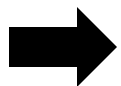


검색엔진 프로젝트

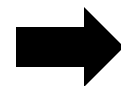
202246116 이재현

목차

구축한 검색엔진
아키텍처 설명

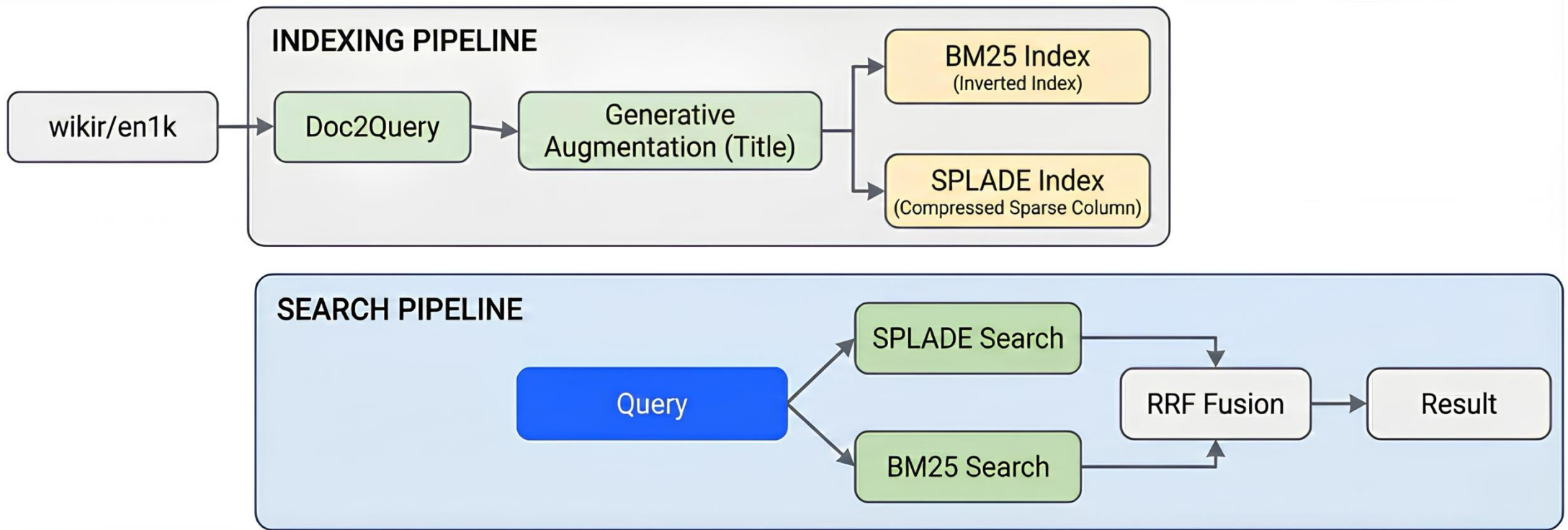


문제 및 해결 방법



결론

전체 아키텍처



BM25

Problem

사용자 쿼리와 문서 단어가
다르면 검색이 되지 않음

Query: Machine learning

Document: artificial intelligence

평가 결과	
MAP:	0.1925
nDCG:	0.4593
P@10:	0.2216
Recall@100:	0.4299
Recall@1000:	0.5691
Recall@2000:	0.6003
Recall@5000:	0.6350

Solution

Doc2Query를 이용하여 문서당
10개의 예상 질문을 생성

Model: doc2query-t5-base-msmarco

평가 결과	
MAP:	0.2202
nDCG:	0.5059
P@10:	0.2504
Recall@100:	0.4391
Recall@1000:	0.5786
Recall@2000:	0.6096
Recall@5000:	0.6454

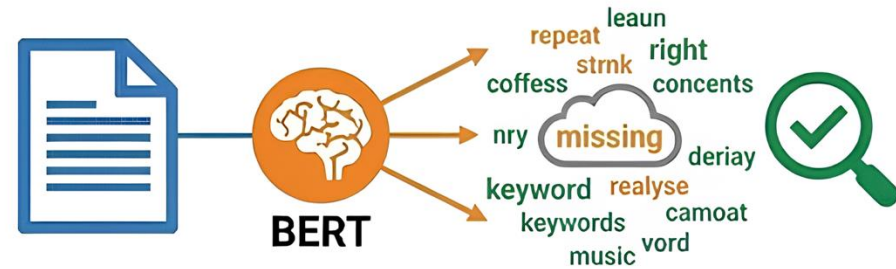
SPLADE (Sparse Lexical and Expansion Model)

기존 BM25 (단순 키워드 매칭)



