 프롬프트 + 초기 응답 토큰 설정
  - USER: <Speech><SpeechHere></Speech> 질문 프롬프트
  - ASSISTANT: According to the audio,
- 최종 출력
  - the emotional state of others with the same emotion is (c) This is terrifying, I can’t stop thinking about it. Therefore, option (d) How dare they treat me so unfairly like this! best matches the emotional state of others with the same emotion expressed in the audio.



exp1-2: set initial tokens | o -> x

- 정답
  - (b) A sharp, piercing gaze and compressed lips
- 초기 출력
  - (b) A sharp, piercing gaze and compressed lips
- 기존 프롬프트 + 초기 응답 토큰 설정
  - USER: <Speech><SpeechHere></Speech> 질문 프롬프트
  - ASSISTANT: According to the audio,
- 최종 출력
  - option (a) Eyes gazing downward with trembling lips best matches the emotion.



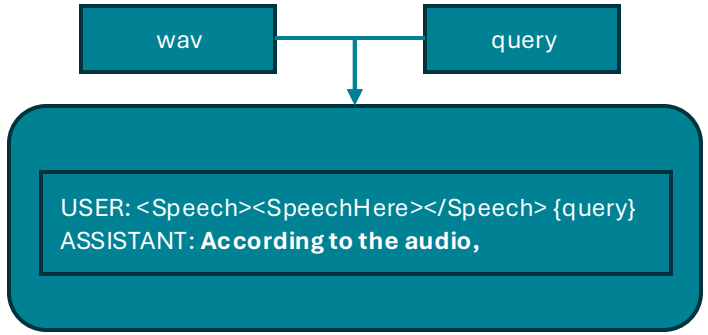
exp1-2: set initial tokens – Code

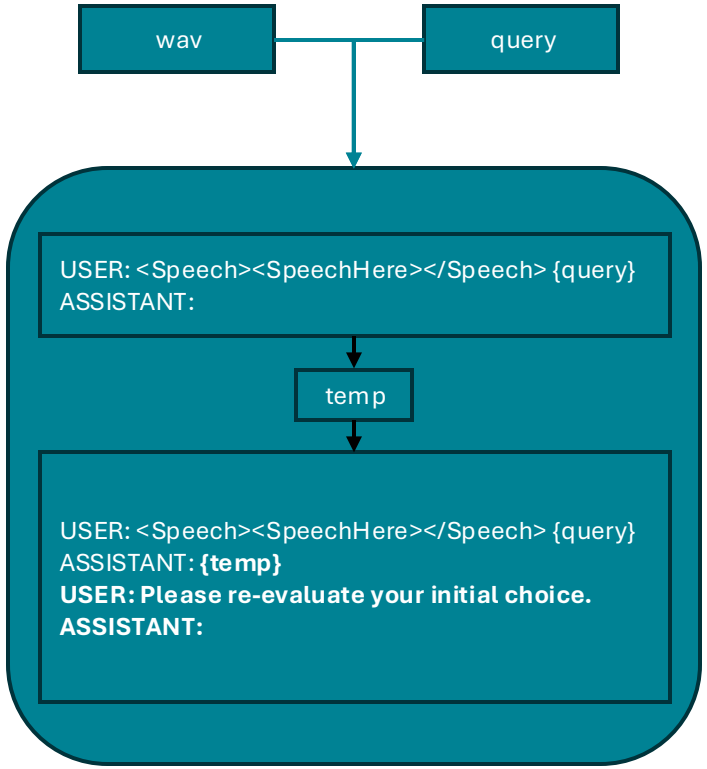

```
prompt_template: "USER: {}\nASSISTANT:"
```

```
prompts = [
    self.config.model.prompt_template.format("<Speech><SpeechHere></Speech> " + q.strip())
    + " According to the audio,"
    for q in samples["query"]
]
```

