2조: 해조 [김지은, 박다정, 오학성, 정소영, 황수현]

산출물

Ⅰ. 수집된 데이터 및 데이터 전처리 결과서

---

1. 데이터 전처리 목적 및 개요

본 문서는 HuggingFace Weekly Papers에서 제공하는 최신 AI·ML·DL·LLM 논문 데이터를 기반으로 구축한RAG(Retrieval-Augmented Generation) 챗봇 시스템의 데이터 전처리 산출물을 정리한 보고서이다.

RAG 시스템의 검색 품질과 응답 정확도는 초기 데이터 전처리 단계에서 문서를 어떻게 정제·구조화하느냐에 크게 의존한다.

이에 따라 본 전처리 과정은 다음의 목적을 중심으로 설계되었다.

- 최신 연구 데이터 확보 및 구조화
  - HuggingFace Weekly Papers에서 최근 10주간 논문 메타데이터와 초록을 수집하여 RAG 학습 및 검색에 활용 가능한 표준화된 JSON 문서로 가공한다.
- 검색 품질 향상을 위한 Chunking 및 Embedding 최적화
  - 서로 다른 chunk_size/overlap 조합을 실험하여 가장 정보 손실이 적고 검색 효율이 높은 조건을 선정한다.
    (자세한 실험 조건 및 비교 결과는 Ⅱ. 데이터 계획 및 결과보고서에 포함한다.)
- 검색 효율 향상을 위한 Auto-clustering
  - NLTK 기반 전처리 후 TF-IDF 및 KeyBERT를 이용하여 문서별 핵심 키워드를 추출하고 초기 검색 필터링 전략을 검증하였다.
  - 다만 태그가 1,455개 이상으로 분산되어 검색 품질이 저하되는 한계를 확인하였고, 이를 보완하기 위해 Auto-Clustering 기반 주제 그룹화를 적용하였다.
    (세부 태그 분포 및 분석 결과는 Ⅱ. 데이터 계획 및 결과보고서에 상세 기술한다.)
- VectorDB 구축을 위한 전처리 일관성 확보
  - 크롤링링 → 청크 → 임베딩 → 클러스터링 → VectorDB 구축까지의 파이프라인을 자동화하여 RAG 시스템의 재현성·확장성을 보장한다.

본 보고서는 위 목표 달성을 위한 데이터 수집, Chunking, VectorDB (Auto-Clustering, Embedding 포함) 구축 과정을 단계별로 설명하고 최종 전처리 산출물을 체계적으로 제시한다.

---

2. 데이터 수집(Crawling)

2.1 Crawling 대상

- HuggingFace Weekly Papers
- 최근 10주차 자료 수집 (2025년 40주~49주)

2.2 Crawling 방식

- Python 기반 Requests/BeautifulSoup 사용
- HuggingFace Weekly Papers 상세 페이지 접근 후 아래 항목 추출
- 요청 과부하 방지를 위해 time.sleep(160초) 적용

2.3 Crawling 데이터 구조(JSON)

- 수집 결과는 문서 단위로 JSON 파일 저장.

필드	설명	예시
context	초록	Large Language Models (LLMs) have demonstrated remarkable capabilities ~
title	논문 제목	Souper-Model: How Simple Arithmetic Unlocks State-of-the-Art LLM Performance
authors	저자 목록	Shalini Maiti, Amar Budhiraja, Bhavul Gauri
publication_year	발행 연도	2025
github_url	관련 코드 저장소	https://github.com/facebookresearch/llm_souping
hunggingface_url	초록 페이지 링크	https://huggingface.co/papers/2511.20626
upvote	인기도 지표	134

2.3 Crawling 결과

- 1031개 문서 로딩 완료

---

3. Chunking

3.1 Chunking strategy

- embedding model에 따라 하기 chunking 조건 변경하여 최적의 chunking 조건 도출
- 자세한 테스트 내용은 II. 테스트 계획 및 결과보고서를 참조조

	exp1.	exp2.	exp3.	exp4.	exp5.	exp.6
chunk_size	100	200	300	100	200	300
chunk_overlap	15	30	45	0	0	0

*reference : kt cloud Tech Blog_ [Tech Series] kt cloud AI 검색 증강 생성(RAG) #3 : 청킹(Chunking) 전략과 최적화

3.2 Chunking 결과

- chunk_size=200, chunk_overlap=30
- 1031개 문서에 대해서 20,695개 청크 생성 (평균 20.1개 청크/문서)

---

4. Auto clustering

4.1 Auto clustering 목적

- 크롤링 과정에서 TF-IDF or KeyBERT 기반 태그가 문서마다 3개씩 생성되었으나, 전체 태그 수가 1,455개 이상으로 증가하면서 RAG 검색의 효율성이 크게 저하됨
  → 과도한 태그 다양성으로 인해 관련 문서를 제대로 찾지 못하는 문제 해결
- 문서의 주제 기반 의미적 그룹화 → 검색 효율 향상
- RAG 시스템에서의 semantic drift 감소

4.2 처리 단계

1) 임베딩 재사용 (Cache 기반)
   - OpenAI text-embedding-3-small 모델 사용
   - 기존에 임베딩된 문서는 캐시에서 재사용
   - 신규 문서만 추가로 임베딩 → 불필요한 API 비용 절감 및 시간 최적화

---

2) 최적 클러스터(K) 자동 선정
   - K=10~30 범위에서 inertia 계산
   - 1차/2차 미분 기반 elbow-point 자동 탐색
   - 가장 의미적 변곡점이 되는 지점에서 optimal K 결정 → 군집 수 자동화
3) K-Means 기반 군집화
   - Scikit-Learn KMeans 적용
   - 파라미터: max_iter=300, n_init=10
   - 출력: 문서별 cluster_id 할당
4) Cluster 데이터
   - cluster_assignments.json : 각 문서별 cluster_id 및 중심까지의 distance 정보
   - cluster_metadata.json :

