HWP 데이터 학습을 활용한 LLM 기반 RAG 시스템 설계 방안에 관한 연구 –수식 파운데이션 및 이미지 분류를 중심으로

정의현, 차민수, 한승민, 정진우, 윤수연*
국민대학교

rabbitdy0101@kookmin.ac.kr, minns00@kookmin.ac.kr, handsomemin@kookmin.ac.kr, imaboybut@kookmin.ac.kr, *1104py@kookmin.ac.kr

A Study on the Design of an LLM-based RAG System Utilizing Preprocessed HWP Data – Focusing on Equation Preprocessing and Image Classification

Ui Hyun Jung, Min Su Cha, Seung Min Han, Jin Woo Chung, Soo Yeon Yoon*
Kookmin Univ.

요 약

본 연구에서는 한글 문서 내의 콘텐츠를 텍스트, 수식, 이미지, 표의 네 가지 카테고리로 분류하고, 이를 기반으로 정보를 구조적으로 추출하였다. 추출된 한글 수식 개체가 LaTeX 로 변환된 후에도 원본 구조를 잘 유지한 채 시각적 표현에서도 큰 손실 없이 재현되었다. 이미지 분류에 경우 라벨링 개선 전에는 그래프, 수식, 텍스트 각 카테고리에서 분류 정확도가 상대적으로 낮게 나타났으며, 특히 종합 분류 성능이 0.4/0.32 에 머무르는 등 멀티모달 데이터 처리의 한계가 명확하게 드러났다. 그러나 라벨링 개선 이후, 그래프 분류는 0.91/0.952, 수식 분류는 0.94/0.97, 텍스트 분류는 0.885/0.938 로 각각 크게 향상되었으며, 전체 분류 성능 역시 0.99/0.98 로 극적인 개선을 보였다. 이러한 결과는 구조적 일관성 확보와 도메인별 특징 강화가 모델의 분류 능력 향상에 결정적으로 기여했음을 시사한다.

I. 서론

아래아한글(HWP)은 (주)한글과 컴퓨터에서 개발된 워드 프로세서로, 2024 년 기준 국내에서 30% 정도의 시장 점유율을 보이며 교육 기관 및 공공기관에서 문서 작성에 널리 사용되고 있다. 이처럼 국내에서 많은 문서들이 아래아한글 형식으로 작성되고 있음에도 불구하고, 이를 LLM(대규모 언어 모델) 기반 RAG 시스템 설계에 직접적으로 활용하는 데 어려움이 존재한다. 이에 따라 본 연구에서 한글 문서를 LLM 에 활용하기 위해 한글 문서의 데이터를 추출하여 전처리한 뒤, LLM 이 학습할 수 있는 JSON 형태로 저장하는 작업을 수행하였다.

II. 본론

2.1 한글 문서(HWP) 전처리

본 연구에서는 한글 문서 내의 콘텐츠를 텍스트, 수식, 이미지, 표의 네 가지 카테고리로 분류하고, 이를 기반으로 정보를 구조적으로 추출하였다. pyhwpx 라이브러리를 이용하여 HwpCtrl Object 를 통해 각 요소의 데이터를 수집하였으며, 추출된 결과는 후속 처리를 거쳐 JSON 형식으로 저장하였다. 이 중에서도 수식과 이미지 개체는 LLM 학습 과정에서의 정보 손실을 최소화하는 것을 목표로 하여, 해당 요소들의 전처리 방식에 중점을 두고 실험을 수행하였다.

2.2 수식 추출 및 LaTeX 변환

2.2.1 한글(HWP) 수식 개체

한글 문서의 수식 개체는 수식 편집기를 통해 삽입되며, 시각적으로 구성된 템플릿 방식과 스크립트 입력 방식을 모두 지원한다. 수식 개체는 문서 내 텍스트처럼 보이지만, 내부적으로는 독립된 개체로 분류되며, 위치 이동이나 크기 조절이 가능한 별도의 삽입 요소로 관리된다. 이처럼 수식 개체는 일반 텍스트와는 다르게 저장 및 처리되기 때문에, 수식을 추출하거나 변환하려면 별도의 접근 방식이 요구된다.

