
- 출처: LangChain 공식 문서 또는 해당 교재명
- 원본 URL: https://smith.langchain.com/hub/teddynote/summary-stuff-documents

---

4. `LongContextReorder_2`

4) `LongContextReorder 알고리즘의 한계`

- 근본적인 문제: 알고리즘 자체의 한계 = **`LongContextReorder`** = **`단순 순서 변경 알고리즘`**
  - **`검색기의 순서를 맹신`**: 벡터 검색 결과가 완벽하다고 가정
  - **`실제 관련성 무시`**: 문서 내용과 쿼리의 실제 관련도를 재계산하지 않음

```
# 🔍 LongContextReorder의 실제 알고리즘 (웹 검색 결과 분석)
def transform_documents(documents):
    """
    Lost in the middle 해결을 위한 재정렬:
    - 가장 관련성 높은 문서 → 처음과 끝
    - 중간 관련성 문서 → 가운데

    하지만 문제: 원래 검색 순서를 그대로 신뢰!
    """
    reordered = []
    n = len(documents)

    # 홀수 인덱스 → 앞쪽 배치
    # 짝수 인덱스 → 뒤쪽 배치
    for i in range(n):
        if i % 2 == 0:
            reordered.insert(0, documents[i])  # 앞쪽
        else:
            reordered.append(documents[i])     # 뒤쪽

    return reordered
```

  - **`FakeEmbeddings의 랜덤성`**: 의미 없는 순서를 재정렬해봤자 의미 없음

```
# API 키를 환경변수로 관리하기 위한 설정 파일
from dotenv import load_dotenv

# API 키 정보 로드
load_dotenv()                               # True
```

```
from langsmith import Client
from langsmith import traceable

import os

# LangSmith 환경 변수 확인

print("\n--- LangSmith 환경 변수 확인 ---")
langchain_tracing_v2 = os.getenv('LANGCHAIN_TRACING_V2')
langchain_project = os.getenv('LANGCHAIN_PROJECT')
langchain_api_key_status = "설정됨" if os.getenv('LANGCHAIN_API_KEY') else "설정되지 않음" # API 키 값은 직접 출력하지 않음

if langchain_tracing_v2 == "true" and os.getenv('LANGCHAIN_API_KEY') and langchain_project:
    print(f"✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='{langchain_tracing_v2}')")
    print(f"✅ LangSmith 프로젝트: '{langchain_project}'")
    print(f"✅ LangSmith API Key: {langchain_api_key_status}")
    print("  -> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.")
else:
    print("❌ LangSmith 추적이 완전히 활성화되지 않았습니다. 다음을 확인하세요:")
    if langchain_tracing_v2 != "true":
        print(f"  - LANGCHAIN_TRACING_V2가 'true'로 설정되어 있지 않습니다 (현재: '{langchain_tracing_v2}').")
```

```
    if not os.getenv('LANGCHAIN_API_KEY'):
        print("  - LANGCHAIN_API_KEY가 설정되어 있지 않습니다.")
    if not langchain_project:
        print("  - LANGCHAIN_PROJECT가 설정되어 있지 않습니다.")
```

- 셀 출력

```
--- LangSmith 환경 변수 확인 ---
✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='true')
✅ LangSmith 프로젝트: 'LangChain-prantice'
✅ LangSmith API Key: 설정됨
-> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.
```

