인공지능 데이터 전처리 결과서

1. 데이터 개요

1.1. 전처리 목적

- **RAG 최적화 지식 DB 및 RDB 구축**

  상담 데이터를 시스템 학습 및 검색에 적합한 형태로 구조화하여 답변의 일관성 확보

- **RAG 검색 정확도 향상**

  비정형 데이터의 형식을 통일하고 정제하여 검색 정확도 및 응답 품질 향상

1.2. 문제 정의

구분	데이터명	데이터 수집 목적	데이터 형태	데이터 출처
DATA-001	K패스 FAQ	상담 답변용 RAG 문서 구축 (카드 정보 Index)	웹페이지 → JSON	https://korea-pass.kr/notice/faqList.do
DATA-002	국민행복카드 FAQ	상담 답변용 RAG 문서 구축 (카드 정보 Index)	웹페이지 → JSON	http://www.voucher.go.kr/customer/faq/list.do
DATA-003	나라사랑카드 FAQ	상담 답변용 RAG 문서 구축 (카드 정보 Index)	웹페이지 → JSON	https://www.mnd.go.kr/mbshome/mbs/mnd/subview.jsp?id=mr
DATA-004	민생회복소비쿠폰 FAQ	상담 답변용 RAG 문서 구축 (카드 정보 Index)	웹페이지 → JSON	https://www.hyundaicard.com/cpb/gs/CPBGS2011_01.hc
DATA-005	특수목적카드 약관 - 국민행복카드 - 나라사랑체크카드 - 쿠팡와우카드 - 서울시다둥이행복카드 - 네이버페이카드 - 라이프파트너카드	상담 답변용 RAG 문서 구축 (카드 정보 Index)	PDF → JSON	https://card.kbcard.com/SVC/DVIEW/HSHMCXCRSZZC0002 https://www.shinhancard.com/mob/MOBFM12051N/MOBFM12(https://www.samsungcard.com/company/IR/announce/product
DATA-006	신한카드 카드상품별 약관	상담 답변용 RAG 문서 구축 (카드 정보 Index)	PDF → MD	https://www.shinhancard.com/mob/MOBFM12051N/MOBFM12(
DATA-007	삼성카드 신용카드 가이드	상담 답변용 RAG 문서 구축 (카드사 이용 안내 Index)	웹페이지 → JSON	https://www.samsungcard.com/personal/customer-service/cre
DATA-008	삼성카드 금융안내	상담 답변용 RAG 문서 구축 (카드사 이용 안내 Index)	웹페이지 → JSON	https://www.samsungcard.com/home/main/finance/PGHPPCC

구분	데이터명	데이터 수집 목적	데이터 형태	데이터 출처
DATA-009	신한카드 이용약관	상담 답변용 RAG 문서 구축 (카드사 이용 안내 Index)	웹페이지 → JSON	이용약관 < 고객센터 < 신한카드
DATA-010	현대 애플페이 이용 안내	상담 답변용 RAG 문서 구축 (카드사 이용 안내 Index)	웹페이지 → JSON	https://www.hyundaicard.com/cpu/ug/CPUUG4001_01.hc
DATA-011	소비자 주의 경보	상담 답변용 RAG 문서 구축 (공지사항 RDB)	웹페이지 → JSON	https://www.samsungcard.com/personal/notice/alert/UHPPCC(
DATA-012	삼성카드 공지사항	조회용 DB 구축 (공지사항 RDB)	웹페이지 → JSON	https://www.samsungcard.com/personal/notice/news/UHPPCC
DATA-013	하나카드 통합 상담 데이터	상담 사례 RAG 문서 구축 (상담 사례 Index, 상담 사례 RDB)	JSON → CSV	https://www.aihub.or.kr/aihubdata/data/view.do?srchOptnCnd=OPTNCND001&currMenu=115&topMenu=100&se

- 데이터 수집 기간: 2026.01.02-2026.01.05
- 전체 수집 데이터 건수
  - 카드 정보 Index 구축용 : 512건
  - 카드사 이용 안내 Index 구축용 : 281건
  - 상담 사례 Index 구축용 : 6533건
  - 공지사항 RDB 구축용: 52건