Experimental Summary 1

- 프롬프트를 조금 건드리는 것으로는 제대로 고치지 못 함
  - 대화형으로 다시 평가를 요구하는 방식
    - USER: 질문
    - ASSISTANT: 초기 응답
    - USER: Please **re-evaluate** your initial choice.
    - ASSISTANT:
  - 모델 출력의 초기값을 "According to the audio,"로 강제하는 방식





decoder approaches

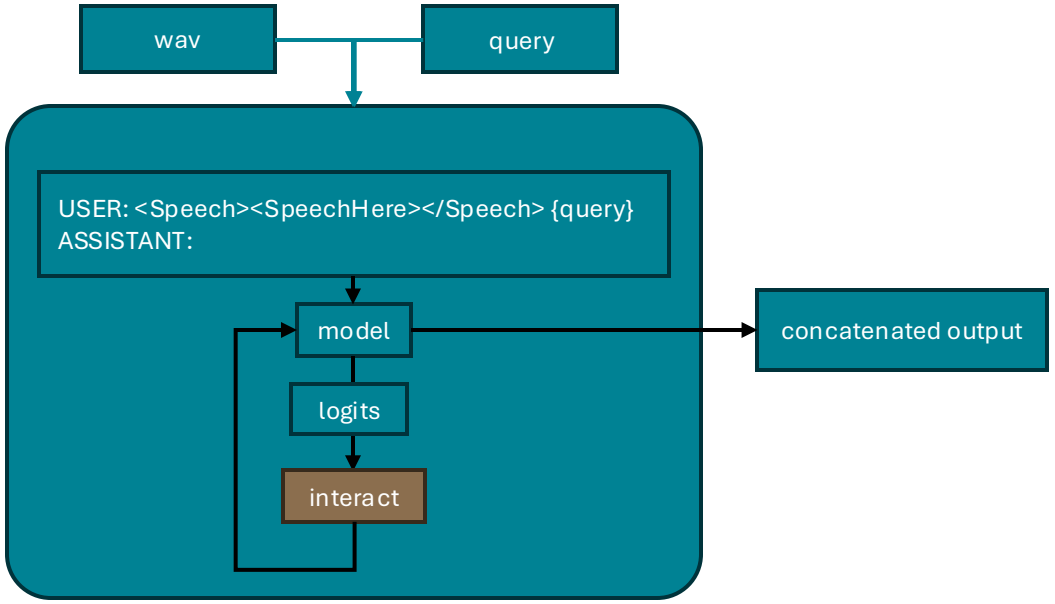
- 목적
  - aLLMs의 multi-hop reasoning 성능을 높이자

- (직접 둔) 제한
  - 기존의 벤치마크를 그대로 사용할 수 있어야 함
  - 새로운 학습 과정이 없어야 함
  - 적용했을 때의 성능과 계산 비용 간의 trade-off를 고려해야 함
  - aLLMs 뿐만 아니라 범용적으로 적용할 수 있으면 좋겠음

- 가정
  - 베이스 모델은 이미 충분히 잘 학습되어 일부 트윅으로 개선될 여지가 있음

exp2: Interactive Inference

- single/multi-hop에 대해 (in)correct일 때 logits를 비교해보자!
  - num_beams=1; greedy



multi-hop incorrect cases

- multi-hop에서 틀렸을 때, candidates 간의 차이가 작음
  - fig2는 0.06쯤, fig3는 0.09쯤 차이

```
answers: ['(a) A sharp, piercing gaze and compressed lips']
initial results: ['(d) Relaxed facial muscles with a cheerful grin']

===== step 1 =====
>
        < next token candidates >
[1] '(' (logit: 21.4531)
[2] 'The' (logit: 19.5000)
[3] 'b' (logit: 19.1719)
[4] 'I' (logit: 18.6406)
[5] 'a' (logit: 18.6250)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
        < next token candidates >
[1] 'd' (logit: 19.7812)
[2] 'b' (logit: 19.7188)
[3] 'a' (logit: 19.6094)
[4] 'c' (logit: 19.1875)
[5] 'A' (logit: 16.2500)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig2

```
answers: ['(d) Stop making excuses; this is entirely your fault!']
initial results: ['(c) I feel like I'm in danger, and I don't know what to do.']