---

5) 의미 기반 재정렬 해보기

```
import numpy as np
from typing import List, Tuple
from langchain_core.documents import Document
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_community.vectorstores import Chroma
from langchain_community.document_transformers import LongContextReorder
import warnings
warnings.filterwarnings("ignore")

class IntelligentLongContextReorder:
    """Jay 전용 지능형 긴 문맥 재정렬기"""

    def __init__(self, embeddings):
        self.embeddings = embeddings

    def calculate_semantic_relevance(self, query: str, docs: List[Document]) -> List[Tuple[float, int, Document]]:
        """쿼리와 각 문서간의 실제 의미적 관련성 계산"""

        print(f"🔍 '{query}'에 대한 의미적 관련성 분석...")

        query_embedding = self.embeddings.embed_query(query)
        scored_docs = []

        for i, doc in enumerate(docs):
            doc_embedding = self.embeddings.embed_documents([doc.page_content])[0]

            # 코사인 유사도 계산
            similarity = np.dot(query_embedding, doc_embedding) / (
                np.linalg.norm(query_embedding) * np.linalg.norm(doc_embedding)
            )

            scored_docs.append((similarity, i, doc))

            # ChatGPT 관련 키워드 보너스 점수
            chatgpt_keywords = ['ChatGPT', '챗GPT', '챗지피티', 'OpenAI', '대화', '인공지능', 'AI']
            bonus = sum(1 for keyword in chatgpt_keywords if keyword in doc.page_content) * 0.1

            final_score = similarity + bonus
            scored_docs[-1] = (final_score, i, doc)

            print(f"  📄 [{i}] 점수: {final_score:.4f} (기본: {similarity:.4f}, 보너스: {bonus:.1f})")
            print(f"      내용: {doc.page_content[:50]}...")

        return scored_docs

    def intelligent_reorder(self, query: str, docs: List[Document]) -> List[Document]:
        """지능형 재정렬: 실제 관련성 + Lost-in-middle 최적화"""

        # 1단계: 실제 관련성 점수 계산
        scored_docs = self.calculate_semantic_relevance(query, docs)

        # 2단계: 관련성 순으로 정렬 (높은 점수부터)
        scored_docs.sort(key=lambda x: x[0], reverse=True)

        print(f"\n📊 관련성 순위:")
        for i, (score, original_idx, doc) in enumerate(scored_docs):
            print(f"  {i+1}위: {score:.4f} | {doc.page_content[:40]}...")

        # 3단계: Lost-in-middle 최적화 재정렬
        n = len(scored_docs)
        reordered = []

        # 높은 관련성 → 앞뒤, 낮은 관련성 → 중간
```

```
        high_relevance = scored_docs[:n//2]                    # 상위 50%
        low_relevance = scored_docs[n//2:]                     # 하위 50%

        # 가장 관련성 높은 문서들을 앞뒤에 배치
        for i, (score, idx, doc) in enumerate(high_relevance):
            if i % 2 == 0:
                reordered.insert(0, doc)                       # 앞쪽
            else:
                reordered.append(doc)                          # 뒤쪽

        # 관련성 낮은 문서들을 중간에 삽입
        mid_point = len(reordered) // 2
        for score, idx, doc in low_relevance:
            reordered.insert(mid_point, doc)
            mid_point += 1

        print(f"\n✅ 지능형 재정렬 완료!")
        print("📍 최종 배치 전략:")
        print("  - 1-3위: 앞쪽 (즉시 인식)")
        print("  - 4-7위: 중간 (보조 정보)")
        print("  - 8-10위: 뒤쪽 (마지막 체크)")
        print("\n", "="*80, "\n")

        return reordered

    def compare_methods(self, query: str, docs: List[Document]):
        """기본 LongContextReorder vs 지능형 재정렬 비교"""

        print("="*80)
        print("🔬 재정렬 방법 비교 실험")
        print("="*80)

        # 원본 순서
