문제 정의: 기사 작성시 '관련 기사'를 직접 찾아서 등록하는 것이 현실적으로 어렵다

사용자: CMS를 통해 기사 초안을 작성하는 기자

해결책: 기자가 작성한 기사 초안에서 유사한 맥락의 기사를 추천하여 손쉽게 '관련 기사'를 등록할 수 있도록 한다.

(궁금증: 기사 초안과 유사한 맥락보다 그 사건을 이해하기 위한 지식을 전하는 기사를 관련 기사로 기입하는 경우도 있는가?)

구체적인 상황

1. 채팅이 아니므로 관련 기사 검색 속도가 매우 빠를 필요는 없다

2. 오히려 검색 속도 보다는 정확도를 높이는 방식의 검색이 중요해 보인다

3. 세부적인 내용에 대한 응답을 해야 하는 것이 아니기 때문에 RAG를 사용할 필요가 없으며, Vector DB와 Embedding 기반 유사도 조회만 이용하면 된다

4. 세부적인 내용에 대한 응답이 필요한 것이 아니므로, 기존의 기사, 그리고 query로 사용될 기사 초안의 모든 텍스트에 대한 embedding이 필요하지는 않아 보인다. (기사의 분량 때문에 기사를 쪼개야 하는 작업이 없어진다)

5. 메타데이터(날짜를 제한한다거나 최신 순으로 정렬한다거나 특정 기자의 기사에서 조회하는 등)를 원활히 활용할 수 있는 것이 중요해 보인다

<서비스 아키텍쳐>

1. 텍스트 임베딩 및 유사도 검색

```
VectorDB = {
        Content = 기사 본문
        Metadata = [
                기사 ID, 기사 링크, 작성 기자, 작성 날짜 ...
        ]
}
```

2. 백엔드 서버: FastAPI

<구체적인 서비스 작동 원리>

1. 기자가 CMS에서 기사의 초안을 완성한다

2. 기사 초안에 대한 '관련 기사 추천' 기능을 사용한다(필요시 날짜, 기자와 같은 필터링을 설정)

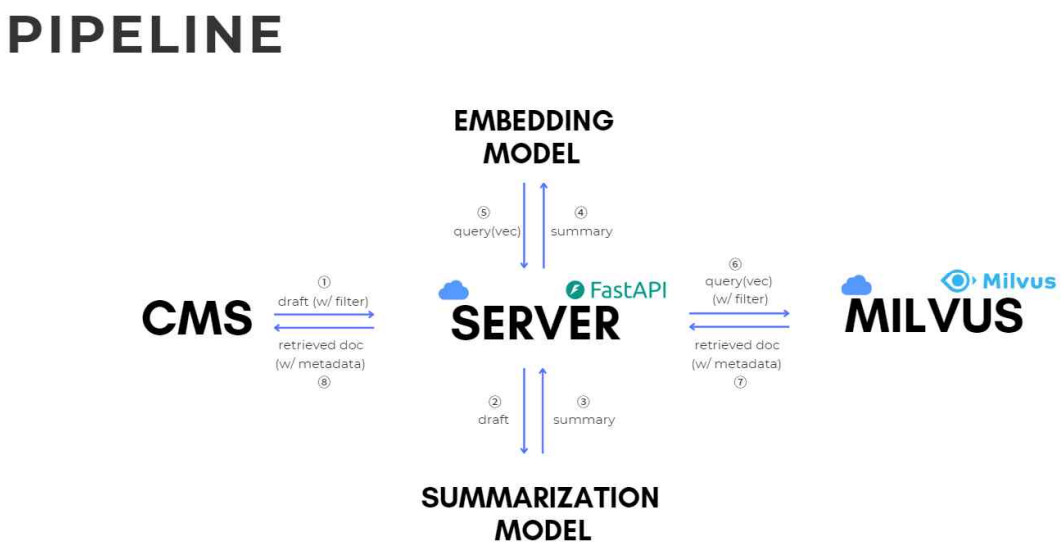
3. 서버에서 해당 기사 초안을 받아 AI 모델을 이용해 요약하고 요약문을 임베딩하여 벡터를 만든다 (필터링용 데이터가 있다면 같이 전송)

4. 서버가 VectorDB에 해당 벡터를 Query로 사용하여 관련 기사를 검색한다(필터링용 데이터가 있다면 메타 데이터 필터링을 수행)

5. 5개의 관련 기사를 검색하여 해당 기사의 요약문, 기자, 작성일, 링크 등을 CMS에 반환하고 이를 토대로 기자가 '관련 기사'를 선정한다

(기자, 작성일, 링크 등을 한번에 기입할 수 있으며, 기사 링크를 들어가서 일일이 내용을 확인하지 않더라도 요약문을 간단히 살펴보고 원하는 맥락의 기사인지를 판별할 수 있다)

