**SK네트웍스 Family AI 과정 13기  
 데이터 전처리 인공지능 학습 결과서**

|  |  |
| --- | --- |
| **산출물 단계** | 데이터 전처리 |
| **평가 산출물** | 인공지능 학습 결과서 |
| **제출 일자** | 2025.08.10 |
| **깃허브 경로** | [SKNETWORKS-FAMILY-AICAMP/SKN13-FINAL-6Team](https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN13-FINAL-6Team) |
| **작성 팀원** | 지형우, 김동욱, 안수민 |

**1. 모델 비교 및 선정 이유**

① 임베딩 모델

|  |  |  |
| --- | --- | --- |
| **모델명** | **종류** | **선정 이유** |
| text-embedding-3-large | openai 임베딩 모델 | 범용성이 넓으며 성능이 우수한 모델이라 가장 먼저 시도. 그러나 유료 API이기 때문에 비용이 부담되며, 3072차원이므로 저장 및 검색 비용이 증가함. |
| intfloat/multilingual-e5-large | 임베딩 모델 | text-embedding-3-large의 비용적인 면을 보완하기 위해 선정한 모델. 비용 최적화가 가능하고 다국어를 지원. 또한, 1024차원으로 저장 및 검색 비용 감소.  한계: 한국어 일반 문서에는 성능이 준수하지만, 한국어 도메인 특화 텍스트(사내 문서)에는 매칭력이 아쉽다고 판단함. |
| nlpai-lab/KoE5 | 임베딩 모델 | intfloat/multilingual-e5-large의 한계를 보완하기 위해 선정. 임베딩할 문서는 한국어로 구성되어 있으므로, 한국어 특화 파인튜닝된 해당 모델이 가장 알맞을 것이라 판단. 성능 및 비용 균형이 우수함. |

② 답변 생성용 LLM 모델

|  |  |  |
| --- | --- | --- |
| **모델명** | **종류** | **선정 이유** |
| Gpt-4o | LLM 모델 | 해당 모델을 사용해 답변을 생성했을 때 정확도 및 속도 면에서 성능이 우수했음. 상용 모델 중 정확도와 토큰별 가격을 고려했을 때 적합하다고 판단. |

③ RAG 모델

|  |  |  |
| --- | --- | --- |
| **모델명** | **종류** | **선정/배제 이유** |
| Qwen2-VL-7B-Instruct-GPTQ-Int8 | 멀티모달 대형 언어 모델 | 영수증 필드를 잘 인식하지만, 일부 한글 인식률이 부족하고 JSON 형식 불일치 빈번 |
| A.X-4.0-VL-Light | OCR·비전-언어 통합 모델 | 출력 품질은 일정하나 처리 속도가 매우 느려 대량 처리 환경에 부적합 |
| Kanana-1.5-V-3B-Instruct | 멀티모달 대형 언어 모델 | 결제처/결제일시/카드사/품목/총합계 등 주요 필드를 안정적으로 추출, JSON 형식 일관성 우수 |

**2. 모델 구조 및 아키텍처**

**2-1. 모델 아키텍쳐**

① RAG 모델

* 입력 → RAG 문서 검색 → LLM 답변 생성

② OCR 모델

* 입력층: 영수증 이미지 + 텍스트 프롬프트
* 비전 인코더: 이미지 특징 추출 (Visual Transformer)
* 텍스트 인코더/디코더: Transformer 기반 멀티모달 통합 처리
* 출력층: 구조화된 JSON 응답

**2-2. 구성 요소 설명:**

① RAG 모델

|  |  |  |
| --- | --- | --- |
| **계층명** | **역할** | **구성 요소** |
| 문서 검색(Retriever) | 질의와 유사한  문서 검색 | Qdrant, nlpai-lab/KoE5 |
| 응답 생성(Response) | 질의와 검색된 문서를 받아 답변 생성 | Gpt-4o |

② OCR 모델

|  |  |  |
| --- | --- | --- |
| **계층명** | **역할** | **구성 요소** |
| Vision Encoder | 이미지에서 특징 추출 | Swin Transformer 기반 비전 백본 |
| Text Encoder/Decoder | 멀티모달 의미 통합 및 생성 | Multi-Head Attention  + Feed Forward Network |
| Generation Head | JSON 포맷 생성 | Auto-regressive  Transformer Decoder |

