# 데이터 수집 보고서

제출일자: 2025.02.12. (수)

작성 팀원 : 김요은, 장정호, 김혜서

#### ▶ 프로젝트 개요

- 프로젝트 주제 : 카메라 사용자 매뉴얼 검색 시스템
- 프로젝트 목표
- 1) 고객상담 업무 효율성을 극대화하기 위해 구매고객을 대상으로 서비스되고 있는 문서의 검색 시스템을 질의 응답 형식으로 구축
- 2) 카메라 사용자의 카메라에 대한 정보 접근성 확대 및 초기 카메라 문제 발생 시 간편 해결 방안 제시로 편의성 증대

#### ▶ 데이터 수집 및 전처리 목적

- 수집 데이터 : 카메라 사용자 메뉴얼
  - 1) 각 브랜드 별 홈페이지에 업로드 되어있는 카메라 사용 메뉴얼 데이터
  - 2) 활용 브랜드: 캐논, 소니, 후지 필름
  - 3) 활용 모델

    - b) 소니 : ILCE-6400 α6400, ILCE-7M3 α7III, DSC-RX100M7, ZV-1, ZV-E10
    - c) 후지: GFX100II, X-E4, X-S20, X-T5, X100V
- 데이터 수집 목적 : 카메라 사용자 메뉴얼 내에서 검색할 수 있는 모델을 구현하기 위한 수집
- 데이터 전처리 목적 : 이미지, 표, 텍스트가 혼합되어있는 PDF 파일을 모델이 이해하고,

비슷한 방식으로 답변을 생성할 수 있도록 하기 위해

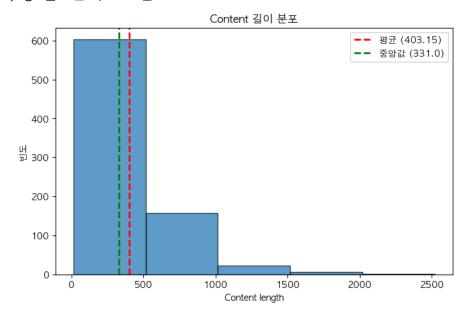
이미지의 위치, 표의 텍스트 등까지 처리할 수 있게 전처리.

### ▶ 데이터 수집 및 전처리 과정

- 데이터 수집 과정 : 활용 브랜드 별 홈페이지에 접속하여 사용자 메뉴얼 PDF 다운로드
- 데이터 전처리 과정
  - 1) 텍스트 파싱
    - a) 사용 Parser: LLama Parser, Pymupdf
    - b) 문제점 : 텍스트와 함께 혼용되고 있는 이모티콘의 제대로 된 파싱이 진행되지 않아 텍스트의 맥락을 잃어버리는 경우가 발생함.
    - c) 해결 방안 : LLama Parser 에서 MultiModal 모드를 활용하여 LLM을 통한 파싱 데이터를 받아옴.
  - 2) 이미지 파싱
    - a) 사용 Parser: LLama Parser, Upstage Parser
    - b) 텍스트로 설명하지 않고 이미지를 통해 설명되어있는 데이터가 다수 있어 이미지에 대한 설명 혹은 이미지를 LLM 모델에 함께 전달해야 함.
    - c) 해결 방안 : LLama Parser 를 통해 파싱 데이터에 이미지 위치를 함께 전달, Upstage Parser 를 통해 PDF 내에서 이미지를 추출하고 크롭하여 저장
  - 3) 메타데이터 생성
    - a) 각 브랜드 노드별 여러 모델의 PDF 를 사용하기 때문에, 데이터의 필터링을 위한 브랜드, 모델명 필요
    - b) PDF 1개의 페이지 수가 적으면 300장, 많으면 1000장이 넘어가고, 이미지를 함께 처리해야 하기 위한 페이지 수 필요
    - c) 메뉴얼 파일이기 때문에 존재하는 인덱스 페이지 제거 및 효율적인 인덱스 작업을 위한 메타데이터 추가 필요
    - d) 진행 방안 : 파싱 진행 시 메타데이터 내 PDF 모델명, 브랜드명, 페이지 수 추가, 파싱된 인덱스 페이지를 바탕으로 챕터 혹은 메인 인덱스 형식으로 메타 데이터 추가

### ▶ 데이터 수집 및 전처리 과정

- 4) 청킹
  - a) 파싱된 데이터 분할
  - b) 목적: LLM 모델 구현 시, 모델에 들어가는 토큰 개수를 효율적으로 활용하기 위함.
  - c) 진행 과정: 각 브랜드 모델 별 다른 방식으로 구현
    - 캐논: RecursiveCharacterTextSplitter 진행(청크 사이즈 1000, 오버랩 500)
      - 페이지별 청크 개수 : 평균 403.15(최대 2519, 최소 15)
      - 주의사항과 같은 페이지는 청크가 큰 경우가 있어 효율적인 RAG 운영을 위한 split 진행
      - page 별 토큰 히스토그램



- 1000 토큰을 넘는 경우가 거의 없으므로, 1000청크를 기준으로 청킹 진행
- 소니: 파싱된 json파일 데이터 문서를 기반으로 텍스트 청킹을 진행
  - page 단위로 데이터를 읽음
  - 청킹 함수(chunk\_text)를 사용하여 max\_length=512 최대 길이 지정
  - 문장 잘림을 방지하여 overlap=100 지정 (슬라이딩 윈도우 방식)

