

2. 공개 모델을 이용한 RAG

발표 자료 소개

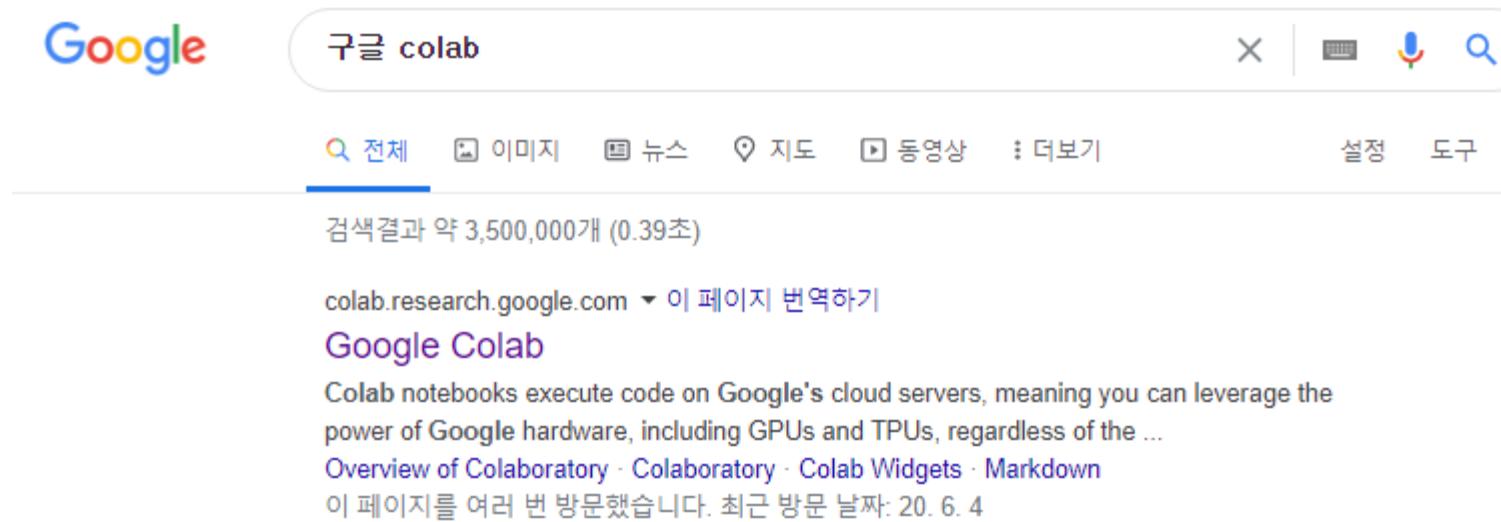
- 아래의 수업 자료 중 핵심 내용 중 일부를 발췌한 것으로 세미나 참여 인원들만 공유드립니다. 🚨
- 자료는 세미나 참석 인원 전원에게 수업이 끝나고나서 일부 비공개 후 공유드릴 예정입니다.
- 현재 진행되고 있는 오프라인 수업도 많은 관심부탁드립니다.
- 오프라인 수업 링크: <https://learningspoons.com/course/detail/llm-master/>



**RAG 성능 고도화를 위한
LLM 파인 투닝과 프롬프팅**

: Llama3, CoT, 에이전트

파이썬 실습 환경: Colab



- 구글 검색창에 구글 colab이라고 검색.
- 설치가 필요하지 않으며 GPU를 사용 가능한 무료 개발 환경
- colab 사용 방법 : https://tykimos.github.io/2019/01/22/colab_getting_started/

LLM에서의 Tokenization

딥 러닝 Tokenizer (KoGPT-2)

입력 문장

“열심히 코딩한 당신, 연휴에는 여행을 가봐요”

코드

```
from transformers import PreTrainedTokenizerFast  
  
tokenizer = PreTrainedTokenizerFast.from_pretrained("skt/kogpt2-base-v2")  
  
tokenizer.tokenize("열심히 코딩한 당신, 연휴에는 여행을 가봐요")
```

결과 (형태소 분석이 아닌 알고리즘 사용)

koGPT : ['__열심히', '__코', '딩', '한', '__당', '신', '__연', '휴', '에는', '__여행을', '__가', '봐', '요']