<파이프라인>



<Vector DB 비교>

1. Chroma

- 메타데이터 필터링을 제공한다
- 간편하게 사용할 수 있고 유연한 API를 제공한다
- 대규모 데이터셋을 처리할 때 성능이 저하될 수 있다

2. FAISS

- 매우 빠른 벡터 검색 성능을 제공한다
- 기본적으로 벡터와 메타데이터를 함께 관리하고 필터링하는 기능은 제공하지 않는다
- 단일 노드에서 실행되며 매우 큰 데이터셋을 처리할 때 확장성이 제한될 수 있다

3. Milvus

- 분산 아키텍쳐를 사용하여 대규모 데이터셋에서도 확장성과 성능을 보장한다
- 벡터와 함께 메타데이터를 저장하고 필터링할 수 있는 기능을 제공한다
- 클라우드, 온프레미스, Docker, Kubernetes 환경에서 쉽게 배포할 수 있다

View Table

	Pinecone	Weaviate	Milvus	Qdrant	Chroma	Elasticsearch	PGvector
오픈소스 여부 / Is open source	❌	✅	✅	✅	✅	❌	✅
자체 호스팅 / Self-host	❌	✅	✅	✅	✅	✅	✅
클라우드 관리 / Cloud management	✅	✅	✅	✅	❌	✅	(✓)
벡터 전용 / Purpose-built for Vectors	✅	✅	✅	✅	✅	❌	❌
개발자 경험 / Developer experience	👍👍👍	👍👍	👍👍	👍👍	👍👍	👍	👍
커뮤니티 / Community	Community page & events	8k☆ github, 4k slack	23k☆ github, 4k slack	13k☆ github, 3k discord	9k☆ github, 6k discord	23k slack	6k☆ github
초당 쿼리 수 / Queries per second (using text nytimes-256-angular)	150 *for p2, but more pods can be added	791	2406	326	?	700-100 *from various reports	141
지연 시간, ms / Latency, ms (Recall/Percentile 95 (millis), nytimes-256-angular)	1 *batched search, 0.99 recall, 200k SBERT	2	1	4	?	?	8
지원하는 인덱스 종류 / Supported index types	?	HNSW	Multiple (11 total)	HNSW	HNSW	HNSW	HNSW/IVFFlat
하이브리드 검색 / Hybrid Search (i.e. scalar filtering)	✅	✅	✅	✅	✅	✅	✅
디스크 인덱스 지원 / Disk index support	✅	✅	✅	✅	✅	❌	✅
역할-기반 접근 제어(RBAC) / Role-based access control	✅	❌	✅	❌	❌	✅	❌
동적 세그먼트 배치 vs. 정적 데이터 샤딩 / Dynamic segment placement vs. static data sharding	?	Static sharding	Dynamic segment placement	Static sharding	Dynamic segment placement	Static sharding	-

https://discuss.pytorch.kr/t/2023-picking-a-vector-database-a-comparison-and-guide-for-2023/2625

3. **커뮤니티의 힘**: Milvus는 가장 큰 커뮤니티를 자랑하며, 그 뒤를 Weviate와 Elasticsearch가 따릅니다. 강력한 커뮤니티는 종종 더 나은 지원, 개선, 버그 수정으로 이어집니다.

> 3. **Community Strength**: Milvus boasts the largest community presence, followed by Weviate and Elasticsearch. A strong community often translates to better support, enhancements, and bug fixes.

4. **확장성, 고급 기능 및 보안**: 많은 엔터프라이즈 애플리케이션에 필수적인 기능인 역할 기반 액세스 제어(RBAC)는 Pinecone, Milvus, Elasticsearch에서 찾아볼 수 있습니다. 확장 측면에서는 Milvus와 Chroma가 제공하는 동적 세그먼트 배치가 끊임없이 진화하는 데이터셋에 적합합니다. 다양한 인덱스 유형이 포함된 데이터베이스가 필요한 경우, 11가지 유형에 대한 Milvus의 지원은 타의 추종을 불허합니다. 하이브리드 검색은 전반적으로 잘 지원되지만, 디스크 인덱스 지원 측면에서는 Elasticsearch가 다소 부족합니다.

Milvus의 hybrid search를 사용하면 기사 제목에 대한 벡터값, 기사 내용에 대한 벡터값을 기반으로 동시에 검색하는데 사용하는 방식으로도 구현할 수 있다고 합니다. 또한 WeightedRanker를 사용하여 어떤 벡터 필드에 가중치를 부여할지도 손쉽게 적용할 수 있다고 합니다.

추가적으로 milvus에서 다른 기사 관련 데이터셋(9000개)를 기반으로 임베딩하여 테스트해보았는데 BM25, NER 등의 키워드 기반 검색을 추가하여 성능 개선을 시키면 좋지 않을까 합니다.

Cohere이나 Upstage의 최근 임베딩 모델들의 경우 document, query용 임베딩 모델이 구분되어 사용된다고 하는데 저희는 상업용이 아니니 성능을 비교해보고 이러한 모델들을 사용해보는 것도 좋지 않을까 합니다.