---

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

---

## 11. `Convex Combination(CC) 적용된 앙상블 검색기(EnsembleRetriever)`

### 1) Ensemble Retriever `Convex Combination(CC)` 추가

- [참고]: `AutoRAG가 제개한 알고리즘 방식의 차이 설명`

---

### 2) `실험을 위한 사전 셋업`

```
from langchain.retrievers import EnsembleRetriever as OriginalEnsembleRetriever
from langchain_text_splitters import RecursiveCharacterTextSplitter
from langchain_community.document_loaders import PDFPlumberLoader
from langchain_community.vectorstores import FAISS

# 문서 로드(Load Documents)
loader = PDFPlumberLoader("../10_Retriever/data/디지털_정부혁신_추진계획.pdf")

# 문서 분할(Split Documents): 테스트를 위하여 작은 Chunk Size로 설정
text_splitter = RecursiveCharacterTextSplitter(chunk_size=100, chunk_overlap=0)
split_documents = loader.load_and_split(text_splitter)

print(f"   ✅ {len(split_documents)}개 청크 생성")
```

```
   ✅ 217개 청크 생성
```

- ✅ 217개 청크 생성 (`5.7s`)

```
import gc
import numpy as np
import time
from langchain_huggingface import HuggingFaceEmbeddings
import warnings

# 경고 무시
warnings.filterwarnings("ignore")

# 임베딩
embeddings = HuggingFaceEmbeddings()

# 임베딩 차원 크기 계산해보기
dimention_size = len(embeddings.embed_query("hello world"))
print(dimention_size)                                        # 768
```

```
768
```

```
# FaissRetriever 생성

faiss = FAISS.from_documents(
    documents=split_documents, embedding=embeddings
).as_retriever(search_kwargs={"k": 5})

print("   ✅ FAISS 생성")
```

```
   ✅ FAISS 생성
```

- ✅ FAISS 생성 (`13.1s`)

```
# KiwiBM25Retriever 생성(한글 형태소 분석기 + BM25 알고리즘)
from utils.korean_bm25_retriever_2 import KoreanBM25Retriever, pretty_print

# 검색기 생성하기
bm25 = KoreanBM25Retriever.from_documents(
    documents=split_documents,
    embedding=embeddings
)
```

```
bm25.k = 5
print("   ✅ BM25 생성")
```

```
   ✅ BM25 생성
```

- ✅ BM25 생성 (`2.2s`)

```
# Langchain EnsembleRetriever 생성

original_ensemble_retriever = OriginalEnsembleRetriever(retrievers=[faiss, bm25])
```

---

### 3) `CC 방식`과 `RRF 방식`의 `EnsembleRetriever` 생성

- **`RRF` (Reciprocal Rank Fusion)**
  - 특징: 각 검색기의 **`순위를 결합`** → **`상위 순위에 높은 가중치`**
  - 공식

```
RRF_score = sum(1 / ( k + rank ))       # k = 60 (기본값)
                                        # rank = 각 검색기에서의 순위
```

  - 예시

```
문서 A:
- FAISS: 1위 → 1/(60+1) = 0.0164
- BM25: 3위 → 1/(60+3) = 0.0159
- RRF: 0.0323

문서 B:
- FAISS: 2위 → 1/(60+2) = 0.0161
- BM25: 1위 → 1/(60+1) = 0.0164
- RRF: 0.0325                                ← 1위!
```

  - 유용한 경우
    - 검색 점수를 신회할 수 없을 때
    - 다양한 검색기 조합할 때
    - 일반적인 경우 *(안전한 선택)*

- **`CC`** (Convex Combination)
  - 특징: 각 검색기의 **`점수를 결합`** → **`가중치 조정 가능`**
  - 공식

```
CC_score = w1 * score1 + w2 * score2
# w1, w2 = 가중치 (기본: 0.5, 0.5)
# score = 각 검색기의 유사도 점수
```

  - 예시

```
문서 A:
- FAISS: 0.8 → 0.5 * 0.8 = 0.4
- BM25: 0.6 → 0.5 * 0.6 = 0.3
- CC: 0.7

문서 B:
- FAISS: 0.7 → 0.5 * 0.7 = 0.35
- BM25: 0.9 → 0.5 * 0.9 = 0.45
- CC: 0.8                                    ← 1위!
```

