석사학위논문

LLM으로 문서의 Layout을 인식하기 위한 Prompt 구성 전략에 대한 연구

- Text와 Bounding Box 정보를 활용한 In-Context Learning 과정에 대한 실험 -

고려대학교 컴퓨터정보통신대학원

인공지능융합학과

이찬

2024년 2월

임희석 교수지도

석사학위논문

LLM으로 문서의 Layout을 인식하기 위한 Prompt 구성 전략에 대한 연구

이 논문을 공학 석사학위 논문으로 제출함

2023년 10월

고려대학교 컴퓨터정보통신대학원

인공지능융합학과

이찬(인)

이찬의 석사학위논문 심사를 완료함

2023 년 12 월

위원장 임 희 석 (인)

위 원 김 현 철 (인)

위 원 강 재 우 (인)

LLM 으로 문서의 Layout 을 인식하기 위한 Prompt 구성 전략에 대한 연구

이 찬

인 공 지 능 융 합 학 과

지도교수: 임 희 석

초록 (국문)

본 연구는 Visually Rich Document Understanding(VRDU)의 하위 과제인 Key Information Extraction 을 Large Language Models(LLM)과 Prompt Engineering 기법을 활용하여 수행하는 방법을 탐구한다. 기존의 이미지 기반 사전 학습 모델을 사용하지 않고, LLM 만을 이용하여 문서의 Key-Value 쌍을 추출하는 가능성을 실험적으로 검증하였다. 이를 위해 OpenAI 의 ChatGPT 를 활용하여 문서의 Layout 을 인식하고, Key-Value 쌍을 추출하는 실험을 수행하였다. 실험 결과, LLM 이 In-Context Learning 을 통해 문서 인식 Task 에 적용될 수 있음을 확인하였으며, 특히 Text 와 Bounding Box 정보를 함께 활용할 때 추론 성능이 향상되는 것을 관찰하였다. 그러나 실무 활용을 위한 성능은 아직 충분하지 않았고, 향후 연구에서는 예시 정보를 늘려 성능 변화를 확인하고, 이미지를 입력으로 허용하는 GPT4 모델에서의 성능을 비교 분석할 예정이다.

중심어: LLM, In-Context Learning, Document AI, Prompting, FUNSD, Key-Value Extraction

A Study on prompt engineering strategies to recognize document layout with LLM

by Lee Chan

Department of Applied Artificial Intelligence

under the supervision of Professor Lim Heuiseok

ABSTRACT

This study explores the methodology of utilizing Large Language Models (LLMs) and Prompt Engineering techniques for Key Information Extraction, a subtask of Visually Rich Document Understanding (VRDU). The research experimentally verifies the potential of extracting Key-Value pairs from documents using only LLMs, without relying on pre-trained image-based models. Through experiments with OpenAI's ChatGPT, the effectiveness of In-Context Learning in recognizing document layouts and extracting Key-Value pairs was confirmed, suggesting the potential use of LLMs in building automated document processing systems. However, the current inferencing capabilities for practical application are recognized as insufficient, and future research will investigate performance changes with increased example information or with GPT4 models that allow images as input.

Keywords: LLM, In-Context Learning, Document AI, Prompting, FUNSD, Key-Value Extraction

졸업을 위해 논문 작성을 해야 한다는 이유로

임신 중인 아내보다 더 예민해진 제 성격을

옆에서 다 받아주고 인내해 준

나의 아내 김효정에게 감사의 인사를 남깁니다.

항상 저를 지지하고 응원해 주시는 부모님께

석사모를 씌워드릴 수 있어 감사하고

앞으로도 계속 자랑스러운 아들이 되겠습니다.

부모님 사랑합니다.

목차

표 목차

그림 목차

1 장. 서론

Visually Rich Document Understanding(이하 VRDU)이라 하여 Scanned 된 이미지나 PDF 형태로 존재하는 문서를 이해하기 위한 연구가 있다. 이 연구는 Document Classification, Layout Analysis, Information Extraction, Key-Value Extraction, Question Answering, Form Understanding 등 과 같이 특정 목적에 대해 범위를 좁힌 다양한 Sub Task 로 세분화할 수 있다. 이 연구들은 Visually Rich 라는 용어가 의미하듯 텍스트와 함께 문서의 이미지도 활용하는 Multi Modal 형태의 방법론을 활용한다. 실제로 각 벤치마크 데이터셋에 대해 높은 성능을 보이는 사전 학습 모델들은, 많은 양의 이미지를 학습에 함께 활용한 모델들이다. 하지만 Enterprise 환경에서는 이 모델을 바로 사용하지 못한다. 실무에서는 높은 정확도를 요구하기 때문에, 실제로 사용되는 문서에 대해서 미세조정 학습을 한 이후에 활용하게 된다. 하지만 매번 새로운 문서들이 나타나는 업종이라면 사전 학습 모델을 매번 재 학습시키며 활용하는 것이 현실적으로 어려운 부분이 많다.

Large Language Model (이하 LLM)이 등장한 이후 LLM 을 VRDU Task 에 활용하는 연구가 활발하다. LLM 이 특정한 Task 종속되지 않고 다양한 Task 에 대해서도 잘 동작하는 특성을 가지고 있고, LLM 에 Zero-Shot 혹은 Few-Shot 학습 형태로 필요한 예시를 제공하면 LLM 이 별도의 파라미터 변경 없이도 좋은 결과를 내는 In-Context 학습의 효과가 증명되면서 LLM 이 가지는 언어에 대한 이해 능력을 문서를 이해하는 Task 에 활용해 보려는 연구들이다.

방대한 양의 이미지를 활용한 사전 학습 모델을 사용하지 않고, LLM 만을 활용하여 문서를 이해하고 문서상에 존재하는 정보를 잘 추출할 수 있다면 실무에서 다양하게 활용할 수 있다. 본 연구에서는 그 방법과 가능성에 대해 확인해보고자 한다. LLM 을

VRDU Task 에 활용하여 좋은 성능을 낸 선행 연구들을 분석하고 이를 배경으로 실험을 하였다.

1.1 연구 배경

현대 비즈니스 업계에서 가장 큰 화두는 업무 자동화이다. 기계가 절대 사람을 대신할 수 없을 것이라고 생각한 분야에서도 AI 기술을 활용하여 자동화하는 사례가 늘어나고 있다. 업무 자동화 중에서도 문서 기반의 업무를 자동화하고자 하는 노력이 있는데, 그 전반을 IDP (Intelligent Document Processing)이라는 용어로 부른다. 이는 앞서 서론에서 언급한 문서 이해를 위한 다양한 Task 를 포괄하고 있다. Google, Amazon 과 같은 거대 회사에서는 IDP 혹은 Document Understanding 서비스를 API 형태로 제공하고 있다. API 의 종류에는 Invoice 나 모기지 관련 문서, 면허증과 같이 대중화된 문서에 대해 사전 학습된 모델로 데이터를 추출해 주거나, 필기체 서명과 Table 표를 인식하고, 나아가 문서에 존재하는 Key-Value 쌍을 추출해 주는 서비스가 있다. 또한 자연어 Query 기반의 Question-Answering 서비스도 존재한다.