---

2. 데이터 수집 및 활용의 적법성 검토

구분	상업 이용 가능	학습 사용 허용	크롤링 이용약관 준수	개인정보 보호	비고
DATA-001	X	O	O	O	
DATA-002	X	O	O	O	
DATA-003	O	O	O	O	출처 표시 후 사용 가능
DATA-004	X	O	O	O	
DATA-005	X	O	O	O	국민카드 - 카드사의 서비스 정보를 이용하여 얻은 정보를 카드사의 사전 승낙 없이 복제 또는 유통시키거나 상업적으로 이용하는 행위만 불가능
DATA-006	X	O	O	O	
DATA-007	O	O	O	O	
DATA-008	X	O	O	O	
DATA-009	X	O	O	O	
DATA-010	X	O	O	O	
DATA-011	X	O	O	O	
DATA-012	X	O	O	O	
DATA-013	X	O	O	O	

- **저작권 준수**

  해당 데이터의 이용약관을 검토하여 상업적 이용 및 2차 가공 가능 여부 확인
- **크롤링 이용약관 준수**

  Robots.txt 규정에 따라 서버에 부하를 주지 않는 방식으로 크롤링 수행
- **개인정보 보호**

수집 과정에서 개인 식별 정보(이름, 연락처 등)는 비식별 처리

---

3. 데이터 수집 방법

1. DATA-001 ~ DATA-004

- 수집 방식 및 도구: Python 기반 웹 크롤링 방식으로 데이터 수집
  - FAQ 목록 페이지 접근 후 개별 상세 페이지 HTML 파싱 : `BeautifulSoup` , `Selenium` , `webdriver`
  - JSON 파일로 저장 : `question` , `answer` , `category` 로 나눠 저장

2. DATA-005

- 수집 방식 및 도구: pdf 수집 후 텍스트 파싱
  - pdf 개별 상세 페이지 파싱 : `PdfPlumber`
  - JSON 파일로 저장 : `card_name` , `content` , `category` 로 나눠 저장

3. DATA-006

- 수집 방식 및 도구: pdf 수집 후 텍스트 파싱
  - pdf 개별 상세 페이지 파싱 : `LlamaParse`
  - Markdown 파일로 저장 : 큰 글자를 기준으로 섹션을 나눈 후, 마크다운 문법으로 `제목` 과 `내용` 형식으로 저장

4. DATA-007, DATA-008, DATA-010~DATA-012

- 수집 방식 및 도구: Python 기반 웹 크롤링 방식으로 데이터 수집
  - FAQ 목록 페이지 접근 후 개별 상세 페이지 HTML 파싱 : `BeautifulSoup`
  - JSON 파일로 저장 : `id` , `metadata{title, category}` 로 나눠 저장

5. DATA-009

- 수집 방식 및 도구: Python 기반 웹 크롤링 방식으로 데이터 수집
  - FAQ 목록 페이지 접근 후 개별 상세 페이지 HTML 파싱 : `BeautifulSoup`
  - JSON 파일로 저장 : `title` , `content` , `category` 로 나눠 저장

6. DATA-013

- 수집 방식 및 도구: AI Hub에서 다운로드 후 전처리
  - AI Hub에서 원문 데이터(json) 다운로드
  - CSV 파일로 병합 : 개발 과정 중 데이터 조회의 용이를 위함
  - JSON 파일로 저장 : 각 상담별로 개별 파일로 저장되어 있던 형태에서 단일 파일로 저장

---

4. 데이터 저장 및 관리

1. DATA-001 ~ DATA-004

- 저장 형식: JSON

- 저장 환경: 로컬 서버

▼ 데이터 구조

```
// 형식
{
  "id": "식별 ID",
  "title": "질문",
  "content": "답변",
  "text": "임베딩할 실제 텍스트 내용 (질문+답변)",
  "metadata": {
    "card_name": "카드이름",            # 1차 필터링
    "category": "카테고리",             # 2차 필터링
  }
}
```

```
// 예시
{
  "id": "card_012",
  "title": "재발급 신청은 본인만 할 수 있나요?",
  "content": "나라사랑카드 재발급을 포함한 모든 금융업무는 본인만 가능합니다.(부모님, 친구 등 대리인 불가) 현역병사는 병영생활에 불편이 없도록 분실, 파손 등에 주의하여 나라사랑카드를 사용하기 바랍니다.",
  "text": "재발급 신청은 본인만 할 수 있나요?\n나라사랑카드 재발급을 포함한 모든 금융업무는 본인만 가능합니다.(부모님, 친구 등 대리인 불가) 현역병사는 병영생활에 불편이 없도록 분실, 파손 등에 주의하여 나라사랑카드를 사용하기 바랍니다.",
  "metadata": {
    "card_name": "나라사랑카드",
    "category": "발급/신청",
  }
}
```