===== step 1 =====
>
        < next token candidates >
[1] '(' (logit: 22.6094)
[2] 'The' (logit: 19.2969)
[3] 'b' (logit: 19.2188)
[4] 'Answer' (logit: 19.2031)
[5] 'c' (logit: 18.5781)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
        < next token candidates >
[1] 'd' (logit: 20.1562)
[2] 'b' (logit: 20.0625)
[3] 'c' (logit: 20.0312)
[4] 'a' (logit: 19.5000)
[5] 'C' (logit: 16.6875)
[0] EOS (exit)
select (1-5, 0=exit): 0
```

fig3

multi-hop correct cases

- multi-hop에서 맞았을 때, candidates 간의 차이가 큼
  - fig4는 0.5쯤, fig5는 1.5쯤 차이

```
answers: ['(d) Listen attentively without interrupting to let them vent.']
initial results: ['(d) Listen attentively without interrupting to let them vent.']

===== step 1 =====
>
       < next token candidates >
[1] '(' (logit: 21.1562)
[2] 'The' (logit: 19.6250)
[3] 'Option' (logit: 18.5312)
[4] 'B' (logit: 18.4531)
[5] 'b' (logit: 18.2500)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
       < next token candidates >
[1] 'd' (logit: 19.8906)
[2] 'b' (logit: 19.4062)
[3] 'c' (logit: 19.2656)
[4] 'a' (logit: 18.0781)
[5] 'D' (logit: 16.8750)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig4

```
answers: ['(b) Calmly acknowledge their frustration and suggest a solution.']
initial results: ['(b) Calmly acknowledge their frustration and suggest a solution.']

===== step 1 =====
>
       < next token candidates >
[1] '(' (logit: 21.6562)
[2] 'The' (logit: 20.0312)
[3] 'B' (logit: 18.7969)
[4] 'Option' (logit: 18.5781)
[5] 'Answer' (logit: 18.4219)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
       < next token candidates >
[1] 'b' (logit: 20.1875)
[2] 'c' (logit: 18.7188)
[3] 'a' (logit: 18.0156)
[4] 'B' (logit: 17.9531)
[5] 'd' (logit: 17.7500)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig5

single-hop incorrect cases

- single-hop에서 틀렸을 때, candidates 간의 차이가 작음
  - fig6는 0.000..쯤, fig7는 0.05쯤 차이

```
answers: [' (b) angry']
initial results: ['(c) Disgust']

===== step 1 =====
>
        < next token candidates >
[1] '(' (logit: 20.0312)
[2] 'I' (logit: 17.5625)
[3] 'The' (logit: 17.1875)
[4] 'It' (logit: 17.1094)
[5] 'B' (logit: 16.7969)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
        < next token candidates >
[1] 'd' (logit: 20.1875)
[2] 'b' (logit: 20.1875)
[3] 'c' (logit: 19.9531)
[4] 'a' (logit: 19.3906)
[5] 'B' (logit: 18.2969)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig6

```
answers: [' (c) angry']
initial results: ['The emotional tone of the speaker in the audio is (d) happy.']

===== step 1 =====
>
        < next token candidates >
[1] '(' (logit: 20.1719)
[2] 'The' (logit: 18.9375)
[3] 'a' (logit: 16.7812)
[4] 'd' (logit: 16.5312)
[5] 'There' (logit: 16.3750)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
        < next token candidates >
[1] 'b' (logit: 20.0312)
[2] 'd' (logit: 19.9844)
[3] 'a' (logit: 19.9688)
[4] 'c' (logit: 19.7344)
[5] 'A' (logit: 16.6562)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig7

single-hop correct cases

- single-hop에서 맞았을 때, candidates 간의 차이가 큼
  - fig8는 1.4쯤, fig9는 1.3쯤 차이

```
answers: [' (d) angry']
initial results: ['The emotional tone of the speaker in the audio is angry.']

