BFS를 사용한 위키 크롤링으로 RAG 파이프라인 구축

20611 유채호 · 24.11.18

특정 도메인의 지식에 특화된 LLM을 만들기 위한 노력들....

RAG

Fine-Tuning

Prompt Engineering

원본 텍스트

"Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum."

chunking

"Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. "

" It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged."

"It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum."

embedding

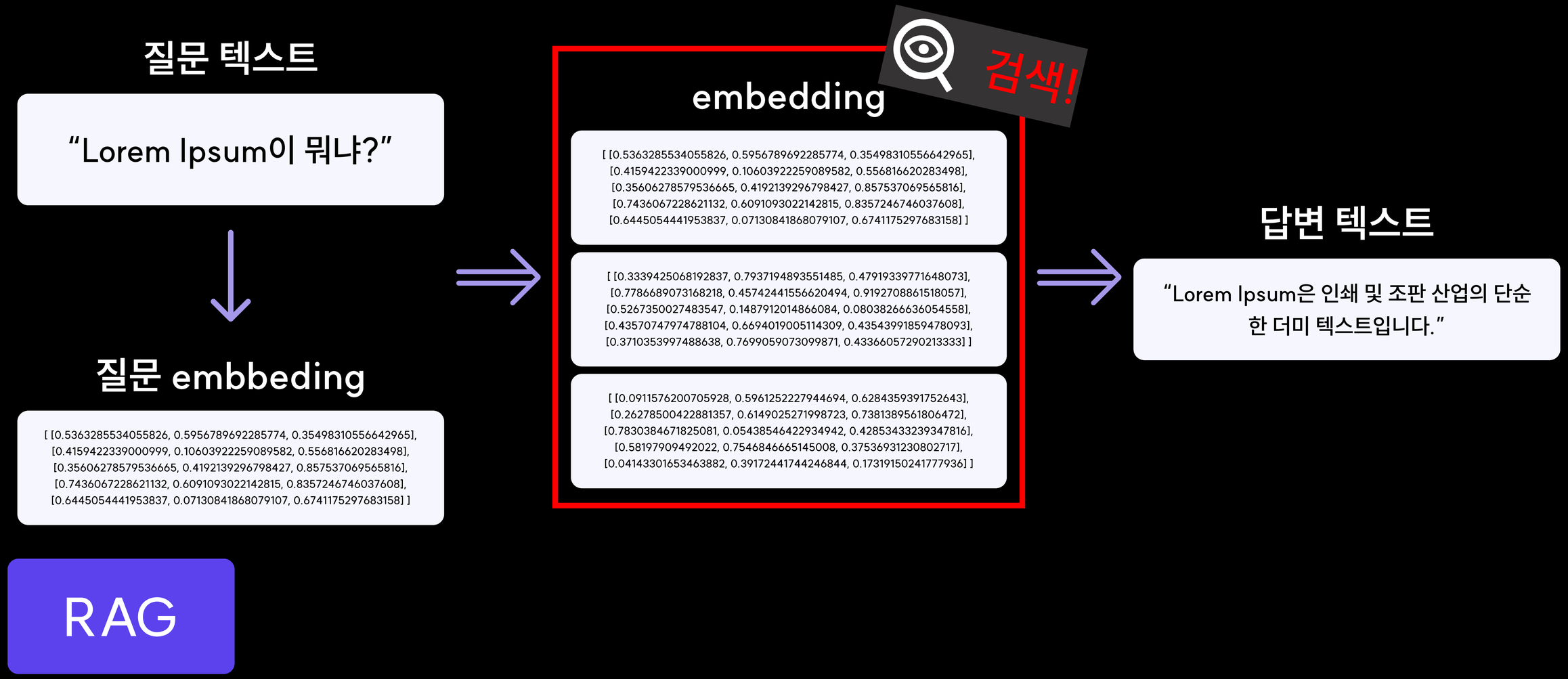
[[0.5363285534055826, 0.5956789692285774, 0.35498310556642965], [0.4159422339000999, 0.10603922259089582, 0.556816620283498], [0.35606278579536665, 0.4192139296798427, 0.857537069565816], [0.7436067228621132, 0.6091093022142815, 0.8357246746037608], [0.6445054441953837, 0.07130841868079107, 0.6741175297683158]]