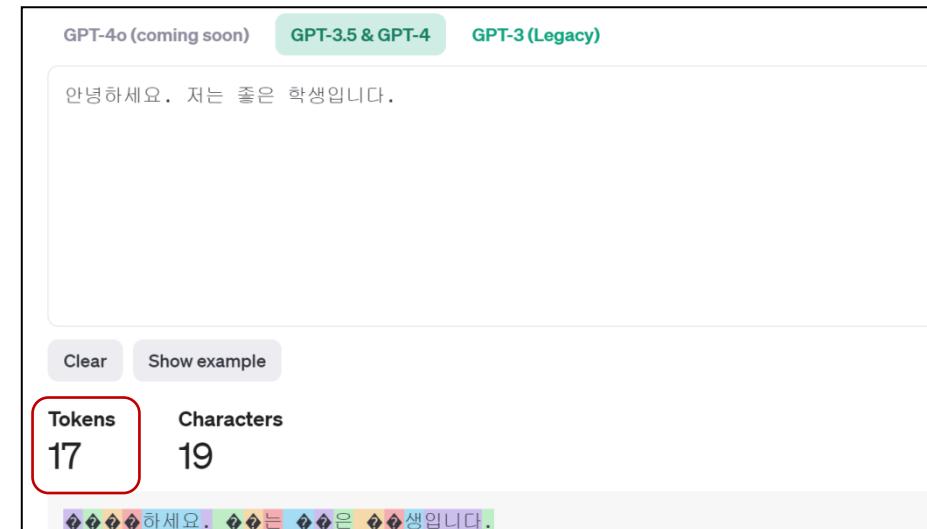
토크나이저(Tokenizer)

- 토크나이저란 언어 모델(Language Model)이 텍스트를 분할하기 위해 사용하는 도구 or 방식을 의미.
- 언어 모델마다 동일한 텍스트라도 토큰 분리 방식이 다르며 효율도 다르다.
- 예를 들어 OpenAI의 GPT만 하여도 최근 GPT 토크나이저는 이전 GPT보다 한글을 더 효율적으로 분할한다.

GPT-3 토크나이저 (과거 버전)



GPT-3.5 토크나이저 (최신 버전)



- 동일한 문장에 대해서 GPT-3 대비 ChatGPT(GPT-3.5)가 토큰 분리 효율이 좋다고 볼 수 있다.
- GPT-3은 40개의 토큰 Vs. GPT-3.5는 17개의 토큰

링크: <https://platform.openai.com/tokenizer>

토크나이저(Tokenizer)

- 동일한 텍스트에 대해서 더 적은 양의 토큰으로 분할되는 것이 중요한 이유는 크게 두 가지이다.
- 언어 모델의 생성 속도에 영향을 미친다. (언어 모델은 토큰을 1개씩 생성하면서 글을 완성한다.)
 - 일반적으로 사용한 토큰만큼 과금이 되는 구조이다.

GPT-4o

GPT-4o is our most advanced multimodal model that's faster and cheaper than GPT-4 Turbo with stronger vision capabilities. The model has 128K context and an October 2023 knowledge cutoff.

[Learn about GPT-4o ↗](#)

Model	Input	Output
gpt-4o	US\$5.00 / 1M tokens	US\$15.00 / 1M tokens
gpt-4o-2024-05-13	US\$5.00 / 1M tokens	US\$15.00 / 1M tokens

- GPT-4o의 경우, 입력에 100만 토큰을 사용하면 5달러, 출력에 100만 토큰을 사용하면 15달러를 과금한다.

GPT-4o Vs. GPT-3.5-turbo Vs. CLOVA-X

- 동일한 텍스트 입력에 대해서 토큰 효율 및 비용을 따져본다면? 네이버가 상대적으로 한글 토큰 효율이 좋은 것을 알 수 있다.
- 물론, 토큰 효율과 LLM의 성능을 모두 고려하여 결정해야 할 것이다.

대한민국의 싱어송라이터이자 배우. 2008년 9월 18일, 중학교 3학년이던 만 15세의 나이에 가수로 데뷔했다. 예명인 '아이유'는 '너와 내가 음악으로 하나가 된다'라는 뜻을 가지고 있다. 매력적인 음색과 뛰어난 작사 · 작곡 능력을 바탕으로 솔로 아이돌이자 아티스트로서 십수 년째 사랑 받고 있을 뿐 아니라 2012년 이래로 매년 국내 및 아시아의 주요 도시에서 대규모 콘서트를 진행하며 공연자로서도 활발히 활동 중이다.