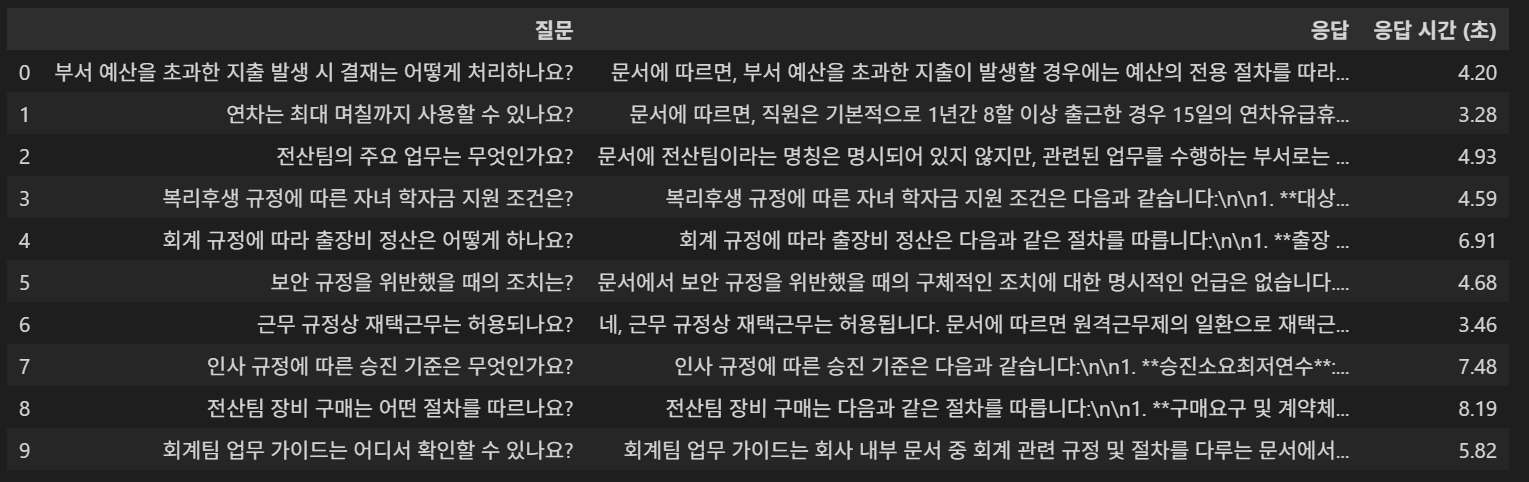
**3. 학습 설정 및 하이퍼파라미터**

모델을 추가로 학습하지 않음.

**4. 성능 평가  
4.1 성능 평가**

① RAG 모델

구성한 RAG 체인의 성능을 평가하기 위해 예시 질문 10개에 대한 응답과 응답 생성 소요 시간을 측정한 결과 아래와 같다.



사내 내규 문서와 응답의 내용이 동일했으며, 사내 내규 문서에서 답변에 참고할 내용을 찾을 수 없는 경우 정확히 답변할 수 없다고 응답하게 해 **할루시네이션을 방지**했다. 또한 **응답 시간**이 10개 질문 **모두 10초 이내**로 나와 속도 면에서 성능이 괜찮다고 판단했다.

② OCR 모델

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **모델** | **결제처**  **정확도** | **품명**  **정확도** | **금액**  **정확도** | **평균 처리 속도(1장)** |
| Qwen2-VL-7B-Instruct-GPTQ-Int8 | 95% | 80% | 99% | 48초 |
| A.X-4.0-VL-Light | **97%** | **98%** | **100%** | 300초 |
| Kanana-1.5-V-3B | 85% | 95% | 95% | **5.5초** |

* Kanana : 속도가 가장 빠르지만 결제처를 탐색하는 능력과 금액 정확도가 부족.
* Qwen2-VL : 품목명을 추출해내는 데 어려움 있음.
* A.X-4.0-VL-Light : 정확도는 높지만 처리 시간이 길어 서비스 배포에 부적합.

### **4.2 성능 최적화 방안 :** A.X-4.0-VL-Light 모델의 속도 개선을 위한 양자화 적용

A.X-4.0-VL-Light 모델이 성능이 가장 우수하지만 속도가 느리다는 문제가 있어 분석한 결과, FP32(32비트 부동소수점)를 사용하여 메모리 및 연산량이 과도하게 소요된다는 점을 발견했다.

* 기존 문제점: FP32를 사용하여 메모리 및 연산량이 과도하게 소요되었다.
* 개선 방법: FP16 양자화 적용, 연산 정밀도를 조절하여 메모리 사용량과 처리 속도를 최적화하였다. (평균 처리 속도 : 43초)

**5. 결론 및 향후 계획**① RAG 모델

* **최종 선정 모델**: 임베딩: nlpai-lab/KoE5, 답변 생성: Gpt-4o
* **활용 방안**: 문서 임베딩, 질의와 검색된 문서를 토대로 답변 생성
* **향후 계획**:
  + 임베딩 시 내규 문서의 부칙 제거 고려 (이미지 및 표 데이터)
  + 임베딩 시 Qdrant DB에 저장한 메타데이터를 어떻게 활용할 수 있을지 고려

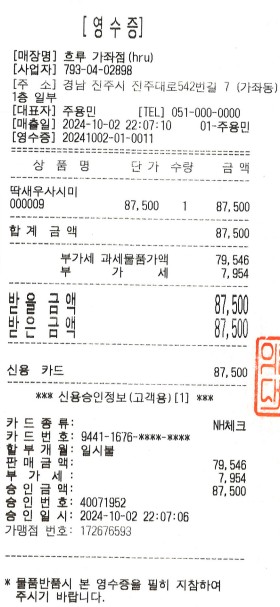
② OCR 모델

* **최종 선정 모델**: A.X-4.0-VL-Light
* **활용 방안**:
  + 대량 영수증 OCR 및 데이터베이스 자동 저장
  + 기간별 필터링 후 CSV 출력 서비스 구축
* **향후 계획**:
  + 프롬프트 수정, 데이터셋 기반 Fine-tuning 등 추가 개선 방안 검토

**7. 부록**

테스트 사진 (예시)

* 원본 영수증



* 출력 결과

|  |
| --- |
| QWEN |
| A.X |
| Kanana |