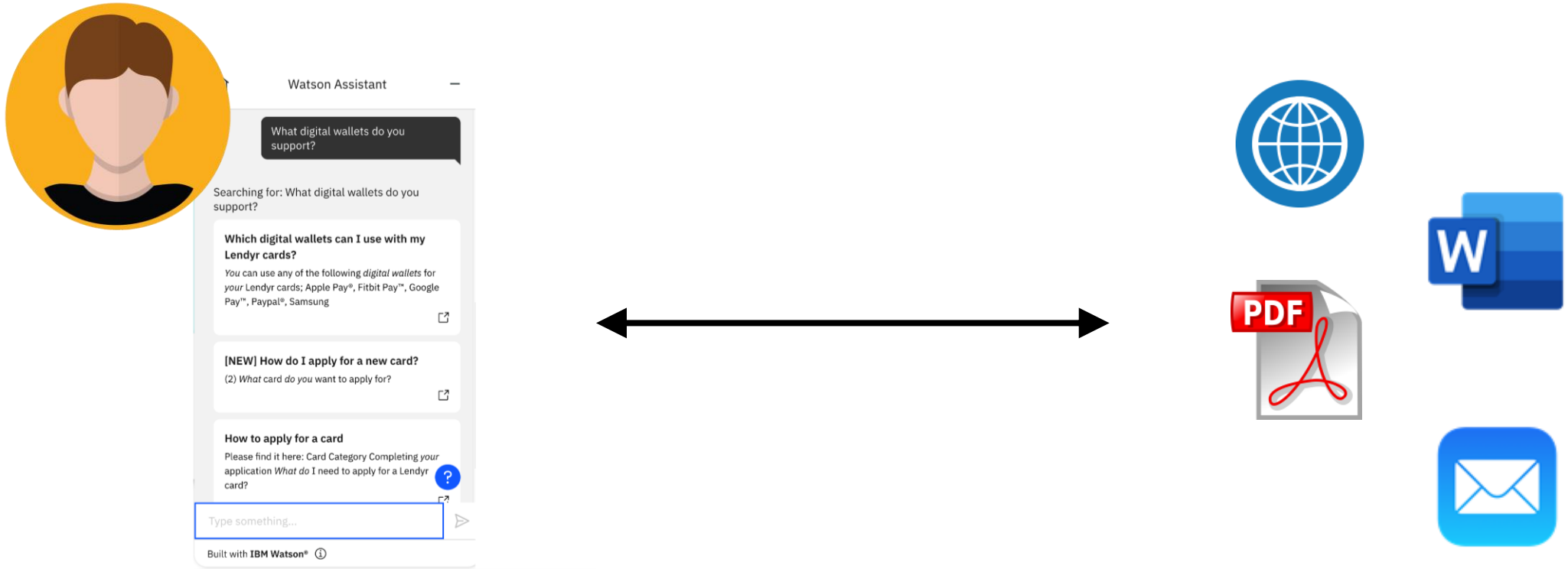
