



수강신청이 어려운 세종대 학생들을 위한 도우미

수강편람은 왜 어려울까?

처음 겪어보는 '수강신청'

익숙하지 않은 대학교 문법

방대한 문서에 흩어져 있는 자료

CRA봇과 함께라면?

처음 겪어보는 '수강신청'

대화를 통해 보다 쉽게 접근

익숙하지 않은 대학교 문법

익숙한 일상어로 정보 제공

방대한 문서에 흩어져 있는 자료

페이지 수를 함께 제공해 사용자가 Cross-check 가능

목적은 결국 "데이터 리터러시 향상"

기능

- 수강편람을 벡터DB에 업로드하여, 질문과 유사한 정보를 검색하여 LLM 답변에 포함
- 정보가 명시된 페이지 수를 함께 출력하여, 사용자의 Cross-check 지원

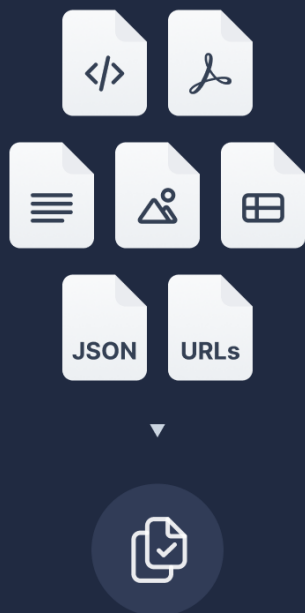
Stacks

- UI: Streamlit
- Document Loader: UpstageDocumentParserLoader
- Vector DB: Chroma DB
- Retriever: MultiVectorRetriever + BM25 Retriever Ensemble
- Embedding: Upstage Embedding "embedding-passage"
- LLM: Google Gemini-1.5-flash
- 기술 구현: LangChain

What is RAG?

Retrieval Augmented Generation

LOAD



SPLIT

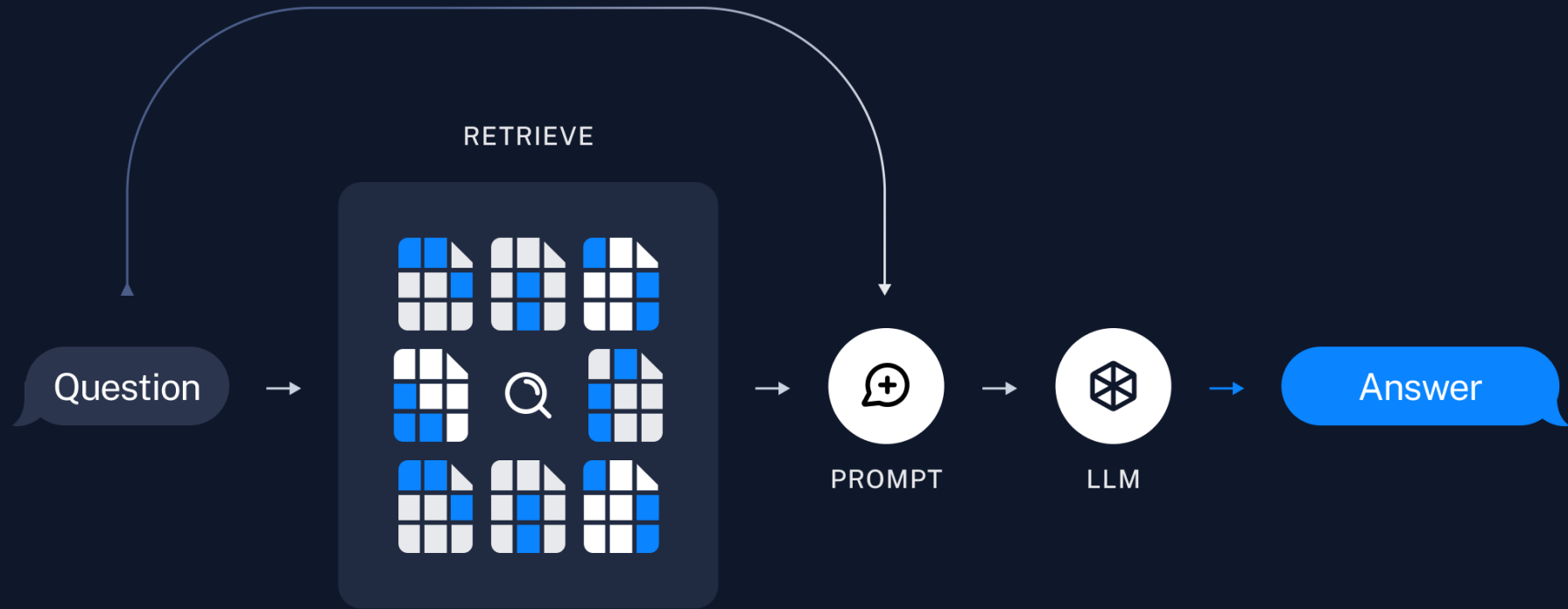


EMBED



STORE





Loader

UpstageDocumentParseLoader

Parser? Loader?