< GPT-4o >

토큰수 : 154

토큰당 가격 : 0.0137원

전체 비용 : 2.1원

< GPT-3.5-turbo >

토큰수 : 250

토큰당 가격 : 0.00137원

전체 비용 : 0.34원

< 클로바X >

토큰수 : 108

토큰당 가격 : 0.005원

전체 비용 : 0.54원

토크나이저(Tokenizer)

- OpenAI는 영어 데이터를 훨씬 많이 학습하여 영어가 상대적으로 훨씬 토큰 소모가 적다. (비용 저렴)
- 이는 학습 시 훨씬 많이 관측한 단어일수록 토큰의 우선 순위가 높아지는 Byte Pair Encoding 방식 때문.

GPT-4o 토크나이저 (영어)

GPT-4o & GPT-4o mini GPT-3.5 & GPT-4 GPT-3 (Legacy)

compresses for quick recovery, and as a result, she has never been seriously injured. Although she once suffered from both hyperthyroidism and hypothyroidism, running helped alleviate these conditions. She now sees staying injury-free as her main goal and continues to run with a bright smile.

In 2017, after running the Guam Marathon together, Mok received a marriage proposal from Lee. Photo courtesy of Mok Young-joo.

[Clear](#) [Show example](#)

Tokens Characters
1,432 6408

I met my husband on a day when our marathon club had a gathering. From that moment on, we trained together and participated in races. I was afraid to challenge myself to run a full marathon, but my husband kept entering full marathons and consistently placing in the top ranks. I wondered what it felt like to run like that, and I thought I wouldn't be able to fully understand him until I completed a full marathon myself. So, I decided to take on the challenge.

Mok Young-joo is seen sprinting in a marathon. She started running in 2009 and, after meeting her husband Lee Byung-do at a marathon club, married him in 2017. Since then, the couple has participated in races both domestically and internationally, leading a healthy lifestyle together. Photo courtesy of Mok Young-joo.

Mok Young-joo, an office worker, first challenged herself to run the 42.195 km full marathon in the fall of 2016. At the time, her then-boyfriend and now-husband Lee Byung-do paced her, and she finished with a time of 3 hours and 47 minutes. From that point on, she became deeply engrossed in marathons. Mok and Lee, whose relationship started through their passion for running, got married in 2017 and continue to

GPT-4o 토크나이저 (한글)

GPT-4o & GPT-4o mini GPT-3.5 & GPT-4 GPT-3 (Legacy)

2017년 괌마라톤을 함께 달린 뒤 목영주 씨(오른쪽)가 남편 이병도 씨의 프로포즈를 받고 있는 모습. 목영주 씨 제공.

2017년 괌마라톤을 함께 달린 뒤 목영주 씨(오른쪽)가 남편 이병도 씨의 프로포즈를 받고 있는 모습. 목영주 씨 제공.

[Clear](#) [Show example](#)

Tokens Characters
2,778 4413

목 씨는 지난해부터 남편과 함께 '◆◆도달자'를 운영하고 있다. 그는 "오랜 기간 달려온 터에 많은 응원을 받아 그에 대해 무언으로 보답할지 고민하다 려닝 비주기 때 한시적으로 누구나 참여할 수 있는 럭닝 ◆◆드립 프로그램을 만들었다"고 했다. 일종의 재능기부 자원봉사 프로그램이다. ◆◆드립은 속도를 전천히 시작해 km마다 점점 빠르게 뛰는 ◆◆된다. 남편은 플코스 '서브스리(3시간 이내 기록)'을 목표로 하는 '급행달자'를 운영하고, 목 씨는 10km 50분 이내를 목표로 달리는 '원행달자'를 운영한다.

목 씨는 이전 오솔로다는 ◆◆런(즐기며 달리기)에 초점을 ◆◆다. ◆◆런도 무장 방지에 중점을 준다. 달리기 전 스트레칭 세션을 많이 해주고 달리는 러 ◆◆을 살려주는 스◆◆(Skip) 등 보조운동도 많이 한다. 목 ◆◆ 및 발목 부근 근육을 강화하는 근육 운동도 자주한다. 달리고 단 뒤 회복을 빠르게 하기 위해 ◆◆◆◆질도 한다. 그래서 아직 달리다 마친 적은 없다. 한때 감상선 기능 저하증과 기능 저하증이 동시에 나타난 적도 있지만 달리면서 증세가 사라졌다. 목 씨는 "오래 달리라면 딱하면 안 된다. 이전 목표가 딱치지 않고 달리는 것"이라며 활짝 웃었다.

2017년 ◆◆마라톤을 함께 달린 뒤 목영주 씨(오른쪽)가 남편 이병도 씨의 프로포즈를 받고 있는 모습. 목영주 씨 제공.