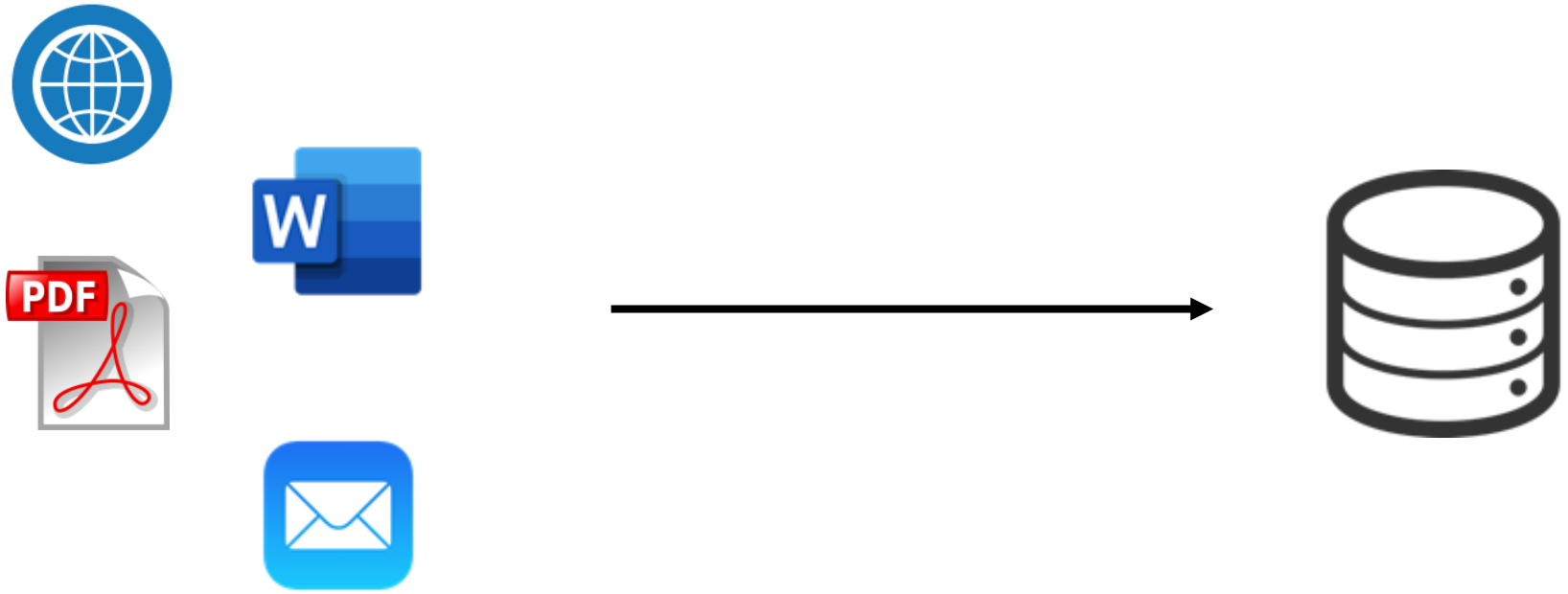
Retrieval Augmented Search - Overview