사실 AI 기술이 발전하기 이전에도 문서 처리를 기계를 통해 자동화하려는 노력은 오래전부터 있어왔다. 특히 OCR (Optical Character Recognition) 기술을 활용한 노력이 대표적이다. OCR 기술은 매우 발달하여 다양한 언어에 대해서 잘 동작하며, 인쇄체뿐만 아니라 필기체에 대해서도 이미 높은 수준으로 동작하고 있다. OCR 은 우리 실 생활에서도 자주 살펴볼 수 있는데, 주차장 입구에서 자동차 번호판 인식을 하거나, 주민등록번호에서 필요한 값을 읽어내는 것이 그 대표 활용 사례이다. 문서 처리에 OCR 을 활용하는 것은 주로 다음과 같은 방식으로 동작한다. OCR 기술로 인식한 텍스트와 그 평면 상의 위치를 사전에 미리 정의한 Rule 혹은 Template 과 같은 규칙과 함께 비교하여 문서의 정보를

추출하는 것이다. 하지만 이 기술은 단순히 텍스트와 그 위치만 활용할 뿐 문서에 대해 이해를 하고 있다고 할 수 없다. Table 표와 같은 구조를 이해하지 못하고, Title, Description, Question, Answer 와 같은 Entity 를 구별하지 못하며, 각 텍스트 간의 의미적 관계를 발견하지 못한다. 앞서 언급한 Rule 과 Template 을 작성하면 해결할 수 있으나, 세상에는 너무 많은 종류의 문서가 존재하기 때문에 효율성 측면에서도 한계가 있다. AI 기술이 발전하면서 문서를 이해할 수 있는 기술이 나타났다. 앞서 언급한 Google, Amazon 의 유료 서비스들도 이러한 기술을 상용화한 것이다.

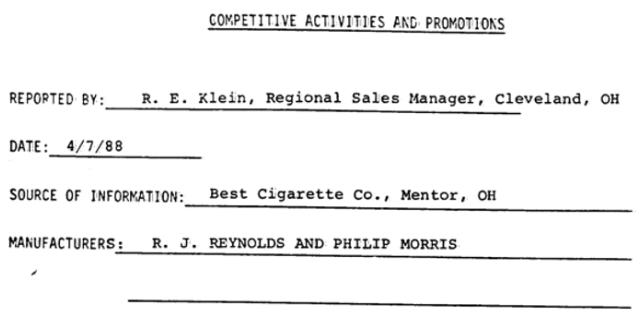
물류는 문서 처리 자동화를 도입했을 때 큰 효용을 얻을 수 있는 업종이다. 대부분의 물류 프로세스가 실물 종이 문서 기반으로 이루어진다 업종 특성 때문이다. 국제 무역에서의 물류 업계는 화주, 금융기관, 선사, 항공사, 내륙운송사, 세관, 항만 등 다양한 이해관계자와 참여자들이 존재하고, 또한 국가별로 문화, 법, 언어가 다른 특성이 있기 때문에 애초부터 모두가 하나의 통일된 시스템과 규칙 하에서 동작하기 어려웠다. 국제 무역에서 사용되는 선적 서류만 해도 운송장, 선하 증권, 상업 송장, 적하목록, 원산지 증명서 등 다양한데, 이를 발행하는 주체 또한 모두 다르다. 또한 각 문서의 형태가 비록 담고 있는 정보는 유사해도 업체 별로 그 양식이 다르다. 각 문서 별 존재하는 정보를 추출한다 하더라도, 정상적인 국제 무역 프로세스를 진행하려면 모든 문서의 정보를 취합하여야 비로소 가능하다. 이러한 이유 때문에 통합 시스템 기반의 자동화 구축이 어렵고 결국 사람이 직접 문서를 눈으로 보고 정보를 찾는 형태로 업무 운영을 진행하게 되는 케이스가 많다. 이러한 어려움 때문에 물류 업계에 문서 처리 자동화 시스템이 잘 적용된다면, 그 활용도나 효용 측면에서 큰 이익을 기대할 수 있다.

문서 처리 자동화를 위해 Google 과 Amazon 등에서 제공하는 IDP 상용 서비스를 사용하는 것은 사용자가 문서 인식을 위한 시스템에 대한 고민 없이도 높은 수준의 성능을 내는 모델을 쓸 수 있다는 점에서 합리적인 접근 방식이다. 하지만 상용 모델의 소유 주체가

외부에 있기 때문에, 시간이 지남에 따라 사용하던 해당 모델의 버전이 변경되면서 기존에 적용된 시스템에 영속성에 영향을 미칠 수 있고, Down Stream Task 에 활용하기에는 어렵다. 또한 상용 서비스는 Task Specific 하게 제공되기 때문에 활용도 측면에서도 단점이 존재한다.

1.2 연구 목적

본 연구의 목적은, 별도의 사전학습 모델 구축이나, 기존에 존재하는 문서 인식을 위한 사전 학습 모델의 미세 조정 학습 없이 LLM 과 Prompt Engineering 기법만을 이용하여 문서에 존재하는 Key-Value 쌍을 찾아내는 것이다. Key Information Extraction 이라고도 불리는 이 Task 는 문서에서 의미적 연관성을 가진 데이터들을 뽑아내는 것이기 때문에, 문서 처리 자동화 시스템을 구축하는 데 있어서 매우 요긴하게 활용될 수 있다. 사용자는 Key-Value 쌍을 찾아낸 이후에 이 데이터를 활용하는 후처리 업무 로직들을 개발하여 다양한 Task 에 활용할 수 있다.