===== step 1 =====
>
        < next token candidates >
[1] '(' (logit: 21.4062)
[2] 'The' (logit: 20.5781)
[3] 'An' (logit: 19.5781)
[4] 'angry' (logit: 19.0469)
[5] 'a' (logit: 17.7031)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
        < next token candidates >
[1] 'd' (logit: 20.9219)
[2] 'a' (logit: 19.5469)
[3] 'c' (logit: 19.5312)
[4] 'D' (logit: 18.8438)
[5] 'b' (logit: 18.5312)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig8

```
answers: [' (b) angry']
initial results: ['The speaker is expressing anger in the audio.']

===== step 1 =====
>
        < next token candidates >
[1] '(' (logit: 20.3438)
[2] 'The' (logit: 20.2188)
[3] 'a' (logit: 16.5156)
[4] 'B' (logit: 16.3750)
[5] 'b' (logit: 16.1719)
[0] EOS (exit)
select (1-5, 0=exit): 1
-- selected: '(' --

===== step 2 =====
> (
        < next token candidates >
[1] 'b' (logit: 21.0469)
[2] 'c' (logit: 19.7500)
[3] 'a' (logit: 19.6250)
[4] 'd' (logit: 18.9688)
[5] 'B' (logit: 16.3125)
[0] EOS (exit)
select (1-5, 0=exit):
```

fig9

Experimental Summary 2

- model uncertainty가 inaccuracy하게 함을 확인함

- Types of uncertainty src: Lecture Note 2. Probability: Univariate Models (1) @고급인공지능수학
  - epistemic uncertainty (model uncertainty)
    - 입출력 매핑에 대한 불완전한 지식 때문
    - 데이터 생성 메커니즘이나 원인에 대한 지식 부족 때문
    - 추가 데이터 수집이나 더 나은 모델로 줄일 수 있는 불확실성
  - aleatoric uncertainty (data uncertainty)
    - 매핑 자체에 내재된 확률적 변동성 때문
    - 본직적인 데이터 변동성 때문
    - 이 종류의 불확실성은 추가 데이터를 수집해도 줄일 수 없음

Keywords for Experiment 2

- keywords
  - dynamic decoding
  - adaptive decoding
  - speculative decoding

src: https://developer.nvidia.com/ko-kr/blog/an-introduction-to-speculative-decoding-for-reducing-latency-in-ai-inference/

Speculative Decoding이란?

Speculative decoding은 추론 성능을 최적화하기 위한 기법으로, 고성능 대상 모델에 빠르고 가벼운 드래프트 메커니즘을 결합하는 방식입니다. 드래프트 모델이 여러 개의 다음 토큰을 빠르게 제안하면, 대상 모델은 이를 한 번의 포워드 패스로 검증한 뒤, 자신의 예측과 일치하는 가장 긴 접두어(prefix)까지 수용하고 그 이후부터 생성을 이어갑니다. 기존 오토리그레시브 방식이 매번 한 토큰씩 생성하는 데 비해, speculative decoding은 여러 토큰을 한꺼번에 처리할 수 있어 지연 시간을 줄이고 처리량은 높이면서도 정확도에는 영향을 주지 않습니다.

src: transformers/generation/utils.py

```
2126        # 10. go into different generation modes
2127 >      if generation_mode == GenerationMode.ASSISTED_GENERATION: …
2169 >      elif generation_mode == GenerationMode.DOLA_GENERATION: …
2185
2186 >      elif generation_mode == GenerationMode.CONTRASTIVE_SEARCH: …
2204
2205 >      elif generation_mode in (GenerationMode.SAMPLE, GenerationMode.GREEDY_SEARCH): …
2224
2225 >      elif generation_mode in (GenerationMode.BEAM_SAMPLE, GenerationMode.BEAM_SEARCH): …
2255
2256 >      elif generation_mode == GenerationMode.GROUP_BEAM_SEARCH: …
2285
2286 >      elif generation_mode == GenerationMode.CONSTRAINED_BEAM_SEARCH: …
```

Related Works for Experiment 2 (1)

- AdaDec: Uncertainty-Guided Adaptive Decoding for LLM-based Code Generation
  - arXiv:2506.08980
    - This paper presents an empirical study revealing that **many generation errors stem from ranking mistakes at high-uncertainty steps**, where the **correct token is present but not top-ranked**.
    - Motivated by these findings, we propose AdaDec, an **uncertainty-guided adaptive decoding** framework that integrates a **token-level pause-then-rerank mechanism driven by token uncertainty** (Shannon entropy).
      - AdaDec **learns model-specific uncertainty thresholds** and applies a **lookahead-based reranking strategy when uncertainty is high**.