Solar 1.7B

Upstage AI

Figure

caption

Figure 1: Depth up-scaling for the case with $n = 32$, $s = 48$, and $m = 8$. Depth up-scaling is achieved through a dual-stage process of depthwise scaling followed by continued pretraining.

paragraph

for wider access and application of these models by researchers and developers globally.

2

Depth Up-Scaling

heading1

paragraph

To efficiently scale-up LLMs, we aim to utilize pre-trained weights of base models to scale up to larger LLMs (Komatsuzaki et al., 2022). While existing methods such as Komatsuzaki et al. (2022) use MoE (Shazeer et al., 2017) to scale-up the model architecture, we opt for a different depthwise scaling strategy inspired by Tan and Le (2019). We then continually pretrain the scaled model as just scaling the model without further pretraining degrades the performance.

paragraph

Base model. Any n -layer transformer architecture can be used but we select the 32-layer Llama 2 architecture as our base model. We initialize the Llama 2 architecture with pretrained weights from Mistral 7B, as it is one of the top performers compatible with the Llama 2 architecture. By adopting the Llama 2 architecture for our base model, we aim to leverage the vast pool of community resources while introducing novel modifications to further enhance its capabilities.

paragraph

Depthwise scaling. From the base model with n layers, we set the target layer count s for the scaled model, which is largely dictated by the available hardware.

paragraph

With the above, the depthwise scaling process is as follows. The base model with n layers is duplicated for subsequent modification. Then, we remove the final m layers from the original model and the initial m layers from its duplicate, thus forming two distinct models with $n - m$ layers. These two models are concatenated to form a scaled model with $s = 2 \cdot (n - m)$ layers. Note that $n = 32$ from our base model and we set $s = 48$ considering

our hardware constraints and the efficiency of the scaled model, i.e., fitting between 7 and 13 billion parameters. Naturally, this leads to the removal of $m = 8$ layers. The depthwise scaling process with $n = 32$, $s = 48$, and $m = 8$ is depicted in ‘Step 1: Depthwise Scaling’ of Fig. 1.

paragraph

We note that a method in the community that also scale the model in the same manner ² as ‘Step 1: Depthwise Scaling’ of Fig. 1 has been concurrently developed.

Continued pretraining.

The performance of the depthwise scaled model initially drops below that of the base LLM. Thus, we additionally apply the continued pretraining step as shown in ‘Step 2: Continued Pretraining’ of Fig. 1. Experimentally, we observe rapid performance recovery of the scaled model during continued pretraining, a phenomenon also observed in Komatsuzaki et al. (2022). We consider that the particular way of depthwise scaling has isolated the heterogeneity in the scaled model which allowed for this fast performance recovery.

Delving deeper into the heterogeneity of the scaled model, a simpler alternative to depthwise scaling could be to just repeat its layers once more, i.e., from n to $2n$ layers. Then, the ‘layer distance’, or the difference in the layer indices in the base model, is only bigger than 1 where layers n and $n + 1$ are connected, i.e., at the seam.

However, this results in maximum layer distance at the seam, which may be too significant of a discrepancy for continued pretraining to quickly resolve. Instead, depthwise scaling sacrifices the $2m$ middle layers, thereby reducing the discrepancy at the seam and making it easier for continued

²<https://huggingface.co/Undi95/Mistral-11B-v0.1>

footnote

2

footer

Understanding Basic Mathematical Concepts

INTRODUCTION

This document provides an overview of fundamental mathematical concepts including arithmetic, algebra, geometry, and calculus. These concepts form the basis of higher-level mathematics and are essential for various applications in science, engineering, and everyday life.

KEY TOPICS COVERED:

Arithmetic Operations

Addition

Subtraction

Multiplication

Division

Algebraic Expressions

Variables

Constants

Coefficients

Geometric Figures

Circles

Triangles

Rectangles

Calculus

Differentiation

Integration

INDEX

1. Arithmetic Operations.....1

2. Algebraic Expressions.....2

3. Geometric Figures.....3

4. Calculus.....4

Table 1: Basic Arithmetic Operations

Operation	Symbol	Example	Result
Addition	+	$5 + 3$	8
Subtraction	-	$9 - 4$	5
Multiplication	*	$6 * 7$	42
Division	/	$20 / 4$	5

Equation 1: Quadratic Formula

The quadratic formula is used to solve quadratic equations of the form $ax^2 + bx + c = 0$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Page 1 of 1

footer



Splitter

RecursiveChracterTextSplitter()

Why Split?