[[0.3339425068192837, 0.7937194893551485, 0.47919339771648073], [0.7786689073168218, 0.45742441556620494, 0.9192708861518057], [0.5267350027483547, 0.1487912014866084, 0.08038266636054558], [0.43570747974788104, 0.6694019005114309, 0.43543991859478093], [0.3710353997488638, 0.7699059073099871, 0.43366057290213333]]

[[0.0911576200705928, 0.5961252227944694, 0.6284359391752643], [0.26278500422881357, 0.6149025271998723, 0.7381389561806472], [0.7830384671825081, 0.05438546422934942, 0.42853433239347816], [0.58197909492022, 0.7546846665145008, 0.37536931230802717], [0.04143301653463882, 0.39172441744246844, 0.17319150241777936]]

RAG

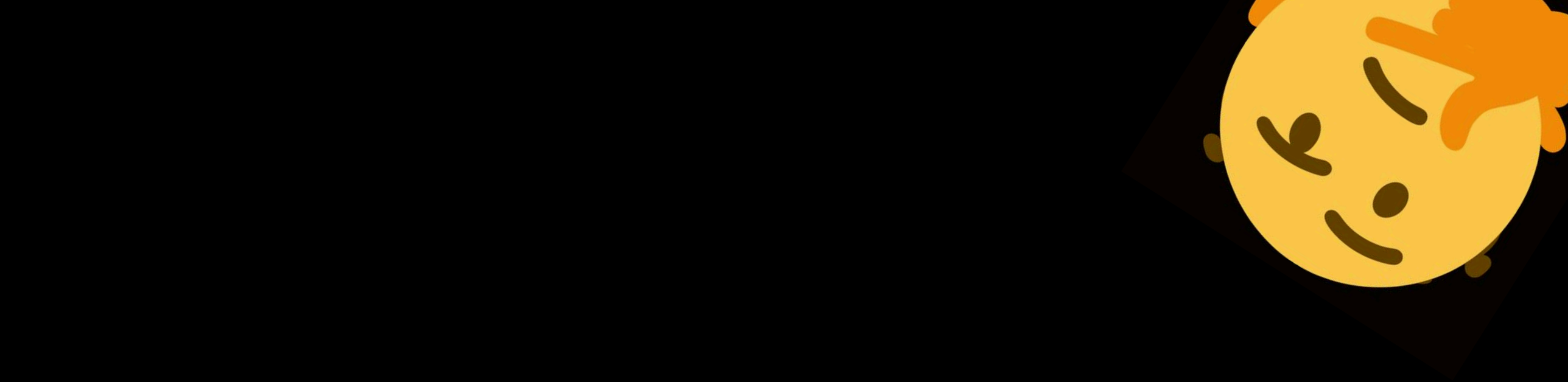
텍스트 데이터를 임베딩 후, 검색하여 QnA 답변 출처에 활용하는 기법



텍스트 데이터를 임베딩 후, 검색하여 QnA 답변 출처에 활용하는 기법

텍스트(문맥 정보를 포함한)만 있으면!

RAG로 많은 양의 지식을 효과적으로 답변 시킬 수 있다!



근데... 많은 양의 텍스트를 다 어디서 구하지?

인터넷 세상의 도서관 “위키”

아하@!!!!!

그냥 위키에서 긁어오면 되겠구나!!!