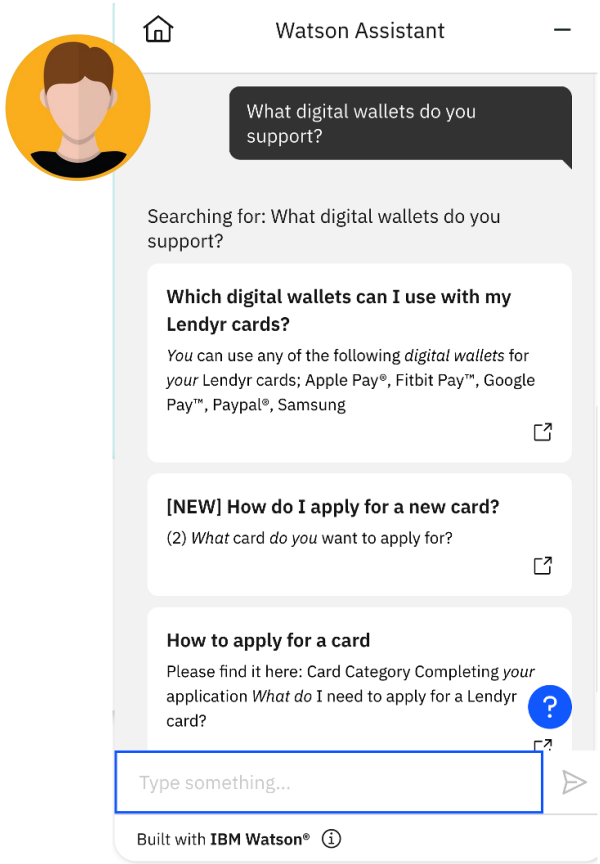
대화형 검색– 문서에 대한 Q&A



Phase 1
데이터 준비



Phase 2
데이터 쿼리

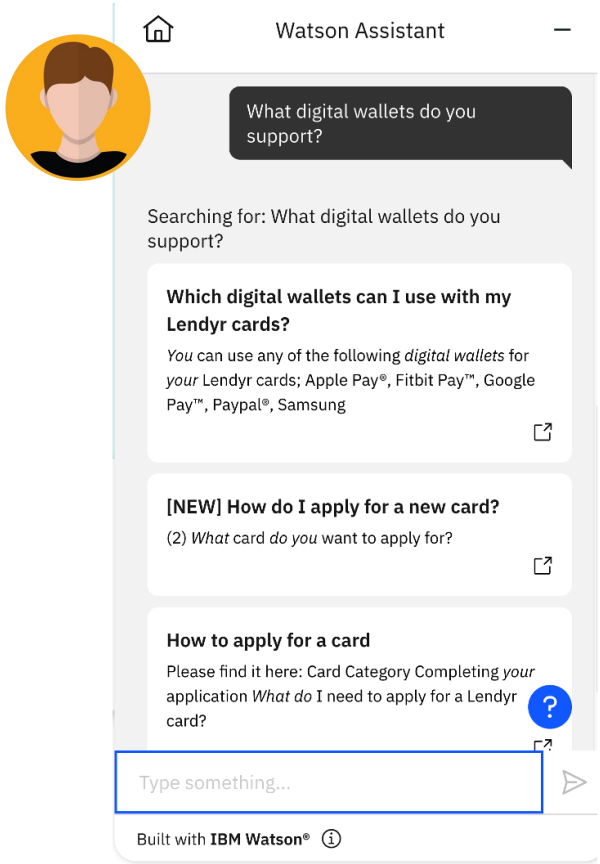


사용자의 질문

질문과 관련된
몇 가지 단락

Phase 2 (LLM을 기반으로 하는 새로운 단계)

데이터 쿼리



사용자의 질문

질문과 관련된
몇 가지 단락

다음 단락을 읽고 질문에 답하십시오.
[제1 단락]
[제2 단락]
[제3 단락]
질문: [사용자 질문]
대답:

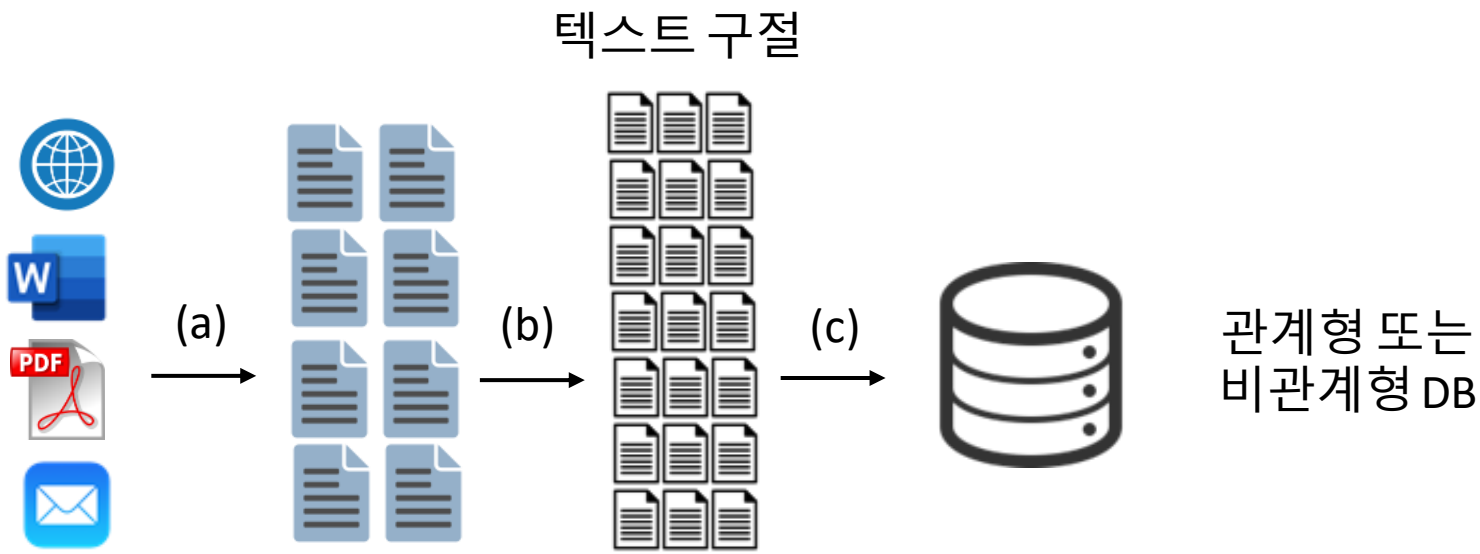
WatsonX.ai

Phase 1: "전통적인" 방식

Phase 1

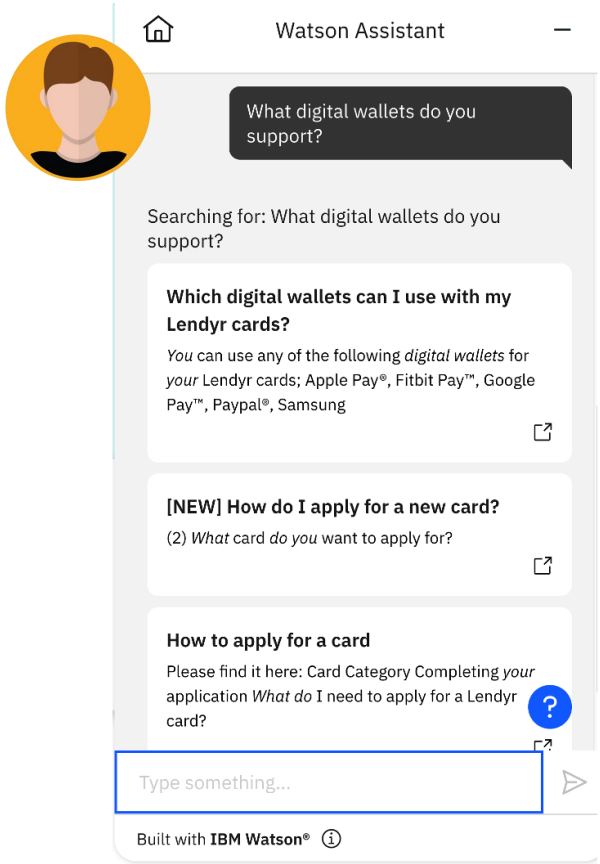
데이터 수집

(a) 원본 문서 파일의 문서화
(b) 문서를 청크(chunk)로 분할
(c) 분할된 청크를 DB로 저장



Phase 2: syntactic(구문) 검색

데이터 쿼리



카드 신청은 어떻게 하나요?

청크에 "신청" 또는 "카드"가 포함되도록 **모든** 행/문서를 검색합니다.

질문과 관련된 몇 가지 단락

다음 단락을 읽고 질문에 답하십시오.
[제1 단락]
[제2 단락]
[제3 단락]
질문: [사용자 질문]
대답:

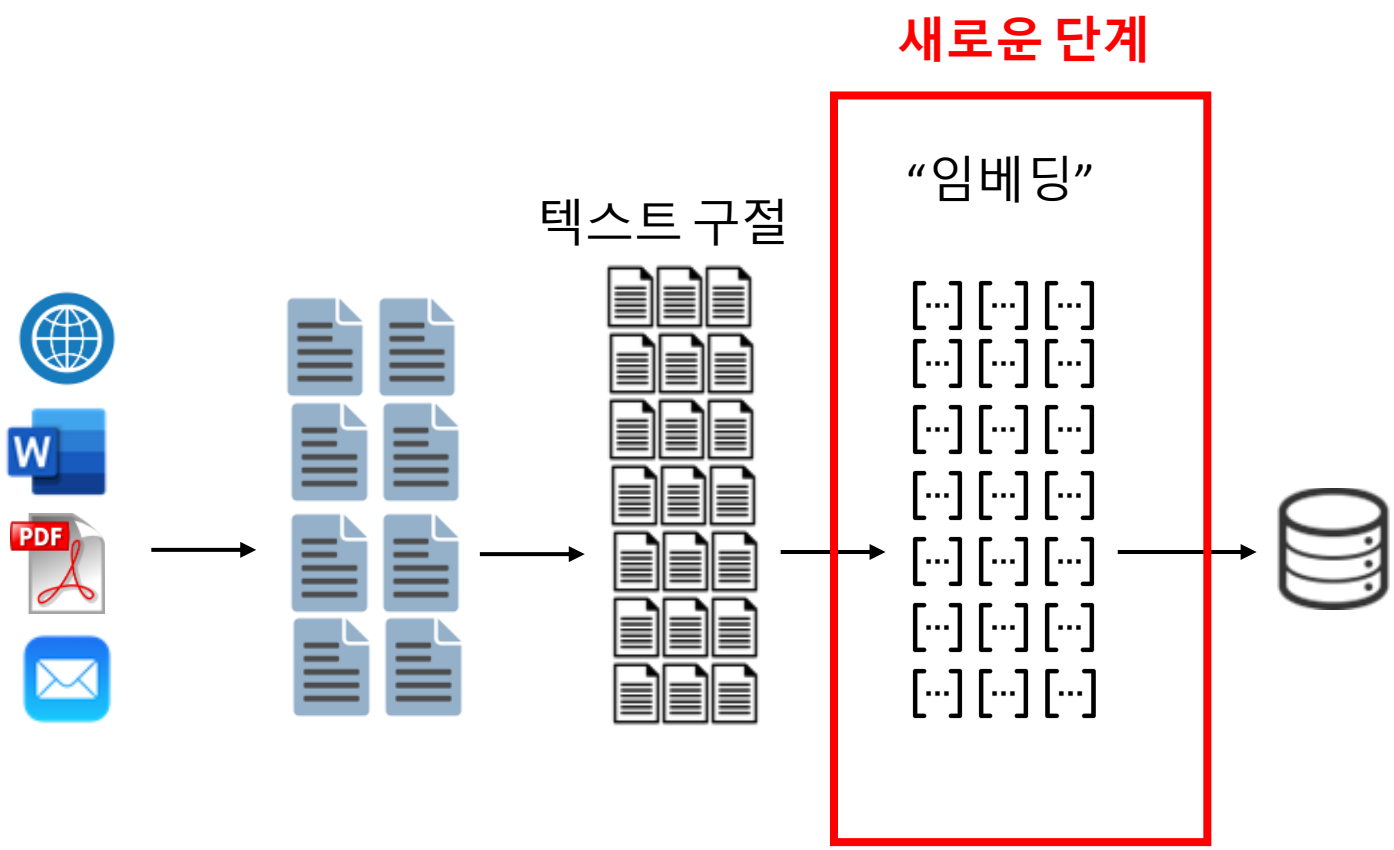
watsonx.ai

Phase 1: "임베딩" 방식

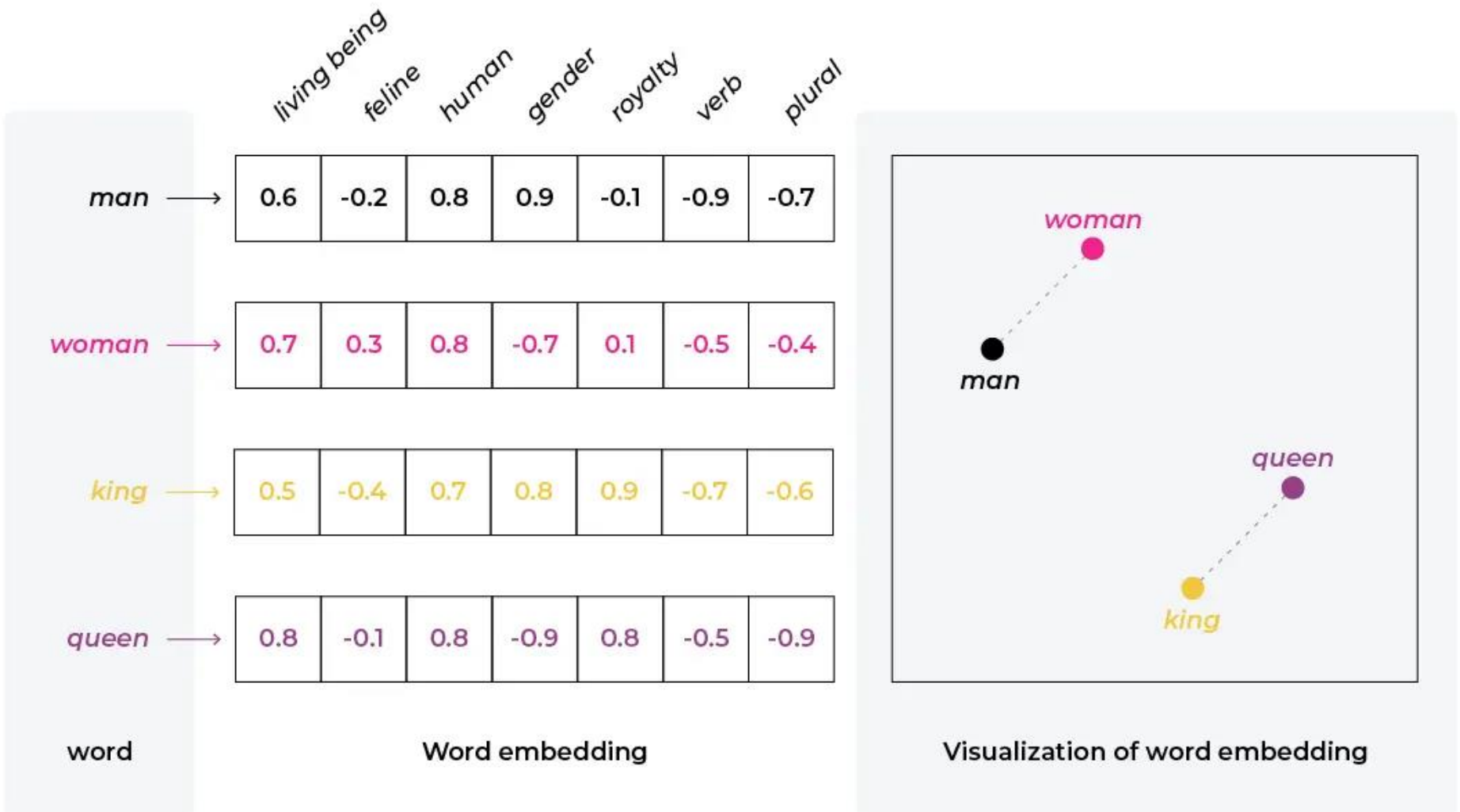
Phase 1

데이터 수집

(a) 원본 파일을 문서로
(b) 문서를 청크로 분할
(c) 청크를 임베딩으로 변환