Upstage Document Parser로 PDF를 Markdown으로 분류했으나,
페이지 기준으로 Parsing 하여 max_length가 Embedding 모델의 context window를
훌쩍 넘김(4000 <<< 5072)

검색 편의성과 Output Token 절약을 위해 2000으로 split

Embedding

```
UpstageEmbedding(model="passage-embedding")
```

UpstageEmbedding

Monthly spend

[View pricing](#)

Month-to-date total (VAT not included)

\$0 | No limit [Edit](#)

Payment date

2025-02-01

Payment method

mastercard ****_****_****_7621 [Edit](#)

Key name	Quantity	Amount	Discount	Subtotal
CRA	8,262,813	\$8.61	\$0	\$8.61
• Document Parse (prev. Layout Analysis)	779 pages	\$7.79		
• Solar Mini - Embeddings	8,262,034 tokens	\$0.82		
	• Input 8,262,034			

Store

Chroma DB

Chroma DB



chroma

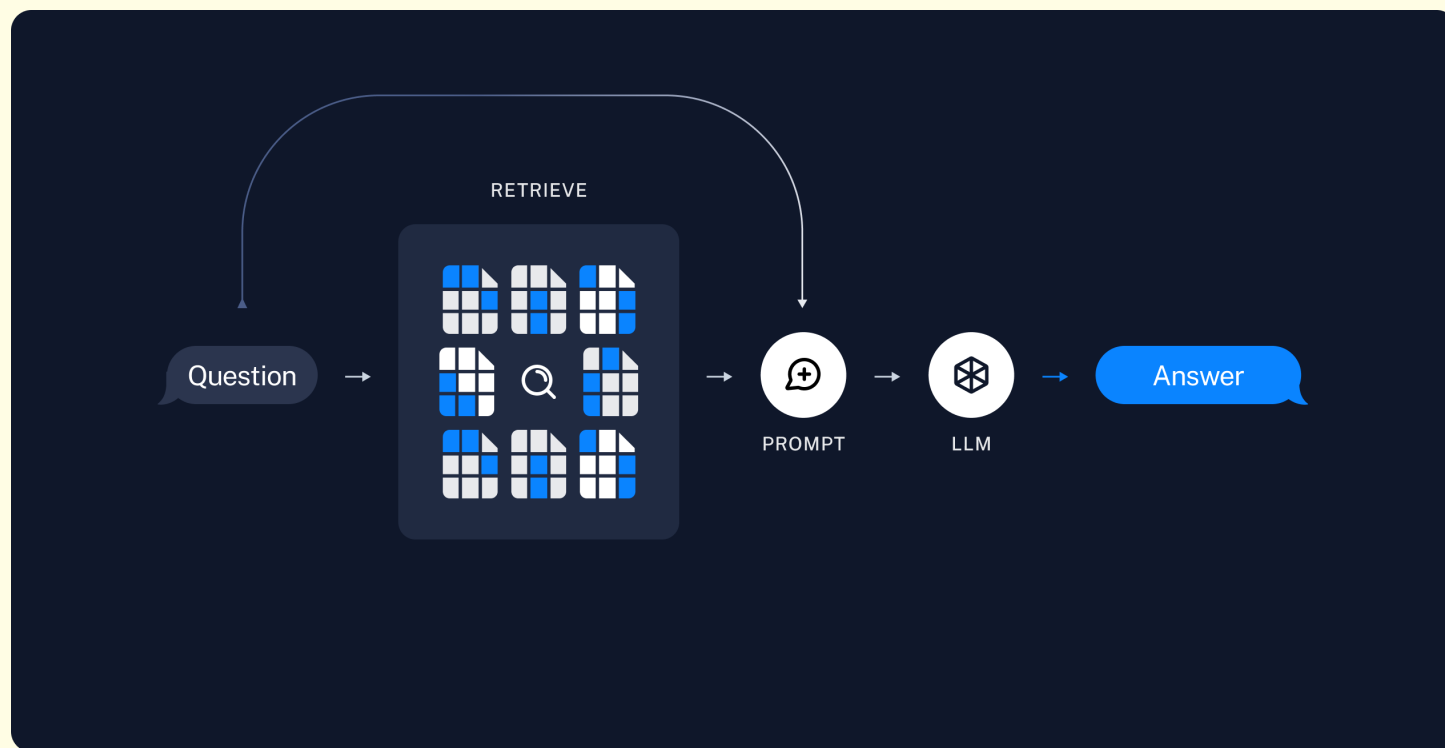
- 오픈소스
- 다양한 Retriever 지원
- 프로덕션에 '일부' 적용 가능...?

Retriever

EnsembleRetriever

BM25 Retriever + MultiVector Retriever

EnsembleRetriever



- Sparse + Dense
- 벡터 차원이 희소할수록 Keyword Search에 강함
- 벡터 차원이 밀집될수록
- Semantic Search에 강함
- 수강편람 검색은 과목명 등 Keyword와 문단 위주의 Semantic 둘 다 필요
- Ensemble 비율은 5:5

LLM

Google Gemini-1.5-flash

Why flash?