1. 위키 문서를 각 노드로 취급

2. 문서 내부의 하이퍼링크를 간선으로 취급

BFS로 문서를 탐색하면서

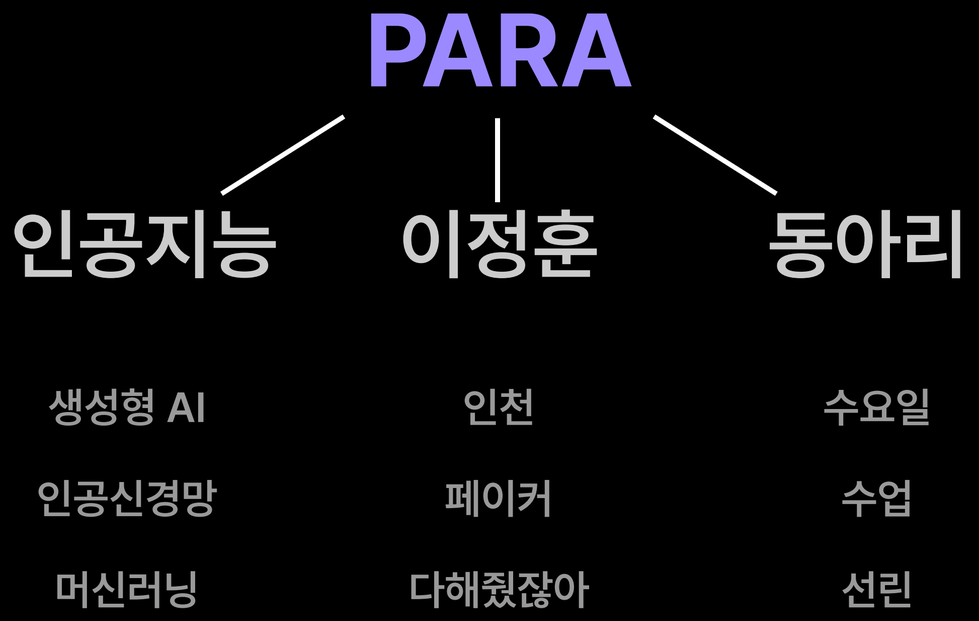
특정 도메인에 대한 텍스트 데이터를 수집!

DFS는 안되나요???

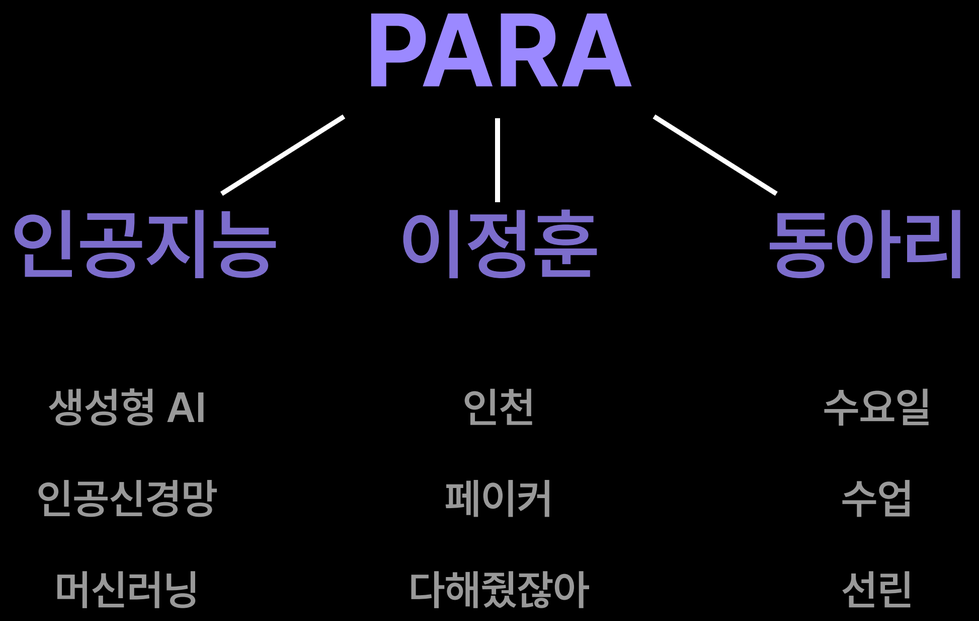
DFS는 안되나요???