2017년 ◆◆마라톤을 함께 달린 뒤 목영주 씨(오른쪽)가 남편 이병도 씨의 프로포즈를 받고 있는 모습. 목영주 씨 제공.

링크: <https://platform.openai.com/tokenizer>

실험한 뉴스 기사: <https://n.news.naver.com/article/020/0003588260>

토크나이저(Tokenizer)

- 따라서 많은 호출을 요하는 경우, 프롬프트를 영어로 작성하는 것은 과금 절약에 도움이 될 수 있다.
- 특히 ChatGPT가 아닌 오픈 모델(라마3, Qwen, Gemma 등….)에서는 영어 프롬프트가 성능에 영향을 줌.

```
def return_answer(query, input_text=''):  
    system_role = f"""You are an AI language model "비타민봇" whose expertise is reading and summarizing the key contents of the document when given.  
you should take a given document and return a very detailed summary of the document in the query language.  
We recommend that you refuse to respond to requests for specific tasks other than the summary.  
  
Here are the document:  
Long Document : """ + str(input_text) + """  
  
You must return it in Korean. Please respond politely and formally.  
Returns accurate answers based on document.  
"""  
  
user_content = f"""Given the question: "{str(query)}". """  
  
response = client.chat.completions.create(  
model="gpt-3.5-turbo",  
messages=[  
    {"role": "system", "content": system_role},  
    {"role": "user", "content": user_content}  
]  
)  
return response.choices[0].message.content
```

LLM API 가격 정리

- LLM API의 가격을 정리해놓은 사이트: <https://llm-price.com/>
- 예를 들어 RAG를 하는 경우, Cohere가 상대적으로 저렴한 선택일 수 있다.

MODEL NAME	PROVIDERS	1 M INPUT TOKENS	1 M OUTPUT TOKENS	SOURCE	UPDATED TIME
command-r+		\$3.00	\$15.00	Cohere	Apr 22, 2024, 08:45:31 AM
fine-tuned-command-r		\$2.00	\$4.00	Cohere	May 22, 2024, 01:52:15 AM
fine-tuned-model		\$2.00	\$4.00	Cohere	May 15, 2024, 01:54:36 AM



LLM Pricing

LLM Pricing is a website that aggregates and compares the pricing information for various Large Language Models (LLMs) offered by official AI providers and cloud service vendors.

Designed, developed and updated by Claude 3 Sonnet, prompted by huhuhang

MODEL NAME	PROVIDERS	1 M INPUT TOKENS	1 M OUTPUT TOKENS	SOURCE	UPDATED TIME
claude-3-haiku		\$0.25	\$1.25	AWS	Mar 16, 2024, 06:58:17 AM
claude-3-haiku		\$0.25	\$1.25	Anthropic	Mar 16, 2024, 06:58:09 AM
mistral-7b		\$0.25	\$0.25	Mistral	Mar 16, 2024, 06:58:27 AM

Encoding & Padding

Integer Encoding

- 토큰화를 수행했다면 각 단어에 고유한 정수를 된다. 이때 중복은 허용하지 않는다.
- 중복이 허용되지 않는 모든 단어들의 집합을 **단어 집합(Vocabulary)**이라고 한다.
- LLM이 기본적으로 토큰으로 나누는 이유는 결국 정수 **인코딩**을 하기 위함이다.

문서1 : [먹고, 싶은, 사과]
문서2 : [먹고, 싶은, 바나나]
문서3 : [길고, 노란, 바나나, 바나나]
문서4 : [저는, 과일이, 좋아요]

토큰화 후라고 가정.