COMPETITIVE ACTIVITIES AND PROMOTIONS

REPORTED BY: R. E. Klein, Regional Sales Manager, Cleveland, OH

DATE: 4/7/88

SOURCE OF INFORMATION: Best Cigarette Co., Mentor, OH

MANUFACTURERS: R. J. REYNOLDS AND PHILIP MORRIS

그림 1 문서 예시 (FUNSD - 9345715.png 일부)

표 1. Key-Value 예시

Key (Question)	Value (Answer)
REPORTED BY:	R. E. Klein, Regional Sales Manager, Cleveland, OH
DATE:	4/7/88
SOURCE OF INFORMATION:	Best Cigarette Co., Mentor, OH
MANUFACTURERS:	R. J. REYNOLDS AND PHILIP MORRIS
생략	생략

1.3 논문의 구성

2 장에서는 문서 인식을 위한 기존 선행 연구들을 시간 순에 따라 발전 과정과 각 특징에 대해 분석하고, 이어 최근 대두되고 있는 LLM 과 In-Context Learning 에 대해 살펴보고자 한다. 3 장에서는 LLM 을 활용하여 문서 인식을 하고자 한 선행 연구에 대해 확인한 내용을 설명한다. 이어 4 장에서는 본 연구에서 수행하는 실험에 대한 내용을 설명한다. 사용한 데이터셋에 대한 소개와, OpenAI 의 ChatGPT 를 활용하여 실제로 수행해 본 실험 내용에 대하여 기술하고, 본 연구에서 하고자 했던 In-Context Learning 기법을 통한 성능 변화가 실제로 어떻게 나타나는지 살펴본다. 이어 5 장에서는 본 실험을 통해 도출된 결과에 대해 정리하고 실무에서의 활용 방안에 대해 고찰해보고자 한다.

2 장. 배경 지식 및 기존 연구

2.1 문서 인식 연구

격년으로 전 세계에서 개최되는 ICDAR(International Conference on Document Analysis and Recognition)라는 학회가 있다. 이 학회는 학회의 명칭에서 알 수 있는 것처럼 문서 분석과 인식 분야에서 주요한 국제 학회이다. 텍스트 및 이미지 문서 처리, 인쇄/필기체 인식, 문서 분석, 문서 이해, 고문서 디지털화 등 문서에 대한 연구에 중요한 역할을 하고 있다. 이 학회에서는 RRC(Robust Reading Competition)를 개최하는데, Challenges 의 목록을 살펴보면 문서 인식을 위한 최근 연구 동향에 대해 알 수 있다.

표 2. RRC 주요 Challenges Key-Value 예시

Challenge	Description	개최 연도
DocVQA	Document Visual Question Answering	2020-23
SVRD	Structured Text Extraction from Visually-Rich Document Images	2023
DUDE	Document UnderstanDing of Everything	2023
DocILE	Document Information Localization and Extraction Key Information Localization and Extraction (KILE) and Line Item Recognition (LIR)	2023
ST-VQA	Scene Text Visual Question Answering	2019
SROIE	Scanned Receipts OCR and Information Extraction	2019

위 표에서 알 수 있는 것처럼 최근의 주요 연구 Task 는 단순히 문서의 글자를 인식하는 수준을 넘어서 문서의 Visual 정보를 활용한 Question Answering, Key Information Extraction 등으로, 문서의 Layout 을 이해하고 의미적 관계를 찾아내는 것을 주요 내용으로 한다.

문서 인식을 위한 연구는 주로 문서 위 텍스트가 위치한 영역을 감지하고 그 영역에 포함된 글자를 인식하는 형태의 광학 문자 인식(OCR) 기술이 그 시초였다. 이어 이렇게 인식된 Plain 텍스트 기반으로 각 단어의 Entity 를 구별하거나 주요 정보를 찾는 신경망

기반의 연구들이 나타났다. 주로 RNN 계열의 모델이 대표적이다. 하지만 문서는 2 차원으로 이루어져 있고, 문서를 인식하는 데 있어서 이미지가 가지는 Visual 정보도 중요하기 때문에 CNN, GNN 과 같은 신경망을 활용한 연구들이 나타났다. 이어 인식된 텍스트와 문서 이미지를 모두 함께 활용한 Multi-Modal 모델들이 나타났다. 특히 Microsoft 에서 발표한 LayoutLM 사전학습 모델은 Visual 정보와 Text 정보를 함께 활용한 대표적인 예이다. 버전 3 까지 나왔는데, BERT 와 동일하게 Transformer 기반의 모델로 문서 내 2d 정보(Layout)와 문서 이미지 패치 정보를 함께 Embedding 한 것이 그 특징으로, 텍스트 정보만 활용하는 것보다 Layout 정보와 이미지 정보를 함께 활용하는 것이 여러 Task 에서 더 좋은 성능을 내는 것을 확인하였다.

2.2 LLM

OpenAI 의 ChatGPT 를 필두로 대규모 텍스트 데이터를 활용하여 학습된 거대 언어 모델이 각광받고 있다. 디지털 플랫폼의 발전으로 대규모의 텍스트 데이터가 구축이 되어있고, 또한 딥러닝 기술과 컴퓨팅 수준의 발달로 대규모 모델에 대해 효과적으로 학습할 수 있는 기술적 기반이 마련됨에 따라 이를 활용하여 기존 언어 모델보다 훨씬 더 큰 거대 언어 모델이 나타나게 되었다. 거대 언어 모델의 자연어를 생성하는 성능이 매우 뛰어나 이를 활용하여 각종 자연어 처리 분야의 작업을 자동화하고 효율화하는 사례들이 나타나고 있다. 또한 기존의 특정 Task 에 대하여 Specific 하게 사전 학습된 모델과는 달리 LLM 이 가지고 있는 언어에 대한 이해 능력을 바탕으로 다양한 Task 에 대해서도 높은 성능을 발휘한다는 것을 여러 연구에서 발견하였다.

보통 실무에서는 시스템이 필요로 하는 특정 Task 를 수행하기 위해서는 사전 학습 모델을 기반으로 두고, 여기에 사용자의 데이터를 가지고 미세 조정 학습을 하여 모델의 파라미터를 변경한 후 사용하게 되는데, 거대 언어 모델의 경우에는 이 미세 조정 학습이

쉽지 않다. 거대 언어 모델이 학습하는 데에 사용한 말뭉치의 규모와 실제 학습된 파라미터의 수가 너무 많고 이를 학습하기 위한 컴퓨팅 자원이 많이 필요하기 때문이다. 물론 적은 컴퓨팅 자원으로 적은 수의 파라미터만 튜닝해도 거대 언어 모델이 비슷한 성능을 낸다는 연구 결과가 발표되어 LoRA 와 같은 PEFT(Parameter Efficient Fine-Tuning) 방법론 등이 많이 활용되고 있다.

2.3 In-Context Learning

In-Context Learning 은 거대 언어 모델에 대한 미세 조정 학습 없이 원하는 Task 를 수행할 수 있도록 하는 방법론이다. 거대 언어 모델에 전달하는 Prompt 데이터의 맥락적 의미를 LLM 이 이해하여 이를 기반으로 답변을 생성하게 되는데, 이 과정에서 모델 파라미터의 weight 값은 업데이트되지 않는다. 어려운 미세 조정 학습 과정 없이 원하는 Task 에 대한 거대 언어 모델의 추론 능력을 향상하고자 하는 많은 노력이 있으며, Prompt 를 얼마나 잘 구성하느냐 하는 것을 Engineering 측면에서 접근하는 것이 그 예이다.