▼ 데이터 정제 및 전처리

- FAQ - 질문/답변 형태 유지
- 불필요한 공백 제거
- 문서 검색에 불필요한 문구 처리
  - 예: 국민행복카드를 이용해 주셔서 감사합니다.

```
# 특정 어휘를 포함한 line은 무시
"감사합니다", "감사드려요", "감사해요"
```

- 특수문자 제거

```
"●▶※■◆!•/·[] ·◌"
```

- 데이터 구조 형식에 맞게 전처리
- 메타데이터로 card_name과 category를 설정하여 각 데이터에 대한 이름, 카테고리로 그룹화

2. DATA-005

- 저장 형식: JSON
- 저장 환경: 로컬 서버

▼ 데이터 구조

```
// 형식
{
```

```
  "id": "식별 ID",
  "title": "제목",
  "content": "본문",
  "text": "임베딩할 실제 텍스트 내용 (제목+본문)",
     "metadata": {
    "card_name": "카드이름",             # 1차 필터링
    "category": "카테고리",              # 2차 필터링
  }
}
```

```
// 예시
{
    "id": "card_100",
    "title": "스타벅스 할인",
    "content": "스타벅스 20% 환급할인  건당 이용금액 1만원 이상 시 건당 최대 이용금액 2만원까지 할인 적용(최대 할인액 4,000원)  상품권 구매 및 스타벅스 카드 충전 시 할인 적용 제외  백화점대형마트 등 입점된 일부 매장은 할인 적용에서 제외",
    "text": "스타벅스 할인 스타벅스 20% 환급할인  건당 이용금액 1만원 이상 시 건당 최대 이용금액 2만원까지 할인 적용(최대 할인액 4,000원)  상품권 구매 및 스타벅스 카드 충전 시 할인 적용 제외  백화점대형마트 등 입점된 일부 매장은 할인 적용에서 제외",
    "metadata": {
      "card_name": "나라사랑체크카드",
      "category": "혜택/할인"
    }
  }
```

▼ 데이터 정제 및 전처리

- 불필요한 공백 제거
- 특수문자 제거

  ```
  "●▶※■◆!•/·[] ·◌"
  ```

- 데이터 구조 형식에 맞게 전처리
- 메타데이터로 card_name과 category를 설정하여 각 데이터에 대한 이름, 카테고리로 그룹화

3. DATA-006

- 저장 형식: Markdown
- 저장 환경: 로컬 서버

▼ 데이터 구조

```
# #Pay 신한카드

# 금융소비자 보호제도 안내

- 금융소비자는 금융소비자보호법 제 19조 제 1항에 따라 해당 금융상품 또는 서비스에 대하여 설명받을 권리가 있으며, 그 설명 듣고 내용을 충분히 이해한 후 거래하시기 바랍니다.
- 신용카드 발급이 부적정한 경우(개인신용평점 낮음, 연체(단기 포함) 사유 발생 등), 카드발급이 제한 될 수 있습니다.
- 카드 이용대금과 이에 수반되는 모든 수수료는 고객님께서 지정하신 결제일에 상환하여야 합니다.

# 부가서비스 변경안내

- 카드이용시 제공되는 포인트 및 할인혜택 등의 부가서비스는 카드 신규출시(2021년 07월 01일) 이후 3년 이상 축소, 폐지 없이 유지됩니다.
- 상기에도 불구하고, 다음과 같은 사유가 발생한 경우 카드사는 부가서비스를 변경할 수 있습니다.
```