Integer encoding

과일이	1
길고	2
노란	3
먹고	4
바나나	5
사과	6
싶은	7
저는	8
좋아요	9

단어 집합

문서에 재적용

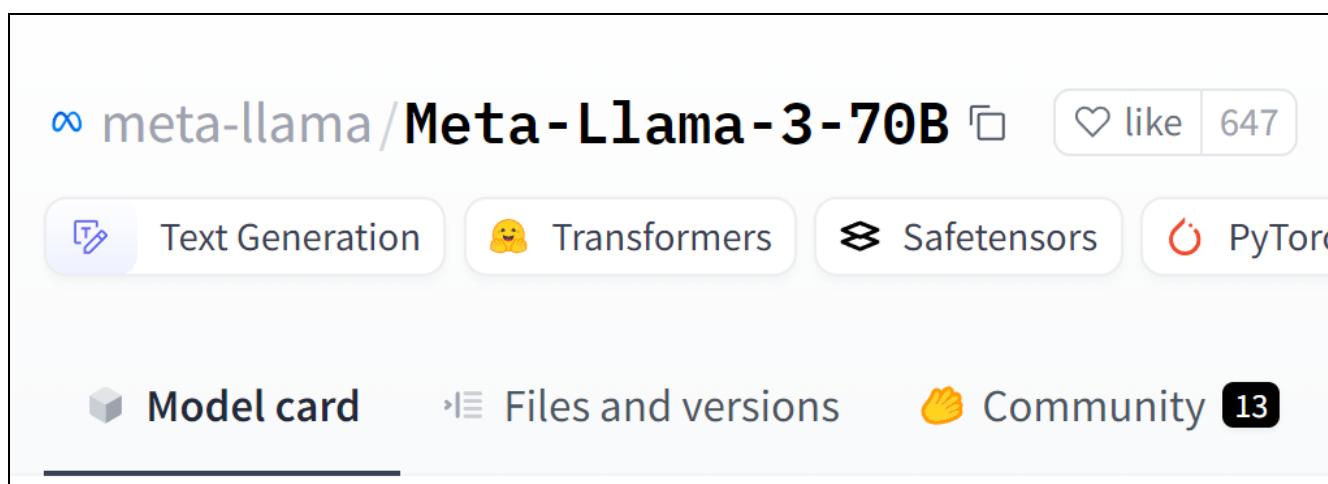
문서1 : [4, 7, 6]
문서2 : [4, 7, 5]
문서3 : [2, 3, 5, 5]
문서4 : [8, 1, 9]

단어 집합을 통해 인코딩.

모델의 파라미터

파라미터 수 이해하기

- 딥 러닝 언어 모델의 파라미터 개수(인간의 뇌세포에 비유)를 말할 때는 10억 => B라는 표현을 사용함.
- 파라미터 수가 클 수록 좀 더 좋은 성능을 가지는 것이 일반적이다. 하지만 그만큼 더 많은 GPU를 필요로 한다.
- GPT-3의 파라미터 개수는 1,750억개 => 175B
- LLaMA-3의 파라미터 개수는 700억개 => 70B
- 온라인에 공개된 B라는 표현으로부터 파라미터 개수를 이해할 수 있어야 한다.



<https://huggingface.co/meta-llama/Meta-Llama-3-70B>

오픈 LLM 살펴보기

LLM 리더보드

- 공개된 LLM을 확인할 수 있는 곳으로 LLM 리더보드 등이 존재.
- <https://huggingface.co/spaces/upstage/open-ko-llm-leaderboard>
- 많은 모델이 오염(테스트 데이터에 맞추어서 학습)되어져 있어서 참고만 하는 것이 필요.

T	Model	Average	Ko-ARC	Ko-HellaSwag	Ko-MMLU	Ko-TruthfulQA
○	chihoonlee10/T30-ko-solar-dpo-v7.0	70.71	79.1	81.5	57.14	83.17
○	MoaData/Myrrh_solar_10.7b_3.0	70.62	78.33	81.04	57.25	82.41
○	chihoonlee10/T30-ko-solar-dpo-v6.0	70.6	77.47	81.19	56.28	84.23
○	hwkwon/S-SOLAR-10.7B-v1.5	70.35	76.28	80.85	56.09	84.33
○	sdhan/SD_SOLAR_10.7B_v1.0	70.31	78.24	78.49	55.48	87.38
○	freewheelin/free-solar-evo-v0.11	70.31	77.22	81.27	56.58	83.81
○	chihoonlee10/T30-ko-solar-dpo-v5.0	70.3	76.19	80.96	56.09	84.32
○	moondriller/anarchy-solar-10B-v1	70.28	76.88	81.25	56.25	83.88
○	freewheelin/free-solar-evo-v0.13	70.26	76.71	81.09	56.78	84.06
○	MoaData/Myrrh_solar_10.7b_2.0	70.24	78.24	80.99	57.24	82.88