당연히 안되죠;;

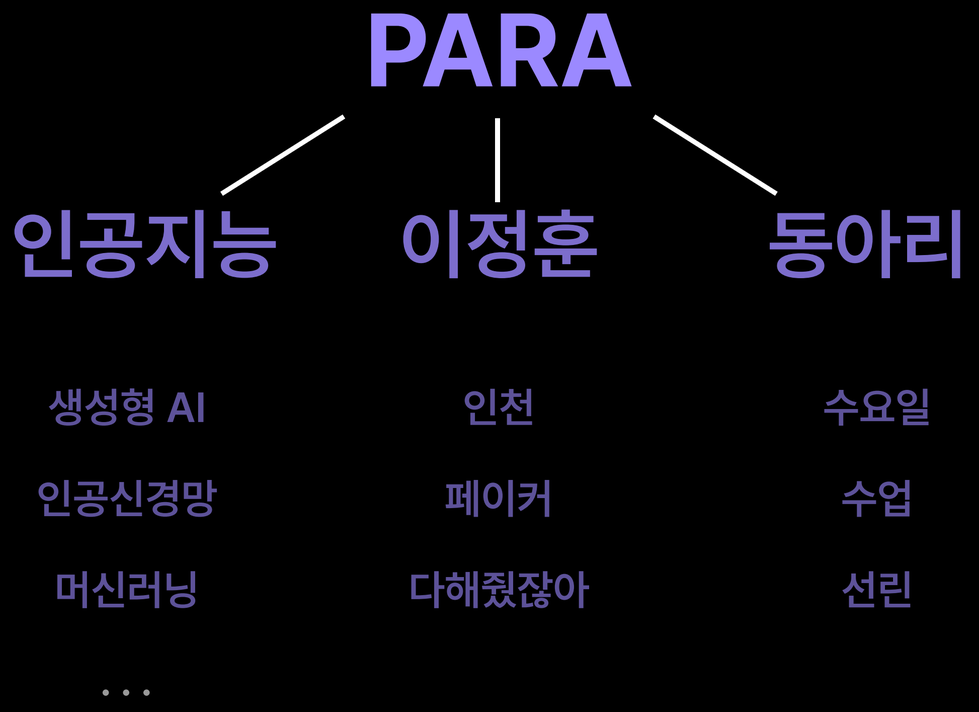
BFS max-depth를 2로 설정했을 경우



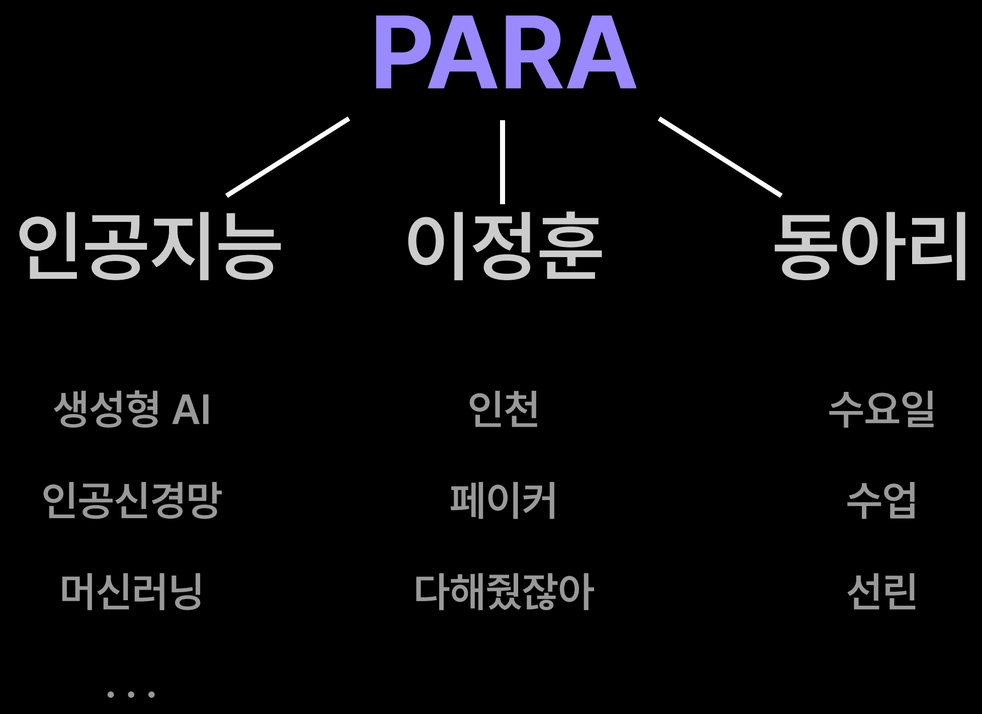
BFS max-depth를 2로 설정했을 경우