2.2.2 수식 추출 및 LaTeX 변환 실험

대학 수학 및 공학 분야의 대학강의 자료 110 개 한글 파일에 대하여 10,684 개의 수식 개체의 추출을 수행하였다. 추출된 수식을 Mathml 로 저장하여 py-asciimath 를 통해 LaTeX 형식으로 변환하였다.

pyhwpx 의 함수는 수식 개체를 하나씩 순차적으로 처리하는 구조를 가지므로, 다수의 수식을 처리할 경우 전체 수행 시간이 과도하게 증가하는 문제가 발생한다. 이를 해결하기 위해 본 실험에서는 다수의 수식 스크립트 문자열을 병합하여 하나의 긴 수식 개체로 삽입한 후, 이를 다시 추출하여 변환하는 방식을 고안하여 적용하였다.

또한 py-asciimath 를 통한 변환 과정에서 반복적으로 발생하는 일부 표현 오류에 대해 매핑 테이블을 정의하여 후처리를 수행하였다.

2.2.3 수식 추출 및 LaTeX 변환 성능 분석

본 실험에서는 변환된 LaTeX 수식이 원본 수식과 시각적으로 구조적 일치를 이루는지를 기준으로 정성적 평가를 수행하였다. 변환된 LaTeX 수식은 [표 2]와 같이 제시하였으며, 이를 렌더링한 결과는 [표 3]에서 확인할 수 있다. 또한 처리 방식별 처리시간은 [표 4]에 제시하였다.

[표 1] HWP 수식 스크립트와 변환된 LaTeX 비교

한글 수식 스크립트 문자열
e=e ^{1} =e ^{lim _{x rarrow 0} {rm ln} (1+x) ^{1/x}} = lim _{x rarrow 0} {} e ^{ln(1+x) ^{1/x}} = lim _{x rarrow 0} {} (1+x) ^{1/x}
변환된 LaTeX 수식
e={e}^{1} ={e}^{\underset{x\to0} {lim}\text{ln} (1+x{})^{1/x}} =\underset{x\to0} {lim} {e}^{\text{ln} (1+x{})^{1/x}} =\underset{x\to 0} {lim} (1+x{})^{1/x}

[표 2] HWP 수식 개체와 변환된 LaTeX 렌더링 결과 비교

한글 수식 개체
$e = e^1 = e^{\lim_{x\to0}\ln(1+x)^{1/x}} = \lim_{x\to0} e^{\ln(1+x)^{1/x}} = \lim_{x\to0}(1+x)^{1/x}$
변환된 LaTeX 수식 렌더링 결과
$e = e^1 = e^{lim_{x\to0}\ln(1+x)^{1/x}} = lim_{x\to0} e^{\ln(1+x)^{1/x}} = lim_{x\to0}(1+x)^{1/x}$

[표 1]와 [표 2]에서 확인할 수 있듯, 한글 수식이 LaTeX 로 변환된 후에도 원본 구조를 잘 유지한 채 시각적 표현에서도 큰 손실 없이 재현되었음을 확인할 수 있었다.

[표 3] 수식 처리 방식 별 처리시간

수식 수	수식 개별 처리 시간	수식 병합 처리 시간
62	139.7 초	8.2 초
198	439.8 초	13.4 초
465	1056.4 초	40.7 초

[표 3]에서 확인할 수 있듯, 본 연구에서 고안한 수식 병합 처리 방식은 동일한 수의 수식들을 훨씬 짧은 시간 내에 처리할 수 있으며, 이는 개별 수식 처리 방식 대비 처리 효율에서 현저한 차이를 보인다.

2.3 이미지 분류

2.3.1 이미지 분류 실험 및 결과

본 실험에서 이미지 분류 성능을 확인하기 위해서 허깅페이스에서 각 타입 당 200 개의 데이터를 사용하였다.