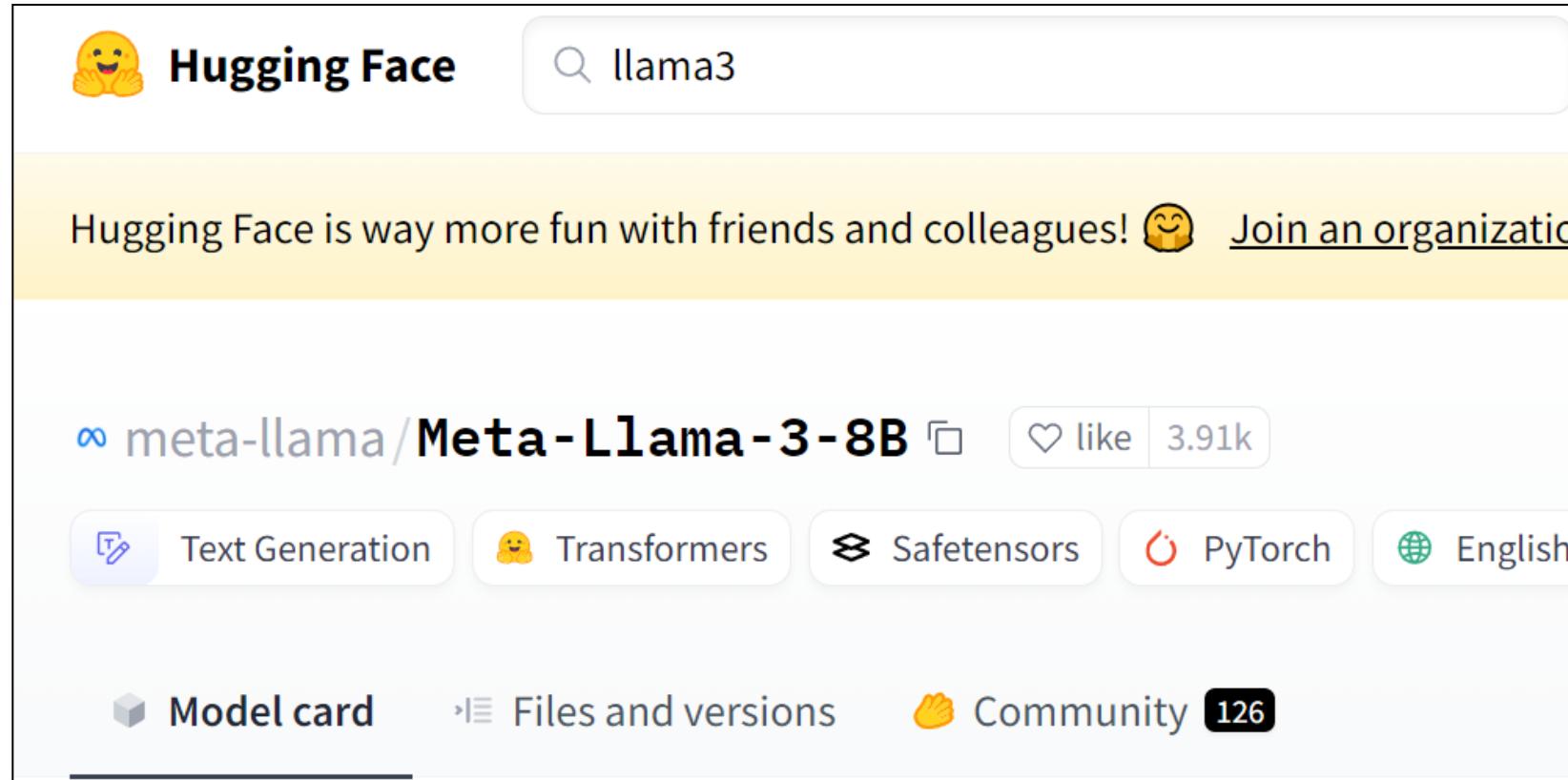
Logickor

- 이곳도 여전히 참고만... 압도적인 모델 크기의 격차는 언제나 성능과 연관되어져 있음.
- 예를 들어서 여기서 8B가 70B보다 점수가 높다고 하더라도 실제 문제에선 70B가 점수가 좋을 가능성이 높음.
- <https://lk.instruct.kr/>