In-Context Learning 은 Prompt 에 주는 예시(Demonstration)에 따라 Zero-Shot, One-Shot, Few-Shot 방식으로 구분된다. In-Context Learning 에 대해서 많은 연구가 이루어졌는데, “Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers” 논문에서는 In-Context Learning 이 미세 조정 학습과 동일하다는 것을 수식으로 증명하고 있다. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models” 논문에서는 거대 언어 모델이 수학 계산이나 일반 상식, 의미적 추론에 대해서는 성능이 좋지 않기 때문에, 단계적으로 추론하는 과정을 Prompt 에 입력함으로써 언어 모델이 이를 따라 하여 추론 능력을 향상하는 내용을 발표하기도 했다.

하지만 In-Context Learning 의 한계에 대해서 연구한 논문들도 많은데, “Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?” 논문에서는 거대 언어 모델에 전달하는 Demonstration 에 정답 Label 이 많은 경우 성능이 좋긴 하나, 정답 Label 만으로 이루어진 경우나, 정답과 오답이 랜덤 하게 들어있는 경우나, Label 이 없는 경우 모두 성능의 큰 차이가 없었다고 밝혔다. 또한 제공하는 예시의 수가 8 개 이상이면 성능이 크게 향상하지 않는다는 것도 밝혔다. “Measuring Inductive Biases of In-Context Learning with Underspecified Demonstrations" 논문에서는 거대 언어 모델의 학습에 사용된 데이터에 따라 모델이 귀납적 편향을 지니고 있을 수 있음을 밝혀, 내가 사용하는 LLM 이 내가 하려고 하는 Task 와 잘 맞지 않는 귀납적 편향을 가지고 있다면 In-Context Learning 이 잘 동작하지 않을 수 있다는 것을 밝혔다. 또한 하고자 하는 Task 를 LLM 이 잘 이해하게 하기 위해서는 Prompt 에 넣는 예시의 수를 늘리면 되나, Prompt 에 사용할 수 있는 토큰의 수에 제약이 있기 때문에 그럴 수 없다는 한계 또한 밝혔다.

3 장. LLM 활용 선행 연구

LLM 과 In-Context Learning 을 문서 인식에 활용하고자 하는 연구가 있어 그 방법과 특징을 확인해 보았다.

3.1 ICL-D3IE

이 연구는 “ICL-D3IE: In-Context Learning with Diverse Demonstrations Updating for Document Information Extraction”이라는 논문 제목에서 알 수 있듯이 Document

Information Extraction Task 를 위해 LLM 의 In-Context Learning 을 활용한 연구이다. 이 연구에서도 FUNSD 데이터셋을 활용하였다. 제목에 있는 Diverse Demonstration 이 의미하는 것은 Prompt 에 입력하는 문서에 대한 예시정보가 하나의 형태가 아닌 다양한 여러 형태를 사용했다는 것이다. 먼저 In-Context Learning 의 효과를 높이기 위하여, Train 문서 M 개에서 Test 문서 N 개와 유사한 문서 N 개를 선별한다. Sentence-BERT 를 이용하여 구한 임베딩 정보가 가장 유사한 순으로 선별한다. 이를 Training 샘플 문서라고 부른다. 이어 Diverse Demonstrations Construction 단계로 prompt 를 만드는데, 보통의 In-Context Learning 의 경우에는 단일 형태의 예시 혹은 특정 Task 를 위해 만들어진 예시정보로 구성하는 반면, 이 연구에서는 Task 와 관련이 없는 예시들도 함께 구성한 것이 특징이다.

다양한 예시정보의 예는 다음과 같다.

첫째, Hard Demonstrations 이라 하여 Training 샘플 문서 위 Text 들의 Bounding Box 와 Entity 정보를 구성한 것이 있다. 이 정보를 구성할 때에는 Zero-Shot Prompting 기법을 이용하여 GPT3 모델에 각 Training 문서의 Text 에 대한 Entity 를 예측해 달라고 먼저 질의하고, 그 결과에 대해 F1 점수를 계산하여 성능이 낮았던 Text 들로 구성하는 것이 특징이었다. GPT 가 제대로 답변을 생성하지 못하는 경우에 대해 어떤 것이 정답인지를 잘 추론할 수 있도록 예시 정보를 생성하는 과정이라고 이해하였다.

둘째, Layout-Aware Demonstrations 이라 하여 DIE Task 를 위해 만들어진 Question 에 해당하는 Text 와 그 Answer 에 해당하는 Text 그리고 그 둘 간의 위치를 말로 설명한 문장 정보가 있다. 이 예시 정보를 구성할 때에는, GPT3 모델에 Text 간의 관계를 설명해 달라는 질의를 던져서 받은 결과를 함께 포함한다는 것이 특징이다. 즉 LLM 이 생성한 답변을 다시 Prompt 에 입력하는 것이다.

셋째, Formatting Demonstrations 이라 하여 각 Text 의 Entity 를 묻는 Question 문장과

이에 대한 정답을 구성한 것이 있다.

이렇게 Training 샘플 문서에 대한 총 3 가지 형태의 예시정보를 Prompt 에 구성하고, 이어 Test 문서 정보와, DIE Task 에 대한 질의를 Prompt 에 추가로 합치는 것으로 Architecture 를 구성하였다. 그 결과로는 이미지 기반의, Training 문서 전체를 활용해 Fine Tuning 된 모델에 비해서 FUNSD, CORD, SROIE 세 벤치마크 데이터셋에 대해 추론 성능이 높아진 것으로 나타났다.

3.2 LATIN-Prompt

LATIN-Prompt 라는 것은 Layout and Task Aware Instruction Prompt for Zero-shot Document Image Question Answering 연구에서 제안한 모델이다. 이 연구 역시, Fine Tuning 된 다른 사전 학습 모델을 사용하지 않고 LLM 과 Prompt Engineering 을 통해서 VQA Task 를 수행하고자 한 연구이다. 이 연구에서 주목한 것은 문서 위에서 "\n"으로 표시되는 Line Break 정보와, 각 Text 간 띄어쓰기(Space)의 길이 정보이다. 문서에 대해 OCR 로 얻은 결과를 Layout Recover 라는 과정을 거쳐 "\n"과 "_"로 이루어진 하나의 Text 로 변환하고, 이를 LLM Prompt 에 던져 VQA Task 를 수행하는 것이다. 다른 예시 정보 없이 Zero-Shot Inference 를 수행하였고, 그 결과는 LayoutLM, LayoutLMv2 와 같은 사전 학습 모델을 이용했을 때의 결과보다 더 좋은 경우가 일부 나타남을 확인했다.