(d) 임베딩을 백터 스토어로 전달

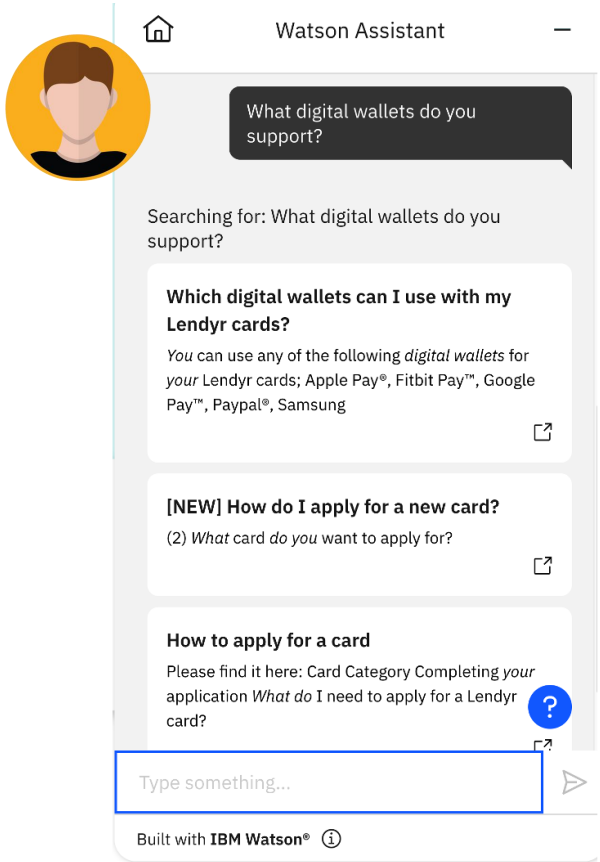


Phase 1: "임베딩" 방식 (예시)



Phase 2: 시멘틱 검색

데이터 쿼리



카드 신청은 어떻게 하나요?

사용자 질문의 임베딩에 가장 가까운 모든 임베딩을 검색합니다.