종합	싱글턴	멀티턴											
순위	모델명		추론	수학	글쓰기	코딩	이해	문법	싱글턴	멀티턴	총점	평가로그	노트
1	claude-3-opus-20240229	Anthropic	8.42	9.21	9.71	9.14	10.00	7.92	8.80	9.33	9.07	평가로그	elo 아레나 랭킹 2위
2	gpt-4-turbo-2024-04-09	OpenAI	9.07	9.85	9.78	9.50	9.14	6.42	9.07	8.85	8.96	평가로그	(2024-04-09 기준) gpt-4-turbo default 모델
3	gpt-4-1106-preview	OpenAI	9.14	8.14	9.07	9.00	10.00	6.92	8.76	8.66	8.71	평가로그	elo 아레나 랭킹 1위
4	gpt-4o-2024-05-13	OpenAI	6.57	9.28	9.57	8.92	9.92	7.92	9.07	8.33	8.70	평가로그	
5	claude-3-sonnet-20240229	Anthropic	7.21	6.42	9.28	8.35	9.85	6.78	7.97	8.00	7.98	평가로그	elo 아레나 랭킹 3위
6	HyperClovaX	NAVER	5.85	7.14	8.50	7.57	9.50	8.50	8.40	7.28	7.84	평가로그	
7	gpt-3.5-turbo-0125	OpenAI	7.35	6.78	8.78	7.35	9.57	6.50	7.50	7.95	7.72	평가로그	elo 아레나 랭킹 9위

오픈 LLM 사용하기

- 기본적으로 오픈된 LLM은 허깅페이스(<https://huggingface.co/>)라는 웹 사이트에서 제공.
- 예를 들어서 Meta에서 공개된 LLama3를 사용하고 싶다면 검색하면 해당 모델을 찾을 수 있음.



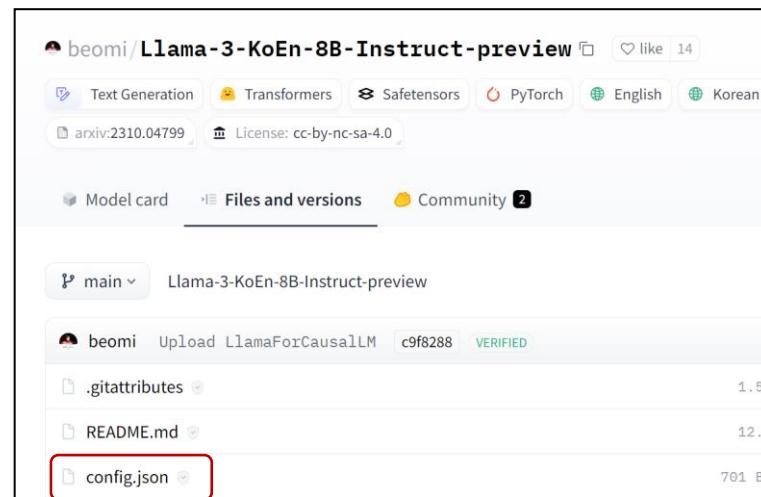
오픈 LLM 정보 확인하기

- 기본적으로 오픈된 LLM은 허깅페이스(<https://huggingface.co/>)라는 웹 사이트에서 제공.
- 사용하고자 하는 모델을 찾았다면 해당 모델 웹 페이지에서 [Files and versions]를 클릭.



오픈 LLM 정보 확인하기

- 기본적으로 오픈된 LLM은 허깅페이스(<https://huggingface.co/>)라는 웹 사이트에서 제공.
- 사용하고자 하는 모델을 찾았다면 해당 모델 웹 페이지에서 [Files and versions]를 클릭.
- 그 후 [config.json]을 클릭.

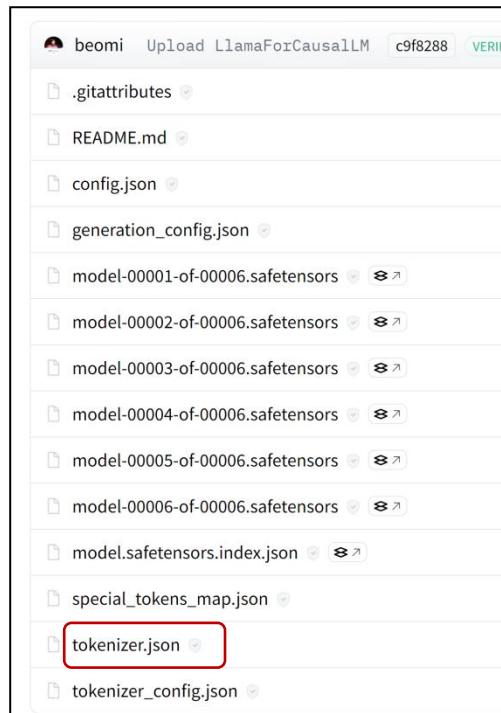


```
1  {
2      "_name_or_path": "beomi/Llama-3-KoEn-8B-preview",
3      "architectures": [
4          "LlamaForCausalLM"
5      ],
6      "attention_bias": false,
7      "attention_dropout": 0.0,
8      "bos_token_id": 128000,
9      "eos_token_id": 128001,
10     "hidden_act": "silu",
11     "hidden_size": 4096,
12     "initializer_range": 0.02,
13     "intermediate_size": 14336,
14     max_position_embeddings": 8192,
15     "model_type": "llama",
16     "num_attention_heads": 32,
17     "num_hidden_layers": 32,
18     "num_key_value_heads": 8,
19     "pretraining_tp": 1,
20     "rms_norm_eps": 1e-05,
21     "rope_scaling": null,
22     "rope_theta": 500000.0,
23     "tie_word_embeddings": false,
24     "torch_dtype": "bffloat16",
25     "transformers_version": "4.35.2",
26     "use_cache": true,
27     "vocab_size": 128256
28 }
29
```