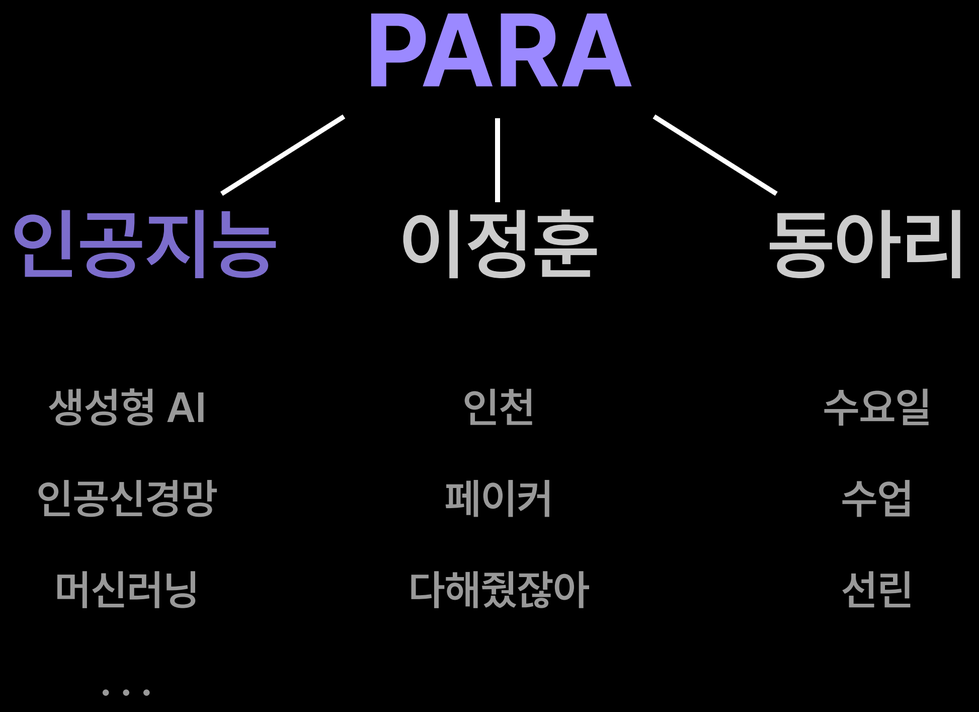
BFS max-depth를 2로 설정했을 경우



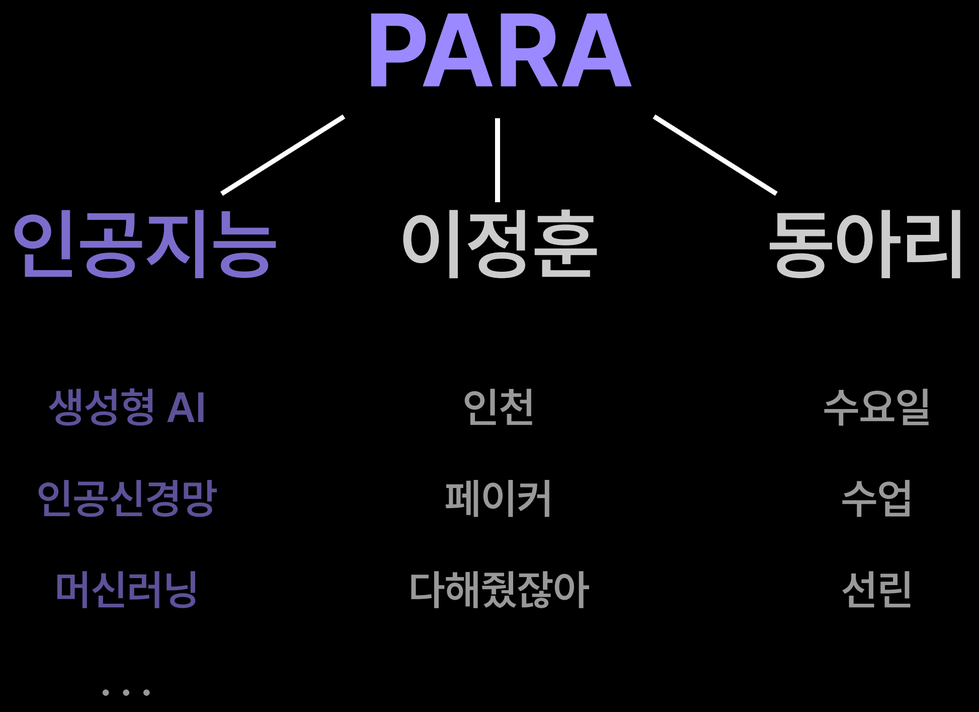
DFS max-depth를 2로 설정했을 경우



DFS max-depth를 2로 설정했을 경우