3.3 선행 연구의 의의

선행 연구를 통해, LLM 을 활용하여 문서의 Layout 인식이나 Visual Question Answering 과 같은 Task 를 수행하고자 하는 연구가 많이 진행되고 있으며, 그 성능이 기존

이미지 기반 사전 학습 모델에서의 성능보다 높거나 준하는 것을 알 수 있었다. 다만 그 Prompting 기법이 매우 다양했고, 사용한 LLM 의 종류에 따라서 성능 차이가 컸기 때문에 아직은 더 많은 연구가 필요한 것으로 확인하였다.

4 장. 실험

4.1 데이터셋

본 실험에서는 FUNSD 데이터셋을 활용하였다. 이 데이터셋은 FUNSD: A Dataset for Form Understanding 논문에서 소개된 대표적인 Document Information Extraction Task 용 벤치마크 데이터셋이다. FUNSD 데이터셋은 RVL-CDIP 데이터셋 400,000 장의 문서 이미지에서 선별된 199 개의 문서 이미지로 이루어져 있으며, 각 문서 이미지에 대해 Json 형태로 annotation 이라는 이름의 Ground Truth 를 제공하고 있다. FUNSD 는 총 4 개의 Entity Type (Question, Answer, Title, Others)으로 구성되어 있고, 각 Entity 는 고유의 id 와, text, box, linking, label, words 값을 가진다. 해당 Entity 가 가지는 Text 는 하나의 단어로 구성될 수도 있고, 여러 단어의 조합일 수도 있어서 그러한 정보가 Word 에 정의되고 있으며, 문서상의 위치가 box 라는 이름으로 좌 상단, 우 하단 꼭짓점 정보를 통해 표시된다. 특히 linking 이라는 각 Entity 간 Semantic Linking 정보를 포함하고 있어 Document Information Extraction Task 에 많이 활용되고 있다.

표 3. FUNSD 데이터셋 원본

종류	문서 수	Words 수	Entities 수	Linking 수	평균 Token 수
Training	149	22512	7411	4236	4232.6
Testing	50	8973	2332	1076	4555.7

표 4. FUNSD 데이터셋 원본 내 Entity

종류	Header	Question	Answer	Other	총 합
Training	441	3266	2802	902	7411
Testing	122	1077	821	312	2332

본 실험에서는 LLM, 그중에서도 OpenAI 의 ChatGPT 를 활용하고 있는데, LLM 의 특성상 Prompt 에 사용되는 Token 에 제한이 있기 때문에 Token 수를 확인해야 한다. 원본 Ground truth 의 경우 평균 토큰 수가 4000 개 수준이었다. 실제 테스트 환경에서 알 수 없는 항목인 label, linking 정보들은 모두 삭제하여 각 토큰 평균 수는 Training 913, Testing 907 개로 감소하였다.

표 5. Ground Truth 가공 전, 후 비교

가공 전 원본

```
{
    "form": [
        {
            "box": [84, 109, 136, 119],
            "text": "COMPOUND",
            "label": "question",
            "words": [
                {
                    "box": [84, 109, 136, 119],
                    "text": "COMPOUND"
                }
            ],
            "linking": [[0, 37]],
```

가공 후

```
[
    {
        "box": [84, 109, 136, 119],
        "text": "COMPOUND"
    },
    ... 생략
]
```

```
            "id": 0
        },
        ... 생략
    ]
}
```

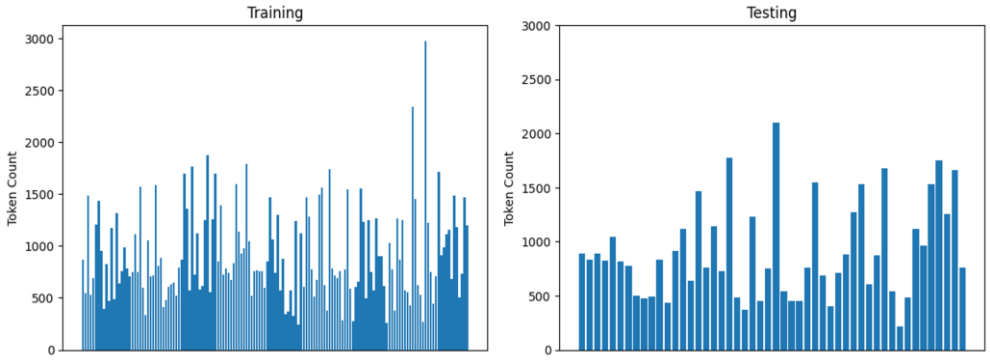


그림 2 Ground Truth 정보에 대한 가공 후 Token 수

앞서 연구 목적에서도 소개하였듯이, LLM 을 통해서 문서의 Key-Value 쌍을 찾는 것이 본 실험의 목적이므로, FUNSD 데이터셋의 Ground truth 데이터의 linking 정보를 활용하여 기대 출력 값을 별도로 생성해 주었다. Entity 가 Question 인 Text 들을 Key 로 간주하였고, 여기에 Linking 으로 매핑되어있는 Answer Entity 들을 모두 합하여 Value 로 간주하였다.

표 6. LLM 을 통해 생성해내고자 하는 기대 출력 값

Key-Value 쌍
[
{
"COMPOUND": "3-Hydroxy-3-methylbutanoic acid (Tur 13) "

```
    },
    {
        "SOURCE": "Lorillard – Organic Chemistry "
    },
    {
        "INVESTIGATOR(S)": "H. S. Tong & M. S. Forte' "
    },
    {
        "SIGNATURE(S)": " "
    },
    {
        "REPORTED": "5/3/79 10/6/80 , Update "
    },
    … 생략
]
```

표 7. 각 문서 별 Key-Value 쌍 정보

종류	평균 Key 개수	Value 가 없는 Key 평균 개수	평균 Key 길이	평균 Value 길이
Training	21.8	8.1	14.0	30.1
Testing	21.4	9.7	12.9	34.2

4.2 실험 환경

본 실험에서는 OpenAI 의 ChatGPT 중에서도 gpt-3.5-turbo-16k 과 gpt-4-1106-preview 모델을 활용하였다. gpt-3.5-turbo 모델은 4,097 개의 토큰을 입력받을 수 있는데, Train 문서와 Test 문서의 정보 그리고 In-Context Learning 을 위해 ChatGPT 에 보내줄 Demonstration 도 필요하므로 최대 16,385 개의 토큰을 입력할 수 있는 gpt-3.5-turbo-16k 모델이 실험에 적합하다고 생각되어 사용하였다. 다른 모델인 gpt-4-1106-preview 의 경우에는 입력 가능한 토큰이 131,073 개로 대폭 늘어나 300 페이지 분량의 책 한 권을

통째로 입력받을 수 있을 만큼 그 활용도가 높아졌다. 다른 LLM 과의 비교도 필요하여 이 모델을 사용해 추가로 실험을 진행하였다. 이미지 자체를 인풋 데이터로 입력받을 수 있는 gpt-4-vision-preview 모델도 발표가 되었는데, 이번 실험의 조건과는 다르다고 생각하여 실험에 사용하지는 않았다.