문서에 없는 단어는 검색 불가

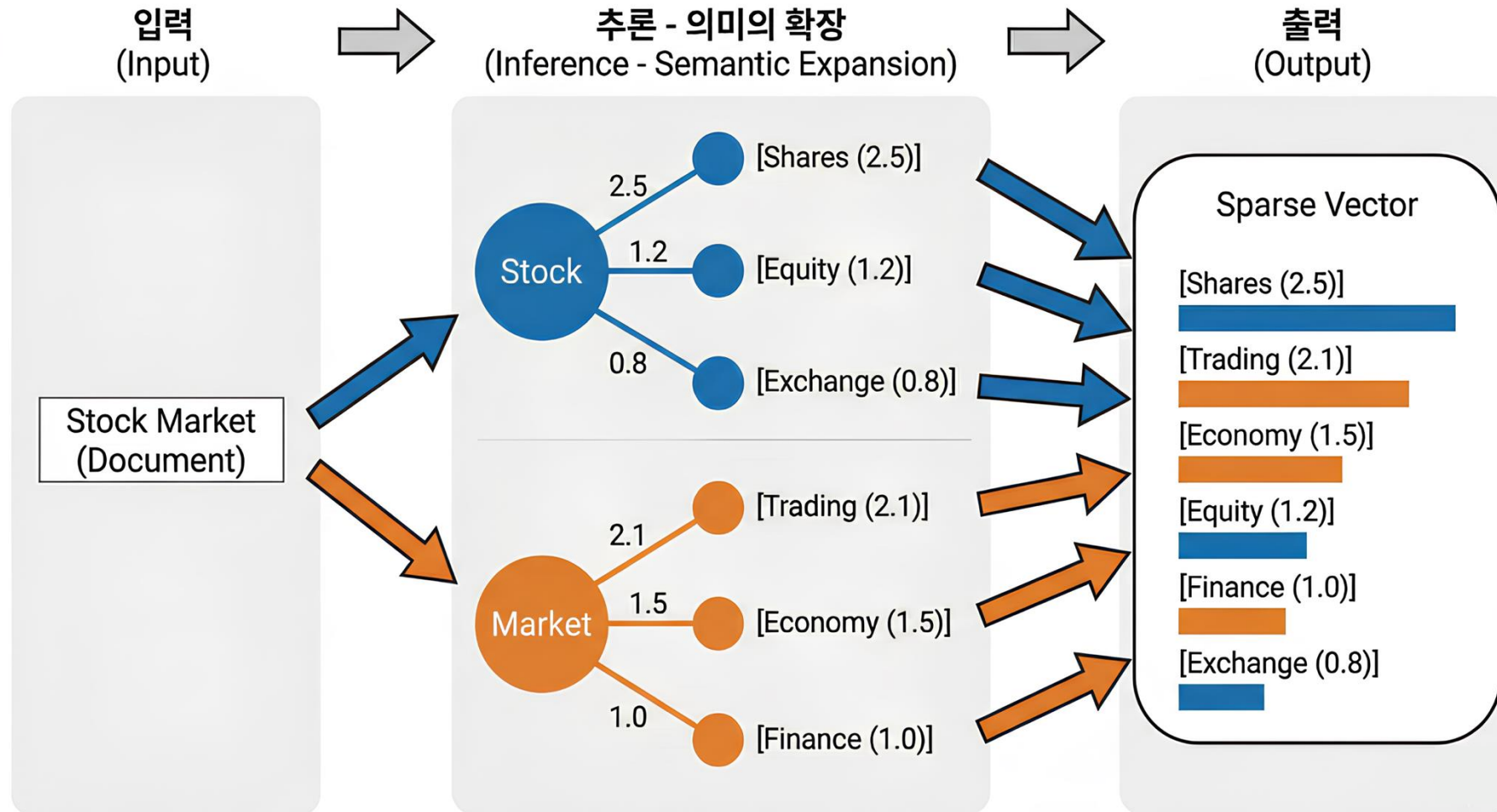
SPLADE (문맥 기반 확장)



문맥을 이해하여 연관 단어까지 검색

BM25의 속도와 BERT의 문맥 이해를
결합한 하이브리드 검색 모델

SPLADE (Sparse Lexical and Expansion Model)



CSC(Compressed Sparse Column)

일반적인 행렬로 저장하는 경우 메모리에 적재 불가능

370,000(문서) ×
30,522(단어) ×
4byte

= 42GB



0이 아닌 유의미한 값과 인덱스만 저장

+ 양자화
Float32 -> Int16

= 약 190MB

BM25 + SPLADE

BM25

TF-IDF 기반의 확률적 점수



SPLADE

벡터 내적 기반의 유사도 점수



점수의 분포가 달라 가중 평균은
특정 모델에게 편향될 위험이 존재

RRF(Reciprocal Rank Fusion)

$$\text{RRF_score}(d) = \sum \left[\frac{1}{k + \text{rank}_i(d)} \right]$$

RRF는 점수가 아닌, 순위를 기반으로 최종 점수를 계산하는 방법.

Scale 차이를 무시하며, 두 모델에서 모두 상위인 문서에 가중치를 부여

SPLADE 적용 전

MAP:	0.2202
nDCG:	0.5059
P@10:	0.2504
Recall@100:	0.4391
Recall@1000:	0.5786

SPLADE 적용 후

MAP:	0.2327
nDCG:	0.5286
P@10:	0.2588
Recall@100:	0.4649
Recall@1000:	0.6296

Generative Augmentation (Title)

사용중인 wikir/en1k 데이터셋에는
제목이 저장되어 있지 않음

쿼리와 제목간의 유사도를 반영하지 못함
UX 저하



michau/t5-base-en-generate-headline 모델을 사용해 문서
의 내용을 함축하는 **제목을 생성하여 검색 피처에 추가**

최종 성능 분석

평가지표	BM25 (Baseline)	Hybrid Search (Final)	성능 향상 (증감률)
MAP	0.1925	0.2356	+ 22.4%
nDCG	0.4593	0.5336	+ 16.2%
P@10	0.2216	0.2636	+ 19.0%
Recall@1000	0.5691	0.6274	+ 10.2%

실사용 환경 속도 테스트 (100개 쿼리)

평균 응답 속도: 0.116초

결론 및 향후 과제

전통적 검색(**BM25**)과 최신 뉴럴 검색(**SPLADE**)을 결합

Generative Augmentation (Title Generation + Doc2Query)

Baseline(BM25) 대비 **MAP 22.4%, nDCG 16.2%** 향상

Cross-Encoder를 이용한 Reranking을 통해 P@10 nDCG 성능 향상
구축한 모델이 다른 대용량 데이터셋에서도 동작하는지 범용성 테스트

감사합니다