DFS max-depth를 2로 설정했을 경우

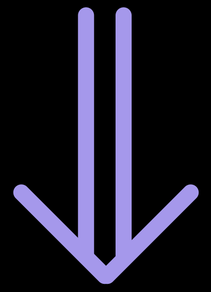


DFS로 탐색했을 경우

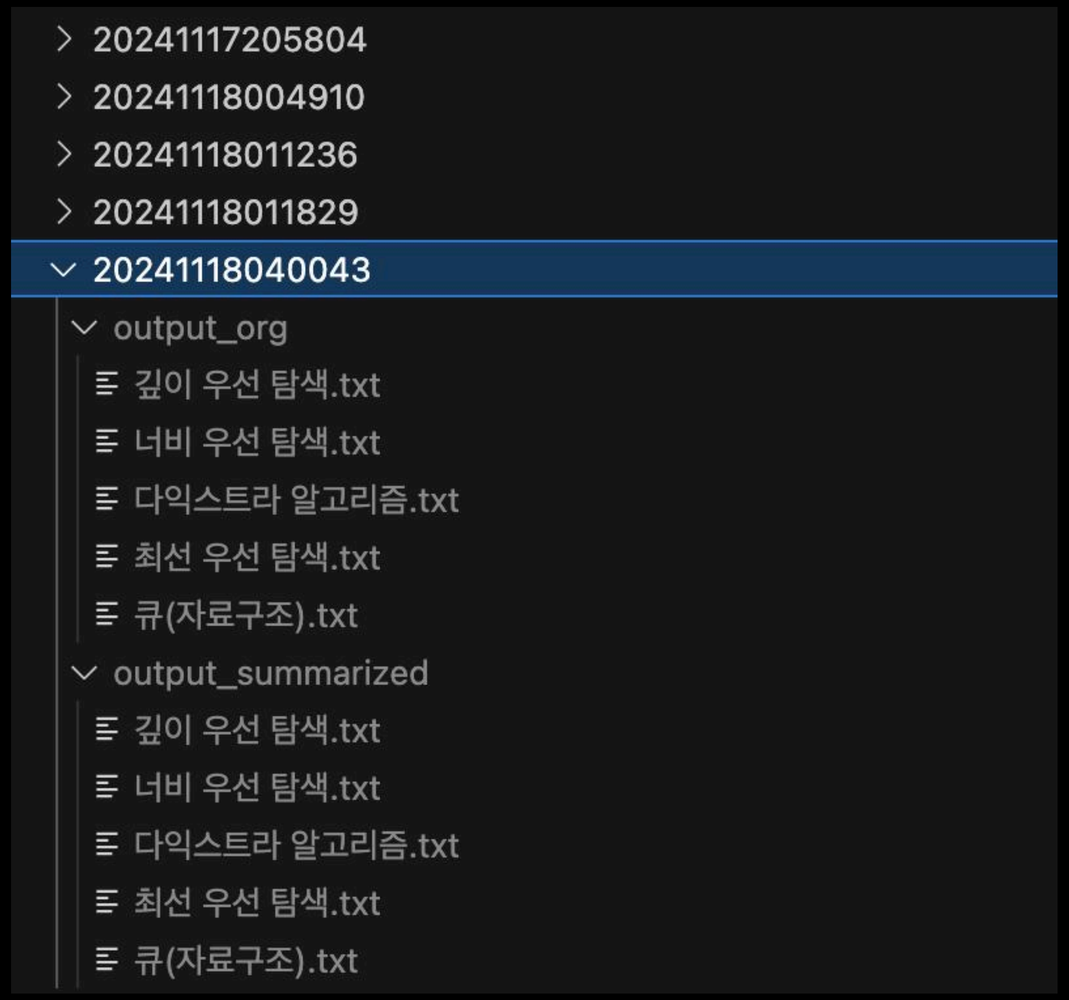
PARA에 대해서 전반적으로 알고 싶었지만...