표 8. Prompt 에 입력한 데이터를 기준으로 구분한 실험의 종류

구분	Train (Text, Bbox)	Train (Key-Value Set)	Demonstration	Test (Text, Bbox)
Zero Shot	-	-	-	O
One Shot	O	O	-	O
Few Shots 1	-	-	{Key's Bbox: Value's Bbox}	O
Few Shots 2	-	-	{Key's Text: Value's Text}	O
Few Shots 3	-	-	{Key's Text & Bbox: Value's Text & Bbox}	O

4.3 Key, Value 의 Text 간 의미적 유사성

4.2 실험 환경 중 Few Shots 라 명명한 실험의 경우, Key-Value 쌍에서 Text 와 Bounding Box 를 따로 혹은 함께 Prompt 에 입력하는 것을 나눠서 실험하였는데, 이렇게 한 이유는 Key 와 Value Text 의 의미적 유사성과 각각의 문서상에서의 위치 관계가 LLM 이 Key-Value 를 찾는 데에 중요하게 작용할 것이라고 가정하였기 때문이다. Key 와 Value 간의 의미적 유사성과 위치관계를 확인해 보았다.

Hugging Face 의 sentence-transformer paraphrase-MiniLM-L6-v2 사전 학습 모델을 활용하여 Key, Value 의 Text 각각의 Embedding 정보를 구하여 코사인 유사도를 구했다.

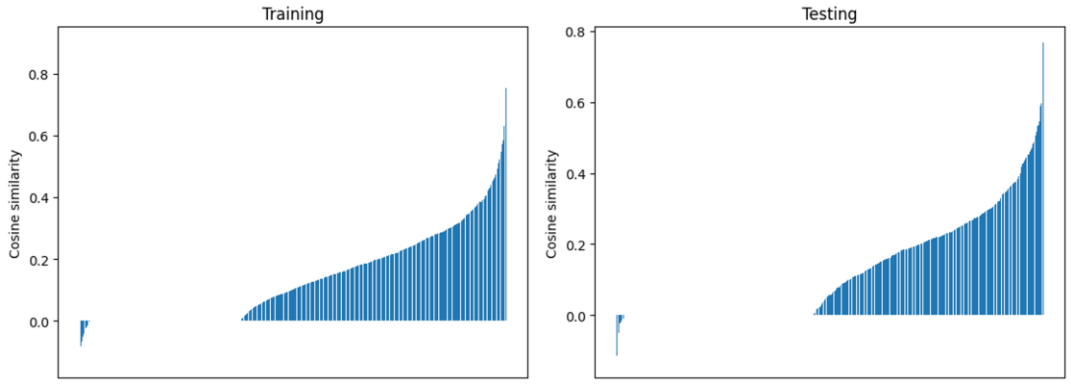


그림 3 Training, Testing 문서 내 Key-Value Text 간 코사인 유사도

그림 3 은 Training, Testing 각각 문서들에 존재하는 모든 Key 와 Value Embedding 간 코사인 유사도를 오름차순으로 정렬한 그래프이다. 유사도가 0 인 것들은 Key 는 존재하나 그에 해당하는 Value 가 없는 경우로, Training 문서의 경우 전체 약 34.6%, Testing 문서의 경우 전체 약 43.3% 로 나타났다.

표 9. Training 문서의 Key, Value, Cos. Sim. 예

Key	Value	Cos. Sim.
PACK AND OR/ CARTON	PACK OR CARTON	0.8994
SQ. INCHES/FEET	7.96875 sq inches	0.7521
PRODUCT	New Products	0.7479
Dollar Cost	$250	0.6327
Date Submitted	September 26 1995 (submit by 30 days after event)	0.5808
Time	1: 45 P. M.	0.5651
Length	85 mm	0.5728
PROJECT NAME	BULL DURHAM Full Flavor King Size Cigarettes	−0.1286
SUBJECT:	HARLEY DAVIDSON 100'S CIGARETTES PROGRESS	−0.1265
CC:	MaDonna Sliker	−0.1122

표 5 에서 알 수 있는 것처럼, Dollar Cost 나 Time 과 같은 Key 에는 실제로 $250 이라는 금액 정보와, 1:45 P.M.이라는 시간정보가 들어왔고, 유사도도 높게 구해졌다. 하지만 표 5 첫 번째 줄의 예처럼, Key 와 Value 에 중복된 값이 있는 경우가 많아 유사도가 높게 나온 경우가 있었다. 또한 제목을 입력하는 곳에 제목과 전혀 다른 내용이 적혀 있어서 음의 유사도가 나온 경우도 있었다. 따라서 Key 와 Value 간의 의미적 유사성이 항상 존재한다고 볼 수 없는 것으로 확인하였다.

4.4 Key, Value 의 Bounding Box 의 위치 관계

Key 의 Bounding Box 정보와 Value 의 Bounding Box 정보를 활용하여 Key 를 기준으로 Value 가 어디에 위치하는지에 대해 확인해 보았다. Key 주변의 공간을 Right, Up Right, Up, Left Up, Left, Down Left, Down, Right Down 이렇게 총 8 방향으로 나누어 어느 위치에 속하는지 계산해 보니 아래 표 6 과 같았다.

표 10. Key 기준 Value 의 위치 비율

Training			Testing		
0.0042	0.0065	0.0014	0.0017	0.0017	0.0067
0.0248	Key	0.8283	0.0334	Key	0.7860
0.0168	0.0861	0.0318	0.0418	0.0853	0.0435

Value 의 대부분이 Right, Right Down, Down 방향에 위치하였고, 그 비율이 Training 문서의 경우 약 93%, Testing 문서의 경우 약 91%로 나타났다.

4.5 Prompt 구성

4.3, 4.4 에서 확인한 바, Key-Value 쌍을 찾기 위해서는 각 Text 의 위치정보를 나타내는 Bounding Box 가 Text 보다는 더 직접적인 영향을 미칠 것으로 가정하고 실험을 진행하였다. 아래는 ChatGPT 에 입력으로 사용한 Prompt 이다.