[표 4] 사용한 데이터셋

데이터 타입	데이터 명
Text	naver-clova-ix/synthdog-ko
Graph	jp1924/ChartLlamaDataset
Formula	Nagase-Kotono/img-latex-ko

본 연구에서는 수식과 그래프가 포함된 멀티모달 데이터 분류 성능 향상을 위해 라벨링의 개선을 수행하였으며, 라벨링 기법이 분류 정확도에 미치는 영향을 Accuracy 와 F1 Score 를 통해 정량적으로 분석하였다. 실험은 SigLIP2 모델을 기준으로 하여 텍스트, 그래프, 수식 카테고리에 해당하는 여러 라벨을 통해 분류를 진행하는 방식으로 진행되었다.

수식의 분수나 화학식의 형태같이 각 분류 도메인의 특징적인 부분을 기준으로 라벨을 개선하였으며 동일 카테고리 내에 여러 개의 라벨을 담았다.

2.3.2 이미지 분류 성능 평가 분석

실험 결과, 라벨링 개선 전에는 그래프, 수식, 텍스트 각 카테고리에서 분류 정확도가 상대적으로 낮게 나타났으며, 특히 종합 분류 성능이 0.4/0.32 에 머무르는 등 멀티모달 데이터 처리의 한계가 명확하게 드러났다. 그러나 라벨링 개선 이후, 그래프 분류는 0.91/0.952, 수식 분류는 0.94/0.97, 텍스트 분류는 0.885/0.938 로 각각 크게 향상되었으며, 전체 분류 성능 역시 0.99/0.98 로 극적인 개선을 보였다. 이러한 결과는 구조적 일관성 확보와 도메인별 특징 강화가 모델의 분류 능력 향상에 결정적으로 기여했음을 시사한다.

[표 5] 라벨링 전과 후 성능 비교

형태	라벨링 전	라벨링 후
텍스트	0.51	0.91
	0.67	0.952
그래프	0.76	0.94
	0.86	0.938
수식	0.59	0.94
	0.74	0.97
종합	0.4	0.99
	0.32	0.98

특히, 동일한 크기의 토큰 단위로 라벨을 표준화함으로써 모델이 다양한 데이터 형식을 균일하게 처리할 수 있게 되었고, 이는 분류 모델의 효율성을 크게 높이는 결과로 이어졌다. 또한, 한국어 텍스트 분류의 경우 기존에 발생하던 판별 오류가 크게 줄어들어, 한국어 멀티모달 데이터 처리의 실용적 가능성도 확인할 수 있었다.

III. 결론 및 향후 연구

본 연구는 아래아한글(HWP) 문서를 LLM 기반 RAG 시스템에 활용하기 위해 한글 문서의 데이터를 추출 및 전처리하여 JSON 포맷으로 변환하는 시스템을 설계하였다. 특히, 수식 파운데이션 및 이미지 분류에 집중하여 한글 문서 내 수식 및 이미지 개체의 데이터 손실을 최소화하는 전처리 방식을 고안하였고, 실험을 통해 한글 문서의 활용성을 넓히는 가능성을 확인하였다.

향후 연구에서는 한글 이외의 다국어 이미지 분류, 대용량 문서의 실시간 변환 최적화, 화질이 낮은 이미지의 OCR 정확도 향상 등 추가적인 기술 고도화를 통해 시스템의 범용성과 활용도를 더욱 높일 계획이다.

ACKNOWLEDGEMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학사업의 연구결과로 수행되었음. (2022-0-00964)

This research was supported by the Ministry of Science and ICT (MSIT) and the Institute for Information & Communications Technology Planning & Evaluation (IITP) under the National Program of Excellence in Software (2022-0-00964).

참 고 문 헌

[1] 정승범, 윤수연. "LMM 기반 한국어 문서 표 이미지 데이터 학습을 통한 전자문서 요약 생성 기법에 관한 연구." 한국통신학회 학술대회논문집 (2025): 740-741.

[2] Michael Tschannen, et al., "SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features," arXiv preprint arXiv:2502.14786, Feb. 2025.

[3] py-asciimath https://py-asciimath.readthedocs.io/en/latest/