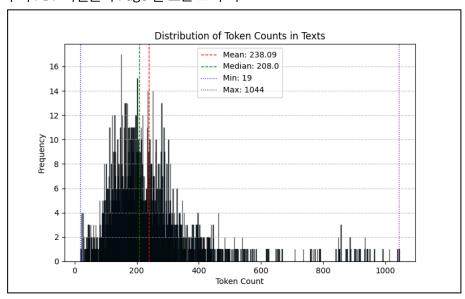
### ▶ 데이터 수집 및 전처리 과정

#### - 후지

- GPT-4o의 Context Window 제한 : 128,000 tokens,

Output Token 제한: 16,384 tokens

- 후지 PDF 파일들의 Page 별 토큰 크기 비교



- 평균(Mean): 238 tokens, 중앙값(Median): 208 tokens
- 즉, 일반적인 토큰 크기는 약 200~250 tokens
- 입력 예상 토큰 수 (3장 입력 시 약 700 tokens) 는 GPT-4o의 Context Window(128,000 tokens) 대비 매우 적은 수준이기 때문에 chunking 없이도 token 제한에 문제가 없을 것으로 판단되어 페이지 별로 청킹된 데이터 사용

# ▶ 데이터 전처리 결과

- 각 RAG 모델 별 파싱 및 전처리 결과
- 캐논

원본	Llama Parser	Upstage Parser
촬영 자세	## 셔터 버튼	
스크린을 보면서 촬영하기 항영하면서 스크린을 경제하기 보장하는 기본 사용하기를 참조하는 기본 사용하기를 참조하는 기본 사용하기 보조하는 기본 사용하기를 참조하는 기본 사용하기를 참조하는 기본 사용하기를 참조하는 기본 사용하기를 참조하는 기본 사용하기를 하고 있다.  (1) (2) (3)	셔터 버튼은 2단계로 되어 있습니다. 셔터 버튼을 반누름한 다음 추가적으로 완전히 누를 수 있습니다.	(1) (2) (3) (1) 일반 앵글
() 일반 영균 () 로우 영균 (3) 하이 영균 ● 부파인다를 보면서 촬영하기	### 반누름	(2) 로우 앵글 (3) 하이 앵글
선명한 이미지를 얻으려면 가다보를 만했다고 같아 카데라 온통함을 최소화력이 합니다.	![그림 자리(셔터 버튼 반누름 설명)]	
(1) 기도 함께 (2) 기도 하는으로 본프로 아이에서까지 (3) 무슨은 이 기로 시작 보이는 이 기로	자동 초점 및 셔터 스피드와 조리개 값을 설정하는 자동 노출 시스템이 실행됩니다. 노출값(셔터 스피드와 조리개 값)이 스크린이나 뷰파인더에 약 8초간 표시됩니다(측광 타이머).	(1) (2) (1) 가로 촬영
	### 완전 누름	
	![그림 자리(셔터 버튼 완전 누름 설명)]	
	셔터를 개방시켜 사진을 촬영합니다.	
	### 카메라 흔들림 방지하기	
	카메라를 손에 들고 있을 때 노출 순간에 일어나는 카메라의 움직임을 카메라 흔들림이라고 합니다. 카메라 흔들림이 발생하면 이미지가 흐릿해질 수 있습니다. 카메라 흔들림을 방지하려면 아래의 사항에 유의하십시오:	
	- **촬영 자세**의 설명을 따라 카메라를 안정적으로 잡으십시오. - 먼저 셔터 버튼을 반누름하여 자동으로 초점을 맞춘 다음 버튼을 천천히 끝까지 누르십시오.	

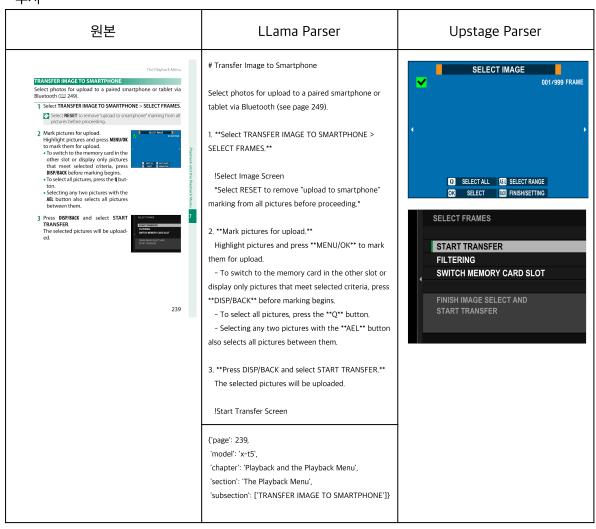
# ▶ 데이터 전처리 결과

#### - 소니



# ▶ 데이터 전처리 결과

## - 후지



#### 데이터 수집 및 저장

# ▶ 데이터 저장 및 관리

- 데이터 저장 장소 : Pinecone Index, BM25 Retriever 객체
- 각 데이터를 임베딩 모델을 활용하여 저장
- 사용하는 각 브랜드별로 DB를 따로 저장하여 관리함.
- 단, Vectorstore 의 경우 API 기반 서버리스 벡터스토어에 저장함.
- 캐논
  - 1) VectorStore: OpenAi embedding 활용하여 저장(사용 모델: text-embedding-3-small)
  - 2) BM25 Retriever: BM25 tokenizer kiwi 사용하여 형태소 분할 및 retriever 객체로 저장
- 소니
  - 1) VectorStore: OpenAi embedding 활용하여 저장(사용 모델: text-embedding-3-small)
  - 2) BM25 Retriever : BM25 tokenizer kiwi 사용하여 형태소 분할(토큰화) 및 retriever 객체로 저장

#### - 후지

- 1) VectorStore : OpenAi embedding 활용하여 저장(사용 모델 : text-embedding-3-small)
- 2) BM25 Retriever: porter\_stemmer 사용하여 어간 추출(Stemming)및 retriever 객체로 저장