- 모델이 처리 가능한 길이: 8,192
- 모델이 구사 가능한 토큰(단어) 종류: 128,256
- 이 모델은 한번의 호출 시 입력 토큰과 출력 토큰을 포함하여 한 번에 총 8,192의 개수까지 커버 가능합니다.
- 예를 들어서 입력으로 총 7,000개의 토큰을 넣었다면 이 모델은 1,192개의 토큰까지 답변에 사용할 수 있으며 그 이상의 길이의 답변을 작성할 수 없습니다.
- 예를 들어 길이 9,000의 입력은 입력받지 못합니다.

오픈 LLM 정보 확인하기

- 기본적으로 오픈된 LLM은 허깅페이스(<https://huggingface.co/>)라는 웹 사이트에서 제공.
- 사용하고자 하는 모델을 찾았다면 해당 모델 웹 페이지에서 [Files and versions]를 클릭.
- 그 후 [tokenizer.json]을 클릭.



```
"jamin": 26312,
"GSB": 26313,
"etermination": 26314,
" ");": 26315,
"Eng": 26316,
"vos": 26317,
"Ginhab": 26318,
"/lang": 26319,
"burgh": 26320,
"Executor": 26321,
"hone": 26322,
"Challenge": 26323,
"_links": 26324,
".Level": 26325,
"Under ground": 26326,
```

단어 집합

- tokenizer.json 파일을 열어보면 토큰(단어)와 각 정수의 맵핑 표가 나오게 됩니다.
- LLM은 내부적으로 모든 토큰(단어)를 각각 정수로 맵핑하여 사용합니다.
- 예를 들어 jamin이라는 단어가 입력으로 들어오면 26312라는 정수로 바뀐 후에 LLM의 입력으로 사용됩니다.
- 이렇게 단어 -> 정수로 바꾸는 과정을 encoding이라고 표현합니다.
- 그리고 이러한 토큰(단어)가 총 128,256개가 있을 것입니다. (config.json에서 확인.)

LLM 호출 및 서빙 가속화

<LLM>

- LLaMA3와 Qwen 2.5 Tokenizer:

https://colab.research.google.com/drive/1qF3ZiwOe6SS_8JnHoPI4iZY3eNZGTpGW?usp=sharing

- 한국어 LLaMA3-8B를 이용한 RAG

https://colab.research.google.com/drive/1otG_a3lC0p5xVN5YJePPqBoiWYMG3Yrz?usp=sharing

- vLLM을 이용한 서빙 속도 가속화

<https://colab.research.google.com/drive/1IXWkzOnIZdirMFTHrsIfirLtlaprJPka?usp=sharing>

LLM 호출 및 서빙 가속화

<LLM>

- LLaMA3와 Qwen 2.5 Tokenizer:
https://colab.research.google.com/drive/1qF3ZiwOe6SS_8JnHoPI4iZY3eNZGTpGW?usp=sharing
- 한국어 LLaMA3-8B를 이용한 RAG
https://colab.research.google.com/drive/1otG_a3lCOp5xVN5YJePPqBoiWYMG3Yrz?usp=sharing
- vLLM을 이용한 서빙 속도 가속화
<https://colab.research.google.com/drive/1IXWkzOnIZdirMFTHrsIfirLtlaprJPka?usp=sharing>

세미나에서는 시간 상 다루지 못하지만 (온라인 수업에서는 다룹니다.)

- 실제 서비스 시에는 vLLM + 멀티 로라 + Fast API 구성으로 Serving을 많이 하는 편입니다.
- 혹시 LLM을 실제 서비스에 다루셔야 하는데 vLLM을 처음 들으신다면 한 번 검색해서 공부해보시길 권합니다.