        print("\n🔍 1. 원본 검색 순서:")
        for i, doc in enumerate(docs):
            print(f"  [{i}] {doc.page_content}")

        # 기본 LongContextReorder
        print(f"\n🔄 2. 기본 LongContextReorder 결과:")
        basic_reorder = LongContextReorder()
        basic_result = basic_reorder.transform_documents(docs)
        for i, doc in enumerate(basic_result):
            print(f"  [{i}] {doc.page_content}")

        # 지능형 재정렬
        print(f"\n🧠 3. 지능형 재정렬 결과:")
        intelligent_result = self.intelligent_reorder(query, docs)
        for i, doc in enumerate(intelligent_result):
            print(f"  [{i}] {doc.page_content}")

        return basic_result, intelligent_result
# 3.1s 소요
```

```
# 🎬 테스트 함수
def ultimate_long_context_test():
    """궁극의 LongContextReorder 테스트"""

    print("🚀 Jay를 위한 완벽한 LongContextReorder 솔루션")
    print("="*80)

    # 고성능 임베딩 사용 (추천)
    embeddings = HuggingFaceEmbeddings(
        model_name="sentence-transformers/all-mpnet-base-v2",              # 768차원
        model_kwargs={'device': 'cpu'},
        encode_kwargs={'normalize_embeddings': True}
    )

    # 텍스트 데이터
    texts = [
        "이건 그냥 내가 아무렇게나 적어본 글입니다.",                                    # 관련성 매우 낮음
        "사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.",      # 관련성 높음
        "아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다.",                    # 관련성 낮음
        "챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다.",                 # 관련성 높음
        "챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 데이터를 학습했습니다.",  # 관련성 매우 높음
        "애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다.",             # 관련성 낮음
        "ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.",  # 관련성 높음
        "비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.",         # 관련성 낮음
        "ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.",            # 관련성 높음
        "FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입니다.",             # 관련성 낮음
    ]

    # 검색기 생성
    retriever = Chroma.from_texts(texts, embedding=embeddings).as_retriever(
        search_kwargs={"k": 10}
```

```
    )

    query = "ChatGPT에 대해 무엇을 말해줄 수 있나요?"

    docs = retriever.invoke(query)

    # 지능형 재정렬기 생성 및 테스트
    intelligent_reorder = IntelligentLongContextReorder(embeddings)
    basic_result, intelligent_result = intelligent_reorder.compare_methods(query, docs)

    # 성능 평가
    print(f"\n📈 성능 평가:")

    # ChatGPT 관련 문서 개수 세기
    def count_chatgpt_docs(docs, positions):
        count = 0
        for pos in positions:
            if pos < len(docs):
                content = docs[pos].page_content
                if any(keyword in content for keyword in ['ChatGPT', '챗GPT', '챗지피티']):
                    count += 1
        return count

    # 앞쪽 3개, 뒤쪽 3개 위치에서 관련 문서 비율
    front_back_positions = [0, 1, 2, 7, 8, 9]

    basic_quality = count_chatgpt_docs(basic_result, front_back_positions)
    intelligent_quality = count_chatgpt_docs(intelligent_result, front_back_positions)

    print(f"  🔄 기본 LongContextReorder: 앞뒤 위치에 ChatGPT 관련 문서 {basic_quality}개")
    print(f"  🧠 지능형 재정렬: 앞뒤 위치에 ChatGPT 관련 문서 {intelligent_quality}개")

    if intelligent_quality > basic_quality:
        print("  ✅ 지능형 재정렬이 더 우수합니다!")
    else:
        print("  ⚠️ 기본 방식과 비슷하거나 더 개선이 필요합니다.")
```

```
# 실행
ultimate_long_context_test()
```