▼ 데이터 정제 및 전처리

- PDF의 구조를 유지하며 마크다운 형식으로 파싱

4. DATA-007 ~ DATA-010

- 저장 형식: JSON
- 저장 환경: 로컬 서버

▼ 데이터 구조

```
// 형식
{
  "id": "식별 ID",
  "title": "제목",
  "content": "본문",
  "text": "임베딩할 실제 텍스트 내용 (제목+본문)",
  "metadata": {
    "category1": "대분류",              # 1차 필터링
    "category2": "중분류",              # 2차 필터링
  }
}
```

```
// 예시
{
    "id": "guide_14",
    "title": "상환방법 안내",
    "content": "1. 원리금 균등: 대출기간 동안 매월 같은 금액(원금+이자)을 납부하는 방식입니다. 매월 납부금액이 같기 때문에 지출 계획을 세우기 좋습니다.\n2. 원금 균등: 대출기간 동안 원금을 매달 같은 금액으로 납부하는 방식입니다. 만기 시점과 가까워질수록 상환해야 할 금액이 줄어듭니다.\n3. 거치 후 원리금 균등: 일정기간 동안 이자만 갚고 싶다면 거치기간 동안 이자만 납입하고 그 이후로는 매월 같은 금액(원금+이자)을 납부하는 방식입니다.\n4. 만기일시: 이자만 내고 원금은 나중에 갚고 싶다면 매달 이자만 상환하고, 만기일에 원금을 상환합니다. 만기 시점에 일시 상환 또는 대출기간을 연장할 수 있습니다.",
    "text": "상환방법 안내 1. 원리금 균등: 대출기간 동안 매월 같은 금액(원금+이자)을 납부하는 방식입니다. 매월 납부금액이 같기 때문에 지출 계획을 세우기 좋습니다.\n2. 원금 균등: 대출기간 동안 원금을 매달 같은 금액으로 납부하는 방식입니다. 만기 시점과 가까워질수록 상환해야 할 금액이 줄어듭니다.\n3. 거치 후 원리금 균등: 일정기간 동안 이자만 갚고 싶다면 거치기간 동안 이자만 납입하고 그 이후로는 매월 같은 금액(원금+이자)을 납부하는 방식입니다.\n4. 만기일시: 이자만 내고 원금은 나중에 갚고 싶다면 매달 이자만 상환하고, 만기일에 원금을 상환합니다. 만기 시점에 일시 상환 또는 대출기간을 연장할 수 있습니다.",
    "metadata": {
      "category1": "금융안내",
      "category2": "카드대금 납부"
    }
  },
```

▼ 데이터 정제 및 전처리

- 항목/세부내용 형태 유지하여 각 title/content에 저장
- 메타데이터로 category1과 category2를 설정하여 각 데이터에 대한 대분류/중분류로 그룹화

5. DATA-011 ~ DATA-012

- 저장 형식: JSON
- 저장 환경: 로컬 서버

▼ 데이터 구조

```
// 형식
{
```

```
  "id": "식별 ID",
  "tag": "태그(이벤트/긴급/시스템/피해)",
  "title": "제목",
  "content": "본문",
  "date": "날짜"
}
```

```
// 예시
{
    "id": "공지사항_15",
    "tag": "[이벤트]",
    "title": "아시아나항공 마일리지 전환 종료 안내",
    "content": "아시아나항공 마일리지 전환 서비스가 2026.6.30(화)까지만 운영될 예정입니다.\n\n신청 마감일\n\n2026.6.30(화)\n\n* 자세한 전환방법 및 대상카드는 삼성카드 홈페이지 또는 모니모 앱 '마이삼성 → 카드 → 포인트 조회 → 포인트 사용 → 아멕스 제휴사 마일리지 전환'에서 확인",
    "date": "2025.12.30"
}
```

▼ 데이터 정제 및 전처리

- 제목-내용 형태로 저장
- 문서 검색에 불필요한 문구 처리
  - 예: 삼성카드를 사용해주셔서 감사합니다.

```
# 특정 어휘를 포함한 line은 무시
"감사합니다", "감사드려요", "감사해요"
```

- 공지 구분을 위한 태그 부여
  - "이벤트", "긴급", "시스템", "피해" 4개의 태그를 설정
  - 소비자주의경보는 "피해" 태그를 가진 공지사항으로 편입