질문과 관련된 몇 가지 단락을 검색 결과 기준 선별

다음 단락을 읽고 질문에 답하십시오.
[제1 단락]
[제2 단락]
[제3 단락]
질문: [사용자 질문]
대답:

watsonx.ai

구문 기반 검색 vs. 시맨틱(의미 기반) 검색

시맨틱 검색이 정보를 검색하는 "더 나은" 방법인 이유

사용자는 자신의 방식으로 표현하는 반면 문서는 주로 "전문적인" 용어를 사용합니다.
예제:

유급 휴가
(IBM HR 문서)

기업 자산
(은행 윤리강령)

매출, 이익, 편익
재무 보고서

연차

회사의 노트북

수익

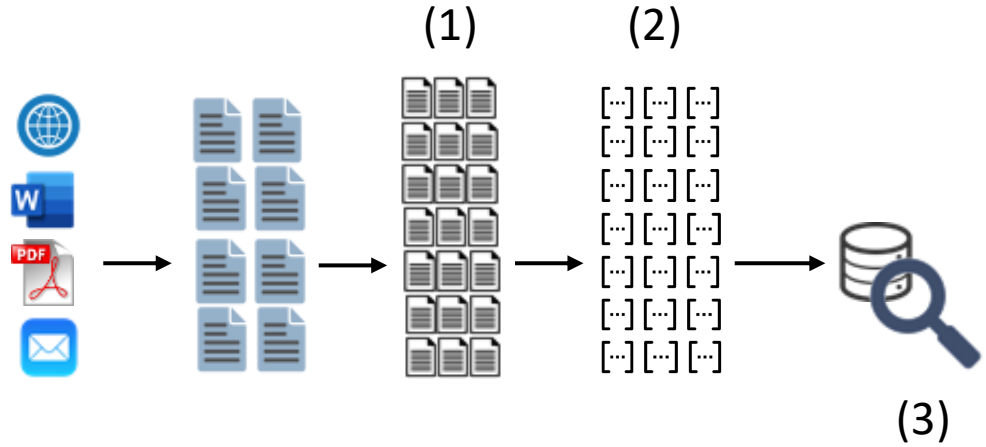
정확도를 향상시키는 방법은 무엇일까요?

예시문서 :**시몬스 은행 – 윤리 강령**

각 단계에서 config 구성요소를 최적화합니다.

1. 텍스트 청크의 길이
2. 임베딩 라이브러리 선택
3. 임베딩 간 거리 함수
4. 데이터베이스에서 검색되는 청크 수
5. 프롬프트
6. LLM 파라미터(temperature, topK, top 등)
7. LLM 선택
8. 기타

그러나 더 효율적인 방법이 있습니다



(5)

(4)
다음 단락을 읽고 질문에 답하십시오.
[제1 단락]
[제2 단락]
[제3 단락]
질문: [사용자 질문]
대답:

(6), (7)

watsonx.ai

Architecture overview

User

하는 일: 새 문서 업로드

1

Watson Discovery

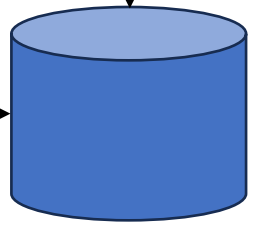
하는 일: 업로드한 파일의 모든 텍스트 추출

2

Cronjob (하루 한 번 실행)

하는 일: 마지막으로 업로드 된 콘텐츠를 가져와 임베딩으로 변환하여 벡터 스토어에 저장

3



벡터 스토어

하는 일: 문서의 모든 내용을 임베딩 형태로 저장

6

User

하는 일: 질문

4

Watson Assistant

하는 일: 대화 히스토리 생성 후 커스텀 익스텐션을 통해 웹 서비스로 보냄.

5

Web service

하는 일: 대화 히스토리를 기반으로 백터 스토어에 쿼리하고 가장 적절한 텍스트 스니펫을 찾아낸 후 watsonx.ai 에서 호스팅되는 LLM을 사용하여 사용자에게 답변을 만들어 냄.

7

watsonx.ai

하는 일: 사용자 친화적인 답변을 만들어냄

Retrieval Augment Generation - Applications

내용이 긴 문서 요약

Problem:
대규모 corpus, 잠재적인 정보 손실 및 비용이 많이 드는 프로세스

Solution:
질문시 RAG 활용 후 결과 수집 후 요약

API 검색 에이전트

Problem:
LLM이 사용할 정보를 전달하기 위해 YAML 파일의 대규모 corpus에 쿼리 및 처리

Solution:
RAG를 이용한 Q&A를 위해 처리된 YAML file에 쿼리합니다

Wisper Bot

Problem:
대화 기록에서 자동 RAG 검색 어려움

Solution:
사용자의 채팅 기록을 실시간으로 처리, RAG를 활용하여 다음 단계를 위한 관련 정보를 얻습니다