표 11. Zero Shot Prompt

Prompt
prompt = f""" Input Document : {input} This is the json type data of document. it has list of recognized text on document. Each item has 'text' and 'box'. 'text' is recognized text and 'box' is bounding box. box has 4 element. first and second is the left upper coordinate and third and fourth is right bottom coorditates. so it can represent 2d location of text on document. And Please analysis the Input Document. And find every expected key-value from the 'Input' document. and return the result as a json format below. "key1":"value1", "key2":"value2", "key3":"value3" ... Prompt: 'Answer' :

표 12. One Shot Prompt

Prompt
prompt = f""" 'Document' : {train_x_data} 'key-value pairs' : {train_y_data} 'Document' is the json type data of document. it has list of recognized text on document. Each item has 'text' and 'box'. 'text' is recognized text and 'box' is bounding box. box has 4 element. first and second is the left upper coordinate and third and fourth is right bottom coorditates.

so it can represent 2d location of text on document.

'key-value pairs' is Document's key-value pairs.

Please analysis the example.
And do same thing to find every expected key-value from the 'Input' document. and return the result as same format as the key-value pairs

Prompt:
'Input' : {test_x_data}
'Output' :
"""

표 13. Few Shots Prompt

Prompt
prompt = f""" {test_x_data} This is the json type data of document. it has list of recognized text on document. Each item has 'text' and 'box'. 'text' is recognized text and 'box' is bounding box. box has 4 element. first and second is the left upper coordinate and third and fourth is right bottom coorditates. so it can represent 2d location of text on document. i will show you example what is key-value pair's text. 'Key-Value Pair's text Example' {random_text_pair} Please analysis this and find some pattern of the key-value pair. And Please analysis the Input Document. And find every expected key-value from the 'Input' document. and return the result as a json format below. 'key1':'value1', 'key2':'value2', 'key3':'value3' ... Prompt: 'Answer' :

"""

Few Shots 실험의 경우, 아래 표 10 과 같이 예시 정보를 구성하였고, Training 문서의 정답 Key-Value 쌍에서 랜덤 하게 500 개씩 추출하였다.

표 14. Demonstration 정보

구분	입력

Bounding Box	{
	'[68, 244, 155, 272]': '[215, 260, 530, 277]',
	'[73, 141, 119, 153]': '[145, 141, 208, 154]',
	'[50, 848, 111, 866]': '[131, 849, 222, 866]',
	'[96, 193, 160, 211]': '[184, 196, 490, 213]',
	'[61, 465, 134, 479]': '[122, 457, 596, 580]',
	'[50, 201, 598, 222]': '[60, 237, 341, 291]',
	'[103, 547, 241, 564]': '[247, 546, 365, 561]',
	'[99, 275, 253, 290]': '[335, 277, 372, 292]',
	'[82, 716, 189, 731]': '[306, 715, 389, 731]',
	'[133, 204, 174, 219]': '[177, 201, 229, 216]',
	'[476, 126, 497, 139]': '[525, 124, 539, 138]',
	'[501, 186, 550, 196]': '[582, 176, 646, 194]',
	'[426, 405, 451, 418]': '[397, 432, 482, 476]',
	'[303, 365, 350, 389]': '[318, 392, 339, 478]'
	... 생략
	}

Text	{ 'NOT TO BE CHARGED BY SUPPLIER': '☑', 'LD50 (95% CONFIDENCE LIMITS)': '3.5 (3.1 to 3.9 g /kg', 'Media Name': 'N/A', 'AMOUNT': 'g100ml', 'SECTION(S)': '13 Plaza', 'EVENT LOCATION:': '68th –86th Street New York, NY', 'KENT K.S.': '1', 'Bobbin Width': '50 mm mm', 'New Mexico': 'x',

	'Current Balance Available:': '(886, 220. 53)', ...생략 }
Text, Bounding Box	[[{"text": "COMPOUND", "box": [84, 109, 136, 119]}, {"text": "3-Hydroxy-3-methylbutanoic acid (Tur 13)", "box": [145, 98, 507, 116]}], [{"text": "SOURCE", "box": [85, 141, 119, 152]}, {"text": "Lorillard - Organic Chemistry", "box": [144, 128, 409, 152]}], [{"text": "INVESTIGATOR(S)", "box": [84, 203, 155, 214]}, {"text": "H. S. Tong & M. S. Forte'", "box": [198, 194, 427, 213]}], ...생략 }

4.6 실험 결과

ChatGPT API 의 응답에서 얻어진 Key-Value 데이터와 Testing 문서의 정답 Key-Value 쌍과의 비교를 통해 Recall, Precision, F1 스코어를 계산하였다. ChatGPT 를 통해 얻은 Key-Value 데이터는 바로 비교하지 않고 전처리를 거쳐야 했다. 정답 Key-Value 쌍에서는 Key 값이 “Date:”처럼 Text 마지막에 Colon 값이 붙어있는 경우가 많았는데, ChatGPT 에서는 이러한 경우 Date 라고 Colon 값을 제거하는 전처리를 거쳐 응답하는 경우가 나타났다. Prompt 를 통해 문서 내 존재하는 Text 를 고치지 않도록 실험해 보았으나, 항상 그렇게 동작하지 않았다. 그래서 Colon 값의 경우에는 있거나 없거나 모두 정답을 찾은 것으로 간주하여 결과를 계산했다.

표 15. 실험 결과 (gpt-3.5-turbo-16k 모델)

	Key-Value Pair			Only Key		
구분	Recall	Precision	F1	Recall	Precision	F1

Zero Shot	0.1513	0.0689	0.0946	0.6996	0.4719	0.5636
One Shot	0.3117	0.2286	0.2638	0.798	0.583	0.6737
Few Shots 1	0.3204	0.2696	0.2928	0.6953	0.6195	0.6552
Few Shots 2	0.3466	0.2655	0. 3007	0.7312	0. 5663	0. 6383
Few Shots 3	0.4122	0.2794	0.3331	0.8504	0.5832	0.6919

Key-Value 쌍이 모두 정답 정보와 일치하는 경우, 단지 Key 값만 찾은 경우 2 가지를 따로 계산하였다. 아무런 예시 정보를 주지 않았던 Zero Shot 실험의 경우보다 예시 정보가 들어있었던 나머지 4 실험에서 모두 GPT 의 추론 성능이 증가한 것을 확인하였다. LLM 에서 In-Context Learning 이 문서의 Layout 을 인식하는 데에도 동일하게 적용됨을 확인하였다. Key-Value Pair 의 Text 와 Bounding Box 를 모두 활용했을 때 가장 추론 성능이 좋아졌다. 다만 Key-Value Pair 의 Bounding Box 만을 사용했을 때 보다 Text 만을 사용했을 때의 추론 성능이 조금 더 높았던 점은 의외의 결과이다.