6. DATA-013

- 저장 형식: JSON
- 저장 환경: 로컬 서버

▼ 데이터 구조

```
// 형식 (RDB용)
{
    "id": "hana_consultation_{source_id}",
    "source_id": "source_id",
    "consulting_category": "카테고리",
    "status": "상태",
    "client_id": "HANA_CLT_{고객id}",
    "client_name": "고객명",
    "client_phone": "전화번호",
    "client_gender": "성별",
    "client_age": "연령대",
    "call_duration": "상담 시간",
    "consulting_turns": "대화턴 수",
    "keywords": "키워드"
}
```

```
// 예시 (RDB용)
{
   "id": "hana_consultation_20593",
   "source_id": "20593",
   "consulting_category": "도난/분실 신청/해제",
   "status": "완료",
   "client_id": "HANA_CLT_82d857dd",
   "client_name": "[고객명#1]",
   "client_phone": "[전화번호#1]",
   "client_gender": "여자",
   "client_age": "50대",
   "call_duration": 166,
   "consulting_turns": 37,
   "keywords": "도난/분실 신청/해제,카드,결제,발급,이용"
}
```

```
// 형식 (VectorDB용)
{
   "id": "hana_consultation_{source_id}",
   "consultation_id": "CS-HANA-{source_id}",
   "document_type": "consultation_transcript",
   "title": "{category} 상담",
   "content": "전처리된 상담 대화 내용 ([타입#번호] 형식 태그)",
   "metadata": {
      "source_id": "{source_id}",
      "category": "카테고리",
      "keywords": ["키워드"],
      "slot_types": ["상담원명", "고객명", "초등학교명"],
      "scenario_tags": ["시나리오 태그"],
      "summary": "요약",
      "created_at": "생성된 날짜 및 시간"
   }
}
```

```
// 예시 (VectorDB용)
{
   "id": "hana_consultation_20593",
   "consultation_id": "CS-HANA-20593",
   "document_type": "consultation_transcript",
   "title": "도난/분실 신청/해제 상담",
   "content": "상담사: 상담원 [상담원명#1]입니다.\\n손님: 저 [카드사명#1]카드 문의좀 드릴려고요.\\n상담사: 고객님. 그럼 본인 확인 후 안내를 해드리겠습니다. 고객님 성함과 생년월일 말씀해 주시겠어요?\\n손님: [고객명#1]이고요, [생년월일#1]요.",
   "metadata": {
      "source_id": "20593",
      "category": "도난/분실 신청/해제",
      "keywords": ["카드", "결제", "발급"],
      "slot_types": ["[상담원명#1]", "[고객명#1]", "[카드사명#1]", "[생년월일#1]"],
      "scenario_tags": ["본인확인", "카드교체발급"],
      "summary": null,
      "created_at": "2025-01-06T23:45:00.000Z"
   }
}
```

▼ 데이터 정제 및 전처리

- 마스킹 기호 통일화

    - 예) `▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲` → `[카드번호#1]`
    - 예) `▲▲▲▲▲▲▲▲▲▲▲` → `[전화번호#1]`
    - 예) `▲▲▲초등학교` → `[초등학교명#1]`
    - 예) `▲▲▲` → `[고객명#1]` (문맥 기반)
- 불용어 처리
    - 반복 불용어 축소: `네 네 네.` → `네.`, `그 그` → `그`, `아 아` → `아`
    - 구두점 정리: `네.,` → `네.`
- LLM 기반 문맥 태깅
    - 정규식으로 처리 불가능한 마스킹을 문맥 분석하여 적절한 태그로 변환
    - 동일 개체는 동일 번호 유지 (Entity Tracking)
    - 예: 손님이 말한 이름과 상담사가 확인한 이름은 같은 번호 사용
- **처리 단계 설명:**
    1. **정규식 전처리**: 고정 길이 패턴을 먼저 처리하여 LLM 부하 감소
    2. **LLM 2단계 처리**: 문맥 분석을 통한 정확한 태깅
    3. **검증 및 재처리**: 품질 보장을 위한 자동 검증
    4. **2차 전처리:** 불용어 제거 및 태그 통합으로 최종 정제

```
flowchart TD
subgraph Input [입력]
A[CSV 원본 데이터]
end

subgraph Regex [정규식 전처리]
   B1["16자리 ▲ → 카드번호#1"]
   B2["10-11자리 ▲ → 전화번호#1"]
   B3["12-15자리 ▲ → 개인정보_구성요소"]
end

subgraph LLM [LLM 2단계 처리]
   C1["1단계: 전체 대화 분석 및 개체 식별"]
   C2["2단계: 태그 번호 할당 및 적용"]
end

subgraph Validation [검증]
   D{"▲ 잔존?"}
   D1["재처리"]
end

subgraph PostProcess [후처리]
   E1["반복 불용어 축소"]
   E2["구성요소 태그 병합"]
end

subgraph Output [출력]
   F1[hana_vectordb.json]
   F2[hana_rdb_metadata.json]
end

A --> B1
A --> B2
A --> B3
B1 --> C1
```

```
B2 --> C1
B3 --> C1
C1 --> C2
C2 --> D
D -->|Yes| D1
D1 --> C1
D -->|No| E1
E1 --> E2
E2 --> F1
E2 --> F2
```