- 생각했던 것과 완전히 동일했다

Related Works for Experiment 2 (2)

- Estimating LLM Uncertainty with Evidence
  - arXiv:2502.00290
    - In this paper, we reveal that the **probability-based method fails to estimate token reliability** due to the loss of evidence strength information which is accumulated in the training stage.
    - Therefore, we present Logits-induced token uncertainty (LogTokU), a framework for **estimating decoupled token uncertainty** in LLMs, enabling **real-time uncertainty estimation without requiring multiple sampling processes**.
    - We employ **evidence modeling** to implement LogTokU and use the estimated uncertainty **to guide downstream tasks**.

- 해보고 싶은 방향이었는데 역시나 이미 논문으로 나와있다
  - 논문보다 더 잘하면 되는 거 아닐까?

Related Works for Experiment 2 (3)

- Chain-of-Thought Reasoning Without Prompting
  - arXiv:2402.10200
    - **CoT reasoning paths can be elicited** from pre-trained LLMs **by simply altering the decoding process**.
    - This approach not only **bypasses the confounders of prompting** but also allows us to **assess the LLMs' intrinsic reasoning abilities**.
    - Moreover, we observe that the **presence of a CoT in the decoding path** correlates with a **higher confidence** in the model's decoded answer.
    - Extensive empirical studies on various reasoning benchmarks show that the proposed CoT-decoding effectively **elicits reasoning capabilities** from language models, which were **previously obscured** by standard greedy decoding.

- exp2: Interactive Inference로 실험할 때 위와 같은 생각을 함

(planned) exp3: decoding algorithms – find detouring path

- 현 시점의 logits가 유사한 경우, 이를 우회할 수 있는 경로가 있지 않을까?
  - exp2의 코드로는 실험이 불가능함
    - exp2의 구현 원리: 선택한 인덱스 이외에는 로짓을 –inf로 줌
      - new_logits = torch.full_like(logits, -float("inf"))
      - new_logits[chosen_token_id] = 0.0

- 체크 포인트
  - 1. KV cache: 반복계산 방지
  - 2. 유사한 정도의 공식화: ex. (sorted[0] / sorted[1]) < x
  - 3. hyper-parameters: top-k, detour-edges
  - 4. reject option

5.1.2.3 Classification with the "reject" option

일부 경우에는, 우리가 신뢰할 수 없는 답변을 반환하는 대신 “모르겠다”라고 말할 수 있다. 이를 **reject option**을 선택한다고 부른다. 이러한 방식은 특히 위험을 회피해야 하는 의료나 금융과 같은 분야에서 중요하다.

$$a^* = \begin{cases} y^* & \text{if } p^* > \lambda^* \\ \text{reject} & \text{otherwise} \end{cases}$$

src: Lecture Note 10. Decision Theory @고급인공지능수학

계획

- 장기적으로 디코딩에 집중해보고 싶음

- exp1은 여기까지만
- exp2-3은 방향에 따라 프로젝트와 관련이 있다면 계속 해볼 생각
  - 관련이 없다면 새로운 방법을 모색해보고, 해당 방법론은 혼자 조금 더 집중해볼 계획입니다.

- 이번에 방향을 제시해주신다면, 그 방향으로 고민해보겠습니다.

Boost aLLMs' multi-hop reasoning
하이퍼파라미터와 간단한 모델 구조 변경을 중심으로

학생 원종찬, 최재원
Language & AI융합전공

2025. 11. 03. 3 PM.
한국외대 교수회관 401호