이정훈과 동아리에 대한 정보는 수집할 수 없었다..

셀레니움을 사용하여 크롤링



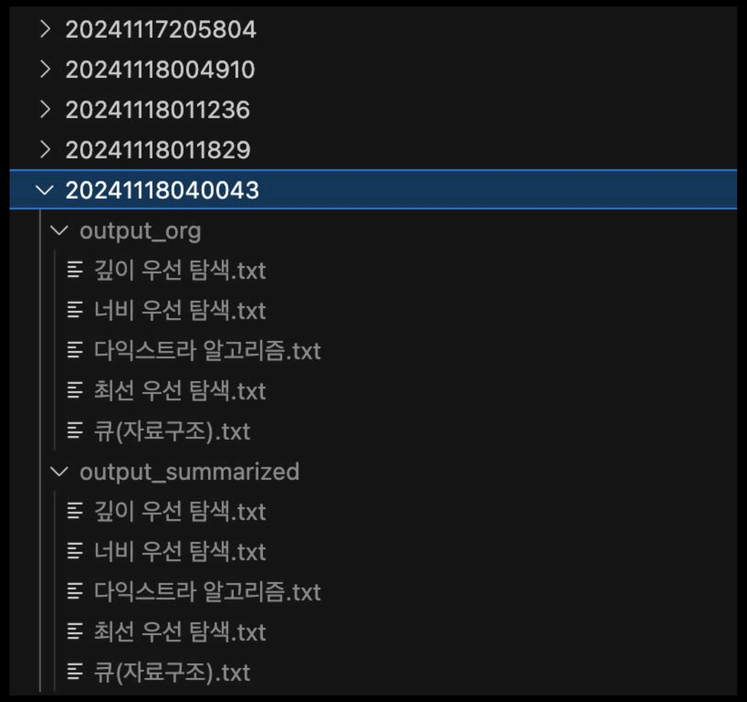
GPT를 사용해 RAG에 맞게 가공

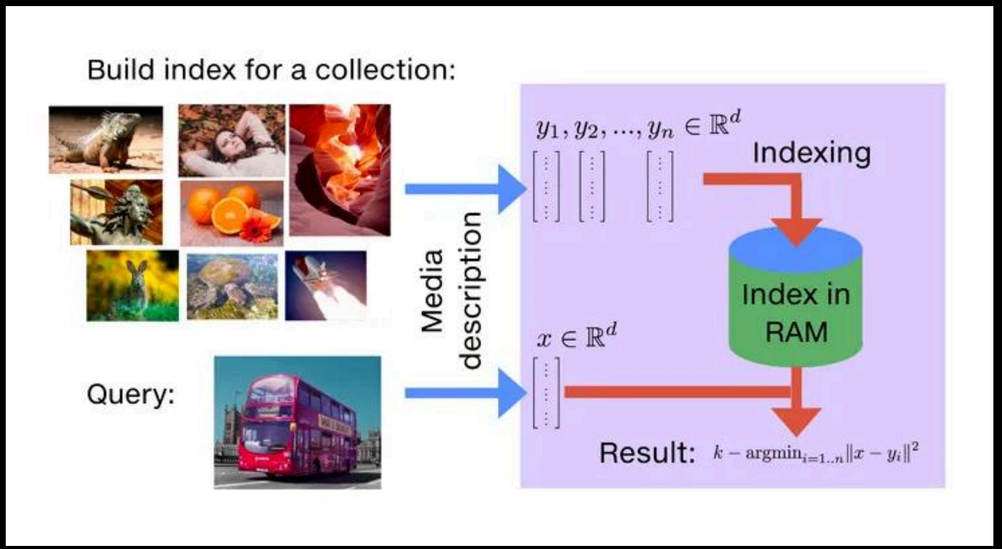

20241117205804
20241118004910
20241118011236
20241118011829
20241118040043
output_org
깊이 우선 탐색.txt
너비 우선 탐색.txt
다익스트라 알고리즘.txt
최선 우선 탐색.txt
큐(자료구조).txt
output_summarized
깊이 우선 탐색.txt
너비 우선 탐색.txt
다익스트라 알고리즘.txt
최선 우선 탐색.txt
큐(자료구조).txt

크롤링 & 가공 결과물!