---

5. 데이터 전처리

5.1. 이상치 탐지 및 처리

- 이상치 기준: 의미 기반 정성평가
- 처리 방법: 해당 텍스트 삭제
- 처리 결과: 상담 응대에 직접적으로 필요없는 문구 삭제

5.2. 결측치 처리

- 대상 결측 필드: 없음
- 결측치 발생 건수(비율): 0건 (0%)
- 처리 방법: 적용 없음
- 처리 결과: 확인된 결측치 없음

5.2. 데이터 정제

(1) 필드명 표준화 적용

- 기준
  - 공통 규칙(`id`, `text`, `metadata`)에 따라 필드명을 변환 및 매핑
- 처리 과정
  - 데이터 소스별로 상이하게 사용되던 필드명(`Id`, `ID`, `content`, `body` 등)을 식별
- 적용 내용
  - `id`, `text`, `metadata` 등 공통 키 구조 유지
  - 서로 다른 표기 형태의 필드명을 단일 표준으로 통합 관리
- 활용 방안
  - 표준화된 필드명을 기반으로 RAG 인덱싱 및 쿼리 파이프라인을 구성
  - 데이터 처리 자동화 및 검색 로직의 일관성 확보

(2) 특수문자 처리

- 기준
  - `●▶※■◆!•/·[] ·◌` 등 의미 없는 특수문자는 제거
  - 문장의 의미에 영향을 주는 구두점(마침표, 쉼표 등)은 유지
- 처리 과정
  - 정규표현식 기반 필터를 적용하여 불필요한 특수문자를 탐지 및 제거

    - 중복 기호, 비정상 공백, 불필요한 문자 패턴 정리
- 적용 내용
    - 텍스트 내 임베딩에 영향을 주지 않는 노이즈 문자 제거
    - 문장 구조와 의미는 보존한 상태로 텍스트 정제
- 활용 방안
    - 임베딩 벡터의 품질 향상
    - 불필요한 토큰 생성을 방지하여 검색 노이즈 감소

(3) 벡터 DB별 형식 정규화

- 기준
    - 모든 데이터를 `id`, `title`, `content`, `text`, `metadata` 구조의 공통 스키마로 통합
- 처리 과정
    - 원천 데이터(JSON, FAQ, PDF 파싱 결과 등)의 구조를 분석
    - 벡터 DB 적재에 적합한 공통 포맷으로 변환
- 적용 내용
    - `text` 필드에 실제 검색 대상 본문 텍스트 저장
    - `metadata` 필드에 `title`, `category` 등 검색 필터용 정보 저장
- 활용 방안
    - 벡터 검색 시 메타데이터 기반 필터링 적용
    - 검색 정확도 및 응답 관련성 향상

---

6. 전처리 프로세스 개요

- 전체 흐름도:
  ① 수집→ ② 데이터 전처리 → ③ 정규화 → ④ 데이터 청킹 → ⑤ 데이터 형식 통일
- 전처리 파이프라인 요약:

단계	목적	수행 작업	사용 도구/라이브러리
수집	다양한 카드 및 공공 FAQ 및 문서 데이터 확보	-Selenium 기반 웹 FAQ 크롤링 - requests 기반 정적 HTML 수집 - PDF 문서 수집	Selenium, WebDriver, requests, BeautifulSoup
데이터 전처리	비정형 데이터를 구조화된 텍스트로 변환	- HTML에서 FAQ 질의/응답 추출 - PDF 다단 레이아웃 분리(1~4단) - 페이지·섹션 단위 텍스트 추출	BeautifulSoup, pdfplumber
정규화	노이즈 제거 및 텍스트 품질 개선	- 특수문자, 불필요 문구 제거 - 개행/탭/중복 문장 정리 - OCR 오타·URL 공백·반복 블록 제거	re
데이터 청킹	RAG에 적합한 입력 단위 생성	- 문장/의미 단위 청킹 - 길이 제한 기반 분할 - 중복 청크 제거	custom chunking logic (Python)
데이터 형식 통일	최종 벡터 DB 적재용 JSON 생성	- 서로 다른 소스(JSON/FAQ/PDF) 통합 - id / text / metadata 스키마로 변환 - 최종 벡터 적재용 JSON 생성	custom Python script