- 셀 출력 (11.1s)

```
🚀 Jay를 위한 완벽한 LongContextReorder 솔루션
================================================================================
```

```
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling p
To disable this warning, you can either:
  - Avoid using `tokenizers` before the fork if possible
  - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
```

```
================================================================================
🔬 재정렬 방법 비교 실험
================================================================================

🔍 1. 원본 검색 순서:
  [0] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
  [1] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
  [2] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
  [3] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.
  [4] 아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다.
  [5] 애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다.
  [6] 챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 데이터를 학습했습니다.
  [7] 이건 그냥 내가 아무렇게나 적어본 글입니다.
  [8] FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입니다.
  [9] 챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다.

🔄 2. 기본 LongContextReorder 결과:
  [0] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
  [1] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.
  [2] 애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다.
  [3] 이건 그냥 내가 아무렇게나 적어본 글입니다.
  [4] 챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다.
  [5] FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입니다.
  [6] 챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 데이터를 학습했습니다.
  [7] 아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다.
```

```
  [8] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
  [9] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.

🧠 3. 지능형 재정렬 결과:
🔍 'ChatGPT에 대해 무엇을 말해줄 수 있나요?'에 대한 의미적 관련성 분석...
  📄 [0] 점수: 0.9724 (기본: 0.8724, 보너스: 0.1)
      내용: ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있...
  📄 [1] 점수: 1.1127 (기본: 0.8127, 보너스: 0.3)
      내용: 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다....
  📄 [2] 점수: 0.8960 (기본: 0.7960, 보너스: 0.1)
      내용: ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다....
  📄 [3] 점수: 0.6381 (기본: 0.6381, 보너스: 0.0)
      내용: 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다....
  📄 [4] 점수: 0.6047 (기본: 0.6047, 보너스: 0.0)
      내용: 아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다....
  📄 [5] 점수: 0.6033 (기본: 0.6033, 보너스: 0.0)
      내용: 애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다....
  📄 [6] 점수: 0.6979 (기본: 0.5979, 보너스: 0.1)
      내용: 챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 데이터를 학습했습니...
  📄 [7] 점수: 0.5719 (기본: 0.5719, 보너스: 0.0)
      내용: 이건 그냥 내가 아무렇게나 적어본 글입니다....
  📄 [8] 점수: 0.5033 (기본: 0.5033, 보너스: 0.0)
      내용: FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입니다....
  📄 [9] 점수: 0.6591 (기본: 0.3591, 보너스: 0.3)
      내용: 챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다....

📊 관련성 순위:
  1위: 1.1127  | 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답...
  2위: 0.9724  | ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데...
  3위: 0.8960  | ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있...
  4위: 0.6979  | 챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 ...
  5위: 0.6591  | 챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다...
  6위: 0.6381  | 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 ...
  7위: 0.6047  | 아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다....
  8위: 0.6033  | 애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다....
  9위: 0.5719  | 이건 그냥 내가 아무렇게나 적어본 글입니다....
  10위: 0.5033 | FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입...

✅ 지능형 재정렬 완료!
📍 최종 배치 전략:
  - 1-3위: 앞쪽 (즉시 인식)
  - 4-7위: 중간 (보조 정보)
  - 8-10위: 뒤쪽 (마지막 체크)

================================================================================

  [0] 챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다.
  [1] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
  [2] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.
  [3] 아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다.
  [4] 애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다.
  [5] 이건 그냥 내가 아무렇게나 적어본 글입니다.
  [6] FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입니다.
  [7] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
  [8] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
  [9] 챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 데이터를 학습했습니다.

📈 성능 평가:
  🔄 기본 LongContextReorder: 앞뒤 위치에 ChatGPT 관련 문서 3개
  🧠 지능형 재정렬: 앞뒤 위치에 ChatGPT 관련 문서 5개
  ✅ 지능형 재정렬이 더 우수합니다!
```

---

6) 키워드 기반 재정렬 해보기

```
class SimpleKeywordReorder:
    """키워드 기반 간단한 재정렬기"""

    def __init__(self, target_keywords):
        self.target_keywords = target_keywords
```

```
    def reorder_by_keywords(self, docs: List[Document]) -> List[Document]:
        """키워드 매칭 점수로 재정렬"""

        scored_docs = []

        for i, doc in enumerate(docs):
            # 키워드 매칭 점수 계산
            score = sum(1 for keyword in self.target_keywords
                        if keyword.lower() in doc.page_content.lower())
            scored_docs.append((score, i, doc))

        # 점수순 정렬
        scored_docs.sort(key=lambda x: x[0], reverse=True)

        # Lost-in-middle 재정렬
        reordered = []
        n = len(scored_docs)

        for i, (score, idx, doc) in enumerate(scored_docs):
            if i < n//3:  # 상위 1/3
                if i % 2 == 0:
                    reordered.insert(0, doc)                    # 앞쪽
                else:
                    reordered.append(doc)                       # 뒤쪽
            else:                                               # 하위 2/3
                mid = len(reordered) // 2
                reordered.insert(mid, doc)                      # 중간

        return reordered
```

```
# 사용법
keywords = ['ChatGPT', '챗GPT', '챗지피티', 'OpenAI', 'AI', '인공지능', '대화']
simple_reorder = SimpleKeywordReorder(keywords)
```

```
# 고성능 임베딩 사용 (추천)
embeddings = HuggingFaceEmbeddings(
    model_name="sentence-transformers/all-mpnet-base-v2",                   # 768차원
    model_kwargs={'device': 'cpu'},
    encode_kwargs={'normalize_embeddings': True}
    )

# 원본 텍스트
texts = [
    "이건 그냥 내가 아무렇게나 적어본 글입니다.",                                          # 관련성 매우 낮음
    "사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.",        # 관련성 높음
    "아이폰, 아이패드, 맥북 등은 애플이 출시한 대표적인 제품들입니다.",                      # 관련성 낮음
    "챗GPT는 OpenAI에 의해 개발되었으며, 지속적으로 개선되고 있습니다.",                   # 관련성 높음
    "챗지피티는 사용자의 질문을 이해하고 적절한 답변을 생성하기 위해 대량의 데이터를 학습했습니다.",  # 관련성 매우 높음
    "애플 워치와 에어팟 같은 웨어러블 기기도 애플의 인기 제품군에 속합니다.",               # 관련성 낮음
    "ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.",  # 관련성 높음
    "비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.",        # 관련성 낮음
    "ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.",            # 관련성 높음
    "FIFA 월드컵은 네 번째 해마다 열리며, 국제 축구에서 가장 큰 행사입니다.",              # 관련성 낮음
    ]

retriever = Chroma.from_texts(texts, embedding=embeddings).as_retriever(
        search_kwargs={"k": 10}
    )

query = "ChatGPT에 대해 무엇을 말해줄 수 있나요?"

docs = retriever.invoke(query)


# 문서 재정렬
reordered_docs = simple_reorder.reorder_by_keywords(docs)                   # 3.5s
```

- 최종 솔루션

```
def jay_perfect_reorder_solution():
    """Jay 전용 완벽한 재정렬 솔루션"""

    from langchain_community.document_transformers import LongContextReorder
    from langchain_huggingface import HuggingFaceEmbeddings
    from langchain_community.vectorstores import Chroma

    # 1. 고성능 임베딩 사용
    embeddings = HuggingFaceEmbeddings(
        model_name="sentence-transformers/all-mpnet-base-v2",
        model_kwargs={'device': 'cpu'},
        encode_kwargs={'normalize_embeddings': True}
    )
```

```
    # 2. 극명한 차이의 데이터 사용
    texts = [
        # ChatGPT 관련 (높은 관련성) - 4개
        "ChatGPT는 OpenAI에서 개발한 혁신적인 대화형 AI 모델입니다.",
        "사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.",
        "ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.",
        "OpenAI의 ChatGPT는 지속적인 학습을 통해 성능이 개선되고 있습니다.",

        # 중간 관련성 - 2개
        "인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.",
        "자연어 처리 기술의 발전으로 인간-컴퓨터 상호작용이 크게 향상되었습니다.",

        # 낮은 관련성 - 4개
        "애플은 아이폰, 맥북, 아이패드 등 혁신적인 전자제품을 생산하는 글로벌 기업입니다.",
        "비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.",
        "축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.",
        "요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.",
    ]

    # 3. 검색기 생성
    retriever = Chroma.from_texts(texts, embedding=embeddings).as_retriever(
        search_kwargs={"k": 10}
    )

    query = "ChatGPT에 대해 자세히 설명해주세요."

    docs = retriever.invoke(query)

    print("🔍 원본 검색 결과:")
    for i, doc in enumerate(docs):
        print(f"[{i}] {doc.page_content}")

    # 4. LongContextReorder 적용
    reordering = LongContextReorder()
    reordered_docs = reordering.transform_documents(docs)

    print(f"\n🔄 LongContextReorder 적용 후:")
    for i, doc in enumerate(reordered_docs):
        print(f"[{i}] {doc.page_content}")

    # 5. 품질 검증
    chatgpt_keywords = ['ChatGPT', 'OpenAI', '대화', 'AI']

    def calculate_quality(docs):
        front_back = [0, 1, 2, 7, 8, 9]                                    # 앞 3개, 뒤 3개
        quality = 0
        for pos in front_back:
            if pos < len(docs):
                content = docs[pos].page_content
                if any(keyword in content for keyword in chatgpt_keywords):
                    quality += 1
        return quality

    original_quality = calculate_quality(docs)
    reordered_quality = calculate_quality(reordered_docs)

    print(f"\n📊 품질 분석:")
    print(f"원본 순서: 앞뒤 위치 관련 문서 {original_quality}/6")
    print(f"재정렬 후: 앞뒤 위치 관련 문서 {reordered_quality}/6")

    if reordered_quality > original_quality:
        print("✅ 재정렬 성공! 품질이 개선되었습니다.")
    else:
        print("⚠️ 재정렬 효과가 제한적입니다.")

    return docs, reordered_docs
```

```
# 실행해보기
original, reordered = jay_perfect_reorder_solution()
```

- 커스텀된 `LongContextReorder` 적용 결과 (3.3s)

```
🔍 원본 검색 결과:
  [0] 사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.
  [1] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
  [2] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
  [3] ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.
  [4] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
  [5] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
  [6] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
```

```
    [7] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
    [8] 요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.
    [9] 인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.

🔄 LongContextReorder 적용 후:
    [0] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
    [1] ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.
    [2] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
    [3] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
    [4] 인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.
    [5] 요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.
    [6] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
    [7] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
    [8] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
    [9] 사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.

📊 품질 분석:
    원본 순서: 앞뒤 위치 관련 문서 4/6
    재정렬 후: 앞뒤 위치 관련 문서 6/6
✅ 재정렬 성공! 품질이 개선되었습니다.
```

- 체인까지 만들어보기

```
from langchain_community.document_transformers import LongContextReorder
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_community.vectorstores import Chroma

# 1. 고성능 임베딩 사용
embeddings = HuggingFaceEmbeddings(
    model_name="sentence-transformers/all-mpnet-base-v2",
    model_kwargs={'device': 'cpu'},
    encode_kwargs={'normalize_embeddings': True}
    )

# 2. 극명한 차이의 데이터 사용
texts = [
    # ChatGPT 관련 (높은 관련성) - 4개
    "ChatGPT는 OpenAI에서 개발한 혁신적인 대화형 AI 모델입니다.",
    "사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.",
    "ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.",
    "OpenAI의 ChatGPT는 지속적인 학습을 통해 성능이 개선되고 있습니다.",

    # 중간 관련성 - 2개
    "인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.",
    "자연어 처리 기술의 발전으로 인간-컴퓨터 상호작용이 크게 향상되었습니다.",

    # 낮은 관련성 - 4개
    "애플은 아이폰, 맥북, 아이패드 등 혁신적인 전자제품을 생산하는 글로벌 기업입니다.",
    "비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.",
    "축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.",
    "요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.",
    ]

# 3. 검색기 생성
retriever = Chroma.from_texts(texts, embedding=embeddings).as_retriever(
        search_kwargs={"k": 10}
    )

query = "ChatGPT에 대해 무엇을 말해줄 수 있나요?"

docs = retriever.invoke(query)

print("🔍 원본 검색 결과:")
for i, doc in enumerate(docs):
    print(f"[{i}] {doc.page_content}")

# 4. LongContextReorder 적용
reordering = LongContextReorder()
reordered_docs = reordering.transform_documents(docs)

print(f"\n🔄 LongContextReorder 적용 후:")
for i, doc in enumerate(reordered_docs):
    print(f"[{i}] {doc.page_content}")

# 5. 품질 검증
chatgpt_keywords = ['ChatGPT', 'OpenAI', '대화', 'AI']

def calculate_quality(docs):
    front_back = [0, 1, 2, 7, 8, 9]                                    # 앞 3개, 뒤 3개
    quality = 0
```

```
    for pos in front_back:
        if pos < len(docs):
            content = docs[pos].page_content
            if any(keyword in content for keyword in chatgpt_keywords):
                quality += 1
    return quality

original_quality = calculate_quality(docs)
reordered_quality = calculate_quality(reordered_docs)

print(f"\n📊 품질 분석:")
print(f"원본 순서: 앞뒤 위치 관련 문서 {original_quality}/6")
print(f"재정렬 후: 앞뒤 위치 관련 문서 {reordered_quality}/6")

if reordered_quality > original_quality:
    print("✅ 재정렬 성공! 품질이 개선되었습니다.")
else:
    print("⚠️ 재정렬 효과가 제한적입니다.")

print(docs)
print(reordered_docs)
```

- 셀 출력 (3.7s)

```
🔍 원본 검색 결과:
    [0] 사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.
    [1] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
    [2] ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.
    [3] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
    [4] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
    [5] 축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.
    [6] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.
    [7] 요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.
    [8] 인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.
    [9] 비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.

🔄 LongContextReorder 적용 후:
    [0] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
    [1] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
    [2] 축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.
    [3] 요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.
    [4] 비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.
    [5] 인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.
    [6] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.
    [7] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
    [8] ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.
    [9] 사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.

📊 품질 분석:
    원본 순서: 앞뒤 위치 관련 문서 3/6
    재정렬 후: 앞뒤 위치 관련 문서 5/6
✅ 재정렬 성공! 품질이 개선되었습니다.
```

```
    # 원본 검색 결과: print(docs)
    [Document(metadata={}, page_content='사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.'),
    Document(metadata={}, page_content='ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.'),
    Document(metadata={}, page_content='ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.'),
    Document(metadata={}, page_content='사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.'),
    Document(metadata={}, page_content='ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.'),
    Document(metadata={}, page_content='축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.'),
    Document(metadata={}, page_content='비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.'),
    Document(metadata={}, page_content='요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.'),
    Document(metadata={}, page_content='인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.'),
    Document(metadata={}, page_content='비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.')]
```

```
    # 재정렬 결과: print(reorderd_docs)
    [Document(metadata={}, page_content='ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.')
    Document(metadata={}, page_content='사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.'),
    Document(metadata={}, page_content='축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.'),
    Document(metadata={}, page_content='요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.'),
```

```
    Document(metadata={}, page_content='비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.'),
    Document(metadata={}, page_content='인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.'),
    Document(metadata={}, page_content='비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.'),
    Document(metadata={}, page_content='ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.'),
    Document(metadata={}, page_content='ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.'),
    Document(metadata={}, page_content='사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.')]
```

- `docs` 포맷 및 프린트하기

```
def format_docs(docs):
    return "\n".join([doc.page_content for i, doc in enumerate(docs)])
```

```
print(format_docs(docs))
```

- `docs` 포맷 출력

```
사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다.
ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다.
ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다.
사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다.
ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다.
축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다.
비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다.
요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다.
인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다.
비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다.
```

```
def format_docs(docs):
    return "\n".join(
        [
            f"[{i}] {doc.page_content} [source: teddylee777@gmail.com]"
            for i, doc in enumerate(docs)
        ]
    )

def reorder_documents(docs):
    reordering = LongContextReorder()                                  # 재정렬
    reordered_docs = reordering.transform_documents(docs)
    combined = format_docs(reordered_docs)
    print(combined)
    return combined
```

```
# 재정렬된 문서를 출력
_ = reorder_documents(docs)
```

- 재정렬된 문서 출력

```
[0] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다. [source: teddylee777@gmail.com]
[1] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다. [source: teddylee777@gmail.com]
[2] 축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다. [source: teddylee777@gmail.com]
[3] 요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다. [source: teddylee777@gmail.com]
[4] 비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다. [source: teddylee777@gmail.com]
[5] 인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다. [source: teddylee777@gmail.com]
[6] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다. [source: teddylee777@gmail.com]
[7] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다. [source: teddylee777@gmail.com]
[8] ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다. [source: teddylee777@gmail.com]
[9] 사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다. [source: teddylee777@gmail.com]
```

```
from langchain.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser
from langchain_core.runnables import RunnableLambda
from operator import itemgetter
from langchain_huggingface import HuggingFaceEmbeddings
from langchain_core.embeddings import FakeEmbeddings
from langchain_google_genai import ChatGoogleGenerativeAI
from google import genai
import os
import json


# 프롬프트 템플릿
```

```
template = """Given this text extracts:
{context}

-----
Please answer the following question:
{question}

Answer in the following languages: {language}
"""

# 프롬프트 정의
prompt = ChatPromptTemplate.from_template(template)

# LLM 정의
# Gemini 모델 초기화
gemini_lc = ChatGoogleGenerativeAI(model="gemini-2.5-flash-lite")

# Chain 정의
chain = (
    {
        "context": itemgetter("question")
        | retriever
        | RunnableLambda(reorder_documents),                    # 질문을 기반으로 문맥 검색하기
        "question": itemgetter("question"),                     # 질문 추출하기
        "language": itemgetter("language"),                     # 답변 언어 추출하기
    }
    | prompt                                                    # 프롬프트 템플릿에 값을 전달하기
    | gemini_lc                                                 # 언어 모델에 프롬프트 전달하기
    | StrOutputParser()                                         # 모델의 출력을 문자열로 파싱하기
)
```

```
answer = chain.invoke(
    {"question": "ChatGPT에 대해 무엇을 말해줄 수 있나요?", "language": "KOREAN"}
)
```

- `question`=쿼리 입력, `language`=언어 입력 (1.0s)

```
[0] ChatGPT는 복잡한 문제를 해결하거나 창의적인 아이디어를 제안하는 데에도 사용될 수 있습니다. [source: teddylee777@gmail.com]
[1] 사용자와 대화하는 것처럼 설계된 AI인 ChatGPT는 다양한 질문에 답할 수 있습니다. [source: teddylee777@gmail.com]
[2] 축구는 전 세계에서 가장 인기 있는 스포츠 중 하나입니다. [source: teddylee777@gmail.com]
[3] 요리는 창의성과 기술이 결합된 예술의 한 형태로 여겨집니다. [source: teddylee777@gmail.com]
[4] 비트코인은 블록체인 기술을 기반으로 한 대표적인 암호화폐입니다. [source: teddylee777@gmail.com]
[5] 인공지능 기술은 현대 사회의 많은 분야에서 혁신을 이끌고 있습니다. [source: teddylee777@gmail.com]
[6] 비트코인은 디지털 금이라고도 불리며, 가치 저장 수단으로서 인기를 얻고 있습니다. [source: teddylee777@gmail.com]
[7] ChatGPT의 기능은 지속적인 학습과 업데이트를 통해 더욱 발전하고 있습니다. [source: teddylee777@gmail.com]
[8] ChatGPT는 복잡한 질문에 대해서도 정확하고 유용한 답변을 제공할 수 있습니다. [source: teddylee777@gmail.com]
[9] 사용자와 자연스럽게 대화할 수 있는 ChatGPT는 뛰어난 언어 이해 능력을 보여줍니다. [source: teddylee777@gmail.com]
```

```
print(answer)
```

- 답변 출력하기

```
ChatGPT는 사용자와 자연스럽게 대화할 수 있도록 설계된 AI로, 다양한 질문에 답하고 복잡한 문제 해결이나 창의적인 아이디어 제안에도 활용될 수 있습니
```

- *next:* ***`상위 문서 검색기 (ParentDocumentRetriever)`***