총비용(2025년 1월 1일~22일) ②

지난 7일

이번 달

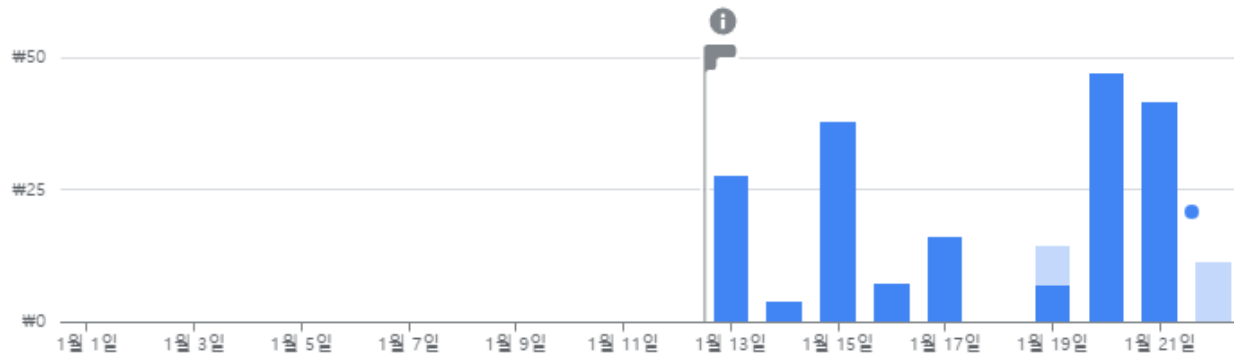
비용 사용된 크레딧 총비용
₩187 ₩0 = ₩187

세부정보 보기

예상 총비용

















₩277

0% vs. 12월



오로지 가격 때문

Performance

GPT-4o (May '24)	 OpenAI	128k	78	\$7.50	77.3	0.66	🔗 Model 🔗 Providers
GPT-4o mini	 OpenAI	128k	73	\$0.26	79.3	0.66	🔗 Model 🔗 Providers
GPT-4o (Nov '24)	 OpenAI	128k	73	\$4.38	97.3	0.50	🔗 Model 🔗 Providers
GPT-4o mini Realtime (Dec '24)	 OpenAI	128k		\$0.00			🔗 Model 🔗 Providers
GPT-4o Realtime (Dec '24)	 OpenAI	128k		\$0.00			🔗 Model 🔗 Providers
Llama 3.3 70B	 Meta	128k	74	\$0.69	72.0	0.46	🔗 Model 🔗 Providers
Llama 3.1 405B	 Meta	128k	74	\$3.50	30.0	0.73	🔗 Model 🔗 Providers
Llama 3.1 70B	 Meta	128k	68	\$0.72	73.3	0.42	🔗 Model 🔗 Providers
Llama 3.2 90B (Vision)	 Meta	128k	68	\$0.90	46.1	0.33	🔗 Model 🔗 Providers
Llama 3.2 11B (Vision)	 Meta	128k	54	\$0.18	132.2	0.28	🔗 Model 🔗 Providers
Llama 3.1 8B	 Meta	128k	54	\$0.10	185.0	0.31	🔗 Model 🔗 Providers
Llama 3.2 3B	 Meta	128k	49	\$0.06	196.4	0.37	🔗 Model 🔗 Providers
Llama 3.2 1B	 Meta	128k	26	\$0.04	314.6	0.35	🔗 Model 🔗 Providers
Gemini 2.0 Flash (exp)	 Google	1m	82	\$0.00	169.0	0.46	🔗 Model 🔗 Providers
Gemini 1.5 Pro (Sep)	 Google	2m	80	\$2.19	60.6	0.61	🔗 Model 🔗 Providers
Gemini 1.5 Flash (Sep)	 Google	1m	74	\$0.13	187.1	0.26	🔗 Model 🔗 Providers

Why flash?

Model

gpt-4o

gpt-4o-2024-11-20

gpt-4o-2024-08-06



사용한 만큼만 지불(가격은 미국 달러(USD) 기준)

Gemini API 사용한 만큼만 지불. 결제 서비스를 사용하여 서비스를 확장하세요. 'API 키'를 설정할 수 있습니다.

RPM (분당 요청) 2,000개
4백만 TPM (분당 토큰 수)

토큰 100만 개당 \$0.075
토큰 100만 개당 \$0.30
토큰 100만 개당 \$0.01875

토큰 100만 개당 \$0.15
토큰 100만 개당 \$0.60
토큰 100만 개당 \$0.0375

컨텍스트 캐싱 (저장소)

시간당 100만 토큰당 \$1.00
[자세히 알아보기](#)

시연