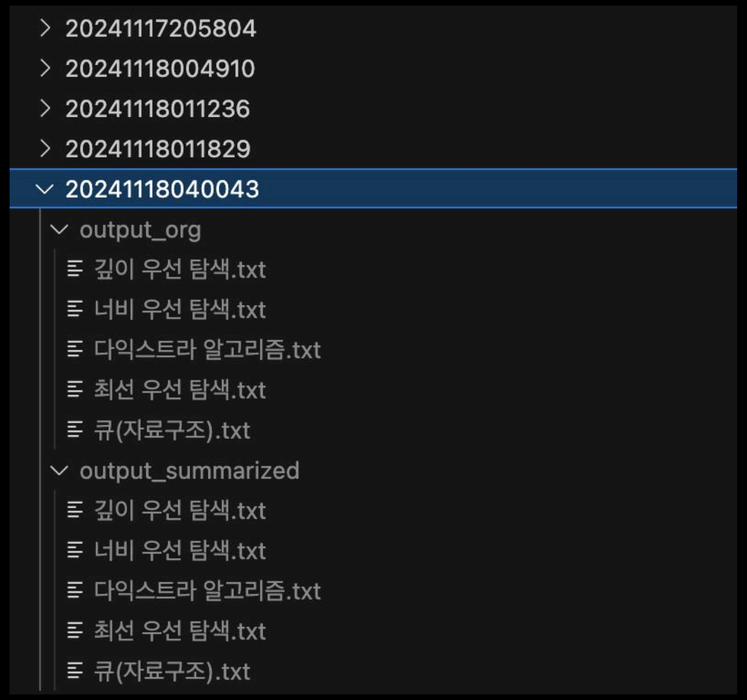
langchain😘

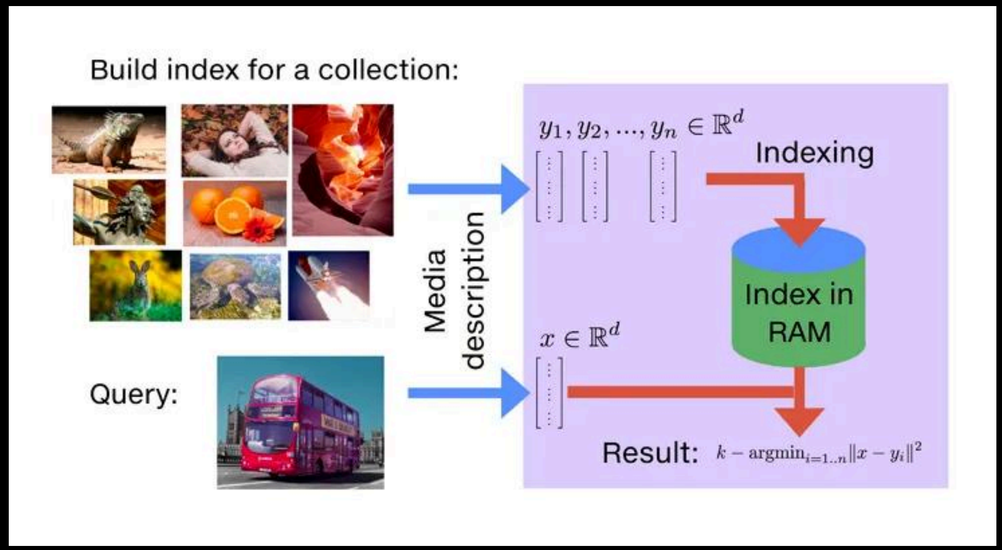
추출한 결과물을 chunking, embedding 후에





FAISS 인덱스에 추가!





RAG Pipeline 구축 완료!

실행 영상 보시겠습니다~

https://youtu.be/M9kzcWVkmiE