표 16. 실험 결과 (gpt-4-1106-preview 모델)

	Key-Value Pair			Only Key		
구분	Recall	Precision	F1	Recall	Precision	F1
Few Shots 3	0.4386	0.5183	0.6178	0.6179	0.7226	0.6662

표 16 은 gpt4-1106-preview 모델에 대한 결과이다. 표 15 의 gpt-3.5-turbo-16k 모델의 결과와 비교했을 때 전반적인 성능이 올라갔고, 특히 Precision 이 0.27 에서 0.51 로 크게 증가한 것을 볼 수 있다. Key-Value 가 아닌 단지 Key 에 대한 추론 능력의 경우에는 Recall 은 떨어졌으나 Precision 은 증가한 것을 볼 수 있는데, gpt-3.5-turbo-16k 모델이 문서의 대부분의 텍스트를 Key 라고 추론하여 Recall 이 높았다면, gpt-4 모델로 오면서 그러한 현상이 줄어들어 Recall 이 낮아진 것으로 해석하였다.

5 장. 결론

문서 처리 자동화 시스템을 구성할 때, 가장 먼저 필요한 것은 문서 이미지에서 Text 를 추출해 내는 OCR 과정이다. 그다음 단계로는 내가 하고자 하는 Task 에 대하여 학습된 모델 혹은 사전 학습 모델에 대한 미세 조정 학습을 거친 모델을 이용하여 실제 Down Stream Task 를 수행하는 것이다. 하지만 매번 다양한 양식의 새로운 문서들이 나타나는 Open Domain 환경에서는 모델을 매번 재 학습시킨다는 것이 쉽지 않다. 그래서 별도의 학습 과정 없이 모든 문서에 대하여 일반적으로 동작할 수 있게 하는 방법을 찾고자 하였다.

거대언어모델은 특정 Task 에 대해 학습되지 않았음에도 불구하고 다양한 Task 에 대하여 추론할 수 있는 능력을 가지고 있음이 이미 많은 실험과 사용 사례를 통해 밝혀져 있다. 그리고 LLM 에 입력하는 Prompt 의 맥락 정보를 확인하여 그에 맞는 추론을 할 수 있다는 사실도 밝혀져 있다. 이 LLM 을 활용하여 문서를 이해하고 찾아진 정보를 이용해 문서 처리 자동화 시스템을 구축할 수 있는 가능성을 확인해보고자 했다.

OCR 을 통해 추출할 수 있는 정보는 Text 와 그 Text 의 문서 내 좌표를 나타내는 Bounding Box 이다. 이 데이터와 LLM 을 이용하여 문서의 Layout 을 인식할 수 있는지에 대한 실험을 진행하였다. 문서의 Layout 을 인식한다는 것을 확인하기 위해 문서의 Key-Value Pair 를 찾아낼 수 있는가에 대한 실험을 구성하였다. 실험은 OpenAI 의 ChatGPT API 를 이용하여 수행하였으며, Prompt 에 Text 와 Bounding Box 정보를 예시로 주어 실제로 Key-Value Pair 를 생성해 내도록 하였다.

실험 결과 Text 와 Bounding Box 를 모두 활용한 경우 아무런 예시 정보를 주지 않은 Zero Shot 실험에 비해 최대 3 배 이상의 Recall, Precision 점수를 보였다. 이를 통해 LLM 이 문서 인식 Task 의 경우에도 다양한 예시 정보를 통해 그 추론 능력이 올라가는 In-Context Learning 의 효과가 나타남을 실제로 확인하였다. 그리고 Input Token 의 한계가 존재하는 LLM 의 특성을 고려하였을 때, One Shot 방식으로 Training 문서의 전체 정보와 정답 데이터를 주는 것보다 실제 Key-Value Pair 의 Text 와 Bounding Box 정보를 함께 Prompt 에 구성하는 것이 오히려 좋은 성능을 냄을 알 수 있었다.

Key-Value Pair 에 대한 Recall 은 최대 0.43, Precision 은 최대 0.52 의 스코어를 기록하였다. 이는 실무에서 활용하기에는 높지 않은 점수이다. LLM 이 문서의 Layout 을 이해하고 정보를 추출하기에는 추론 능력이 아직은 부족하여, 이미지를 함께 활용한 다른 멀티 모달 사전 학습 모델을 대체하기에는 한계가 있는 것으로 확인하였다. 하지만 GPT3.5 모델의 결과와 GPT4 모델의 결과를 비교해 보았을 때, GPT4 모델의 추론 성능이 상당히 높아졌는데, 향후 LLM 성능 개선에 따라 문서 인식의 성능도 높아질 것으로 기대해 볼 수 있다. 또한 Key-Value 쌍이 아니라 Key 만 찾는 성능을 고려했을 때, Recall 의 성능은 최대 0.85 까지도 나왔는데, 이 Recall 을 더 높일 수 있다면, 수동으로 수행하는 문서 처리 업무에 활용할 수 있는 여지가 있을 것으로 보인다.

향후 연구로 Prompt 에 입력하는 예시 정보를 더 많이 늘린 경우 성능이 어떻게 변화하는지 확인해 볼 필요가 있으며, 앞서 실험 환경에서 언급했던 gpt-4-vision-preview 와 같은 이미지를 Input 으로 허용하는 GPT4 모델에서는 OCR 을 통해 사전에 추출한 데이터를 Input 으로 넣은 경우와 비교하여 어떤 차이가 있는지 실험해 보는 것도 의미가 있을 것이다.

참고문헌

[1] Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, He He, "https://paperswithcode.com/paper/measuring-inductive-biases-of-in-context", arXiv:2305.13299, 2023

[2] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, Furu Wei, "Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers", arXiv:2212.10559, 2023

[3] Guillaume Jaume, Hazim Kemal Ekenel, Jean-Philippe Thiran, "FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents", arXiv:1905.13538, 2019

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, Denny Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models", arXiv:2201.11903, 2023

[5] Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. "ICL-D3IE: In-Context Learning with Diverse Demonstrations Updating for Document Information Extraction", arXiv:2303.05063, 2023

[6] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, Luke Zettlemoyer, "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?", arXiv:2202.12837, 2022

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei, "Language Models are Few-Shot Learners", arXiv:2005.14165, 202

[8] Wenjin Wang, Yunhao Li, Yixin Ou, Yin Zhang, "Layout and Task Aware Instruction