- 유용한 경우
  - 검색 점수를 신뢰할 때
  - 특정 검색기에 가중치를 주고 싶을 때
  - 점수 기반 세밀한 조정이 필요할 때

```
from dotenv import load_dotenv
load_dotenv()

from utils.korean_bm25_retriever_2 import KoreanBM25Retriever
from utils.ensemble_retriever import (
    CustomEnsembleRetriever,
    EnsembleMethod,
    compare_methods,
)

import warnings

warnings.filterwarnings("ignore")

print("=" * 60)
print("🚀 Custom Ensemble Retriever 실습")
print("=" * 60)


# Custom Ensemble Retriever

print("\n Custom Ensemble Retriever 생성...")

# RRF 방식
rrf_ensemble = CustomEnsembleRetriever(
    retrievers=[faiss, bm25],
    method=EnsembleMethod.RRF,
    k=5
)
print("   ✅ RRF Ensemble")

# CC 방식 (균등 가중치)
cc_ensemble = CustomEnsembleRetriever(
    retrievers=[faiss, bm25],
    method=EnsembleMethod.CC,
    k=5
)
print("   ✅ CC Ensemble (균등)")

# CC 방식 (FAISS 우선)
cc_faiss_weighted = CustomEnsembleRetriever(
    retrievers=[faiss, bm25],
    method=EnsembleMethod.CC,
    weights=[0.7, 0.3],                         # FAISS 70%, BM25 30%
    k=5
)
print("   ✅ CC Ensemble (FAISS 70%)")

# CC 방식 (BM25 우선)
cc_bm25_weighted = CustomEnsembleRetriever(
    retrievers=[faiss, bm25],
    method=EnsembleMethod.CC,
    weights=[0.3, 0.7],                         # FAISS 30%, BM25 70%
    k=5
)
print("   ✅ CC Ensemble (BM25 70%)")
```

```
============================================================
🚀 Custom Ensemble Retriever 실습
============================================================

 Custom Ensemble Retriever 생성...
   ✅ RRF Ensemble
   ✅ CC Ensemble (균등)
   ✅ CC Ensemble (균등)
   ✅ CC Ensemble (FAISS 70%)
   ✅ CC Ensemble (BM25 70%)
```

- 셀 출력

```
============================================================
🚀 Custom Ensemble Retriever 실습
============================================================

Custom Ensemble Retriever 생성...
✅ RRF Ensemble
✅ CC Ensemble (균등)
```

✅ CC Ensemble (FAISS 70%)
✅ CC Ensemble (BM25 70%)

- 검색 결과 비교하기

```
print("\n 검색 비교결과...")

query = "디지털 트랜스포메이션이란 무엇인가요?"

# RRF vs CC 비교
compare_methods([faiss, bm25], query, k=5)

# 가중치 비교
print("\n" + "="*60)
print("📊 가중치별 비교")
print("="*60)

results_equal = cc_ensemble.invoke(query)
results_faiss = cc_faiss_weighted.invoke(query)
results_bm25 = cc_bm25_weighted.invoke(query)

print("\n[균등 가중치] 1위:")
print(f"  {results_equal[0].page_content[:60]}...")

print("\n[FAISS 70%] 1위:")
print(f"  {results_faiss[0].page_content[:60]}...")

print("\n[BM25 70%] 1위:")
print(f"  {results_bm25[0].page_content[:60]}...")

print("\n" + "="*60)
print("✅ 실습 완료!")
print("="*60)
```

```
 검색 비교결과...

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

============================================================
📊 RRF (Reciprocal Rank Fusion)
============================================================

[1] 점수: 0.0310
    Ⅱ. 디지털 정부혁신 추진계획 ····· 2...

[2] 점수: 0.0164
    ☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

[3] 점수: 0.0164
    Ⅱ. 디지털 정부혁신 추진계획
▸ (비전) 디지털로 여는 좋은 세상 ※ 부제 : 대한민국이 먼저 갑니다....

[4] 점수: 0.0161
    □ 추진체계 강화 △ 디지털정부혁신기획단, 디자인·개발 전문가팀 신설
△ 범정부 T/F 운영...

[5] 점수: 0.0161
    ○ (디지털 고지 수납) 각종 고지서·안내문* 등을 온라인(공공 민간)
으로 받고, 간편하게 납부할 수 있도록 디지털 고지 수납 활성화...

============================================================
📊 CC (Convex Combination)
============================================================

[1] 점수: 0.5000
    ☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

[2] 점수: 0.5000
    Ⅱ. 디지털 정부혁신 추진계획
▸ (비전) 디지털로 여는 좋은 세상 ※ 부제 : 대한민국이 먼저 갑니다....

[3] 점수: 0.3750
    □ 추진체계 강화 △ 디지털정부혁신기획단, 디자인·개발 전문가팀 신설
△ 범정부 T/F 운영...

[4] 점수: 0.3750
    ○ (디지털 고지 수납) 각종 고지서·안내문* 등을 온라인(공공 민간)
으로 받고, 간편하게 납부할 수 있도록 디지털 고지 수납 활성화...
```

```
[5] 점수: 0.2500
    하여 공공분야 디지털 전환을 위한 추진계획 마련
* 관계부처 협의 21회(행안,과기정통,기재,복지,권익위,국정원 등), 민간전문가 의견청취 10...

============================================================
```