항목	설명
cluster_size	군집 내 문서 개수
representative_keywords	TF-IDF 기반 대표 토픽 키워드
cluster_center	임베딩 평균값
mean_distance	군집 중심으로부터 평균 거리
density	평균 거리 기반 밀집도
top-3 representative documents	중심에 가장 가까운 문서 3개
upvote_mean	평판 기반 cluster-level score

4.3 clustering 결과

- 클러스터 키워드

```
클러스터 ID: 0 / 문서 수: 57
  키워드: [ image, feature, high, http, github ]

클러스터 ID: 1 / 문서 수: 83
  키워드: [ visual, multimodal, performance, image, token ]

클러스터 ID: 2 / 문서 수: 58
  키워드: [ video, generation, motion, world, scene ]

클러스터 ID: 3 / 문서 수: 85
  키워드: [ research, reasoning, llm, tool, training ]

클러스터 ID: 4 / 문서 수: 26
  키워드: [ llm, reasoning, knowledge, bench, agent ]
```

```
클러스터 ID: 5 / 문서 수: 53
  키워드: [ training, quality, long, diffusion, prompt ]

클러스터 ID: 6 / 문서 수: 76
  키워드: [ training, reasoning, llm, policy, performance ]

클러스터 ID: 7 / 문서 수: 37
  키워드: [ instruction, generation, image editing, control, video ]

클러스터 ID: 8 / 문서 수: 65
  키워드: [ llm, code, language, large, framework ]

클러스터 ID: 9 / 문서 수: 60
  키워드: [ generation, image, quality, step, training ]

클러스터 ID: 10 / 문서 수: 31
  키워드: [ spatial, reasoning, scene, language, multimodal ]

클러스터 ID: 11 / 문서 수: 81
  키워드: [ training, attention, llm, token, context ]

클러스터 ID: 12 / 문서 수: 44
  키워드: [ vla, reasoning, robot, world, real ]

클러스터 ID: 13 / 문서 수: 60
  키워드: [ system, environment, world, benchmark, scientific ]

클러스터 ID: 14 / 문서 수: 28
  키워드: [ grpo, agent, llm, reward, reasoning ]

클러스터 ID: 15 / 문서 수: 53
  키워드: [ video, image, generation, multimodal, framework ]


클러스터 ID: 16 / 문서 수: 53
  키워드: [ generation, image, understanding, multimodal, unified ]

클러스터 ID: 17 / 문서 수: 81
  키워드: [ training, llm, performance, language, problem ]
```

- 전체 키워드 통계 (빈도순)

```
================================================================================
전체 키워드 통계 (빈도순)
================================================================================
총 키워드 수: 90
고유 키워드 수: 50
================================================================================

llm           : 7      reasoning     : 6      training      : 6      image         : 5      generation    : 5
multimodal    : 4      performance   : 3      video         : 3      world         : 3      language      : 3
token         : 2      scene         : 2      agent         : 2      quality       : 2      framework     : 2
feature       : 1      high          : 1      http          : 1      github        : 1      visual        : 1
motion        : 1      research      : 1      tool          : 1      knowledge     : 1      bench         : 1
long          : 1      diffusion     : 1      prompt        : 1      policy        : 1      instruction   : 1
image editing : 1      control       : 1      code          : 1      large         : 1      step          : 1
spatial       : 1      attention     : 1      context       : 1      vla           : 1      robot         : 1
real          : 1      system        : 1      environment   : 1      benchmark     : 1      scientific    : 1
grpo          : 1      reward        : 1      understanding : 1      unified       : 1      problem       : 1

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

---

5. Embedding

- 실행 파일 경로: vectorDB 실행시, 자동 실행됨.
- 파일 저장 경로: vector_db에 함께 반영
- chunking size, overlap 6가지 및 Embedding 모델(7가지)을 변화하여 실험 진행
  (산출물 II. 테스트 계획 및 결과 보고서 파일 참조)
- 최적의 조건을 하기와 같이 도출함
  - ① Chunk =200, overlap=30 / paraphrase-multilingual-mpnet-base-v2
  - ② Chunk =300, overlap=45 / text-embedding-3-small
- 본 챗봇에 적용된 조건 : Chunk =300, overlap=45 / text-embedding-3-small (OpenAI)

---

6. VectorDB (Embedding + Clustering 기반)

6.1 vectorDB 목적 및 역할

- VectorDB 구축 자동화
  - 기존 크롤링 문서(.pkl)를 자동으로 로드하여 청크 단위로 임베딩
  - 다양한 임베딩 모델(OpenAI / HuggingFace)을 선택적으로 적용
- Auto-Clustering 기반 메타데이터 강화 (→ 4. Auto-Clustering)
  - 문서 단위 클러스터링 수행 후 각 청크에 cluster_id 부착
  - 유사 문서끼리 의미적 그룹화되어 검색 효율이 높아짐
- 검색 효율 향상
  - 클러스터 기반 검색 제한
  - VectorDB의 복잡도 완화 및 성능 향상

6.2 vectorDB 구조

크롤링 문서
↓ chunking
문서 청크 (.pkl)
↓
Auto Clustering (cluster_id)
↓
임베딩 모델 적용
↓
Chroma VectorDB 구축

---

7. 전처리 결과 요약

- 총 수집 문서 수: 1031건
- 최적 chunk 조건: Chunk =300, overlap=45
- 생성된 chunk 수: 20,695개
- 클러스터 수: 17개 (Auto-selected K)
- 최종 선택된 Embedding 모델: text-embedding-3-small (OpenAI)
- 구축된 VectorDB: Chroma 기반, 클러스터 메타데이터 포함