- 검색 비교 결과

```
검색 비교결과...

============================================================
🔍 검색어: 디지털 트랜스포메이션이란 무엇인가요?
============================================================

============================================================
📊 RRF (Reciprocal Rank Fusion)
============================================================

[1] 점수: 0.0310
    Ⅱ. 디지털 정부혁신 추진계획 ····· 2...

[2] 점수: 0.0164
    ☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

[3] 점수: 0.0164
    Ⅱ. 디지털 정부혁신 추진계획
▸ (비전) 디지털로 여는 좋은 세상 ※ 부제 : 대한민국이 먼저 갑니다....

[4] 점수: 0.0161
    □ 추진체계 강화 △ 디지털정부혁신기획단, 디자인·개발 전문가팀 신설
△ 범정부 T/F 운영...

[5] 점수: 0.0161
    ○ (디지털 고지 수납) 각종 고지서·안내문* 등을 온라인(공공 민간)
으로 받고, 간편하게 납부할 수 있도록 디지털 고지 수납 활성화...

============================================================
📊 CC (Convex Combination)
============================================================

[1] 점수: 0.5000
    ☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

[2] 점수: 0.5000
    Ⅱ. 디지털 정부혁신 추진계획
▸ (비전) 디지털로 여는 좋은 세상 ※ 부제 : 대한민국이 먼저 갑니다....

[3] 점수: 0.3750
    □ 추진체계 강화 △ 디지털정부혁신기획단, 디자인·개발 전문가팀 신설
△ 범정부 T/F 운영...

[4] 점수: 0.3750
    ○ (디지털 고지 수납) 각종 고지서·안내문* 등을 온라인(공공 민간)
으로 받고, 간편하게 납부할 수 있도록 디지털 고지 수납 활성화...

[5] 점수: 0.2500
    하여 공공분야 디지털 전환을 위한 추진계획 마련
* 관계부처 협의 21회(행안,과기정통,기재,복지,권익위,국정원 등), 민간전문가 의견청취 10...

============================================================
📊 차이 분석
============================================================

⚠️ RRF와 CC가 다른 1위 선택!

RRF 1위: Ⅱ. 디지털 정부혁신 추진계획 ····· 2...
CC 1위:  ☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

============================================================
📊 가중치별 비교
============================================================
```

```
[균등 가중치] 1위:
☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

[FAISS 70%] 1위:
☞ (기존) 공공시설 이용료 감면 혜택을 주기 위해 이용자에게 각종 증명서(장애인증명, 기초생활...

[BM25 70%] 1위:
Ⅱ. 디지털 정부혁신 추진계획
▸ (비전) 디지털로 여는 좋은 세상 ※ 부제 : 대한민국이 먼저 갑니다....

============================================================
✅ 실습 완료!
============================================================
```

---

- next: **`CH11. 리랭커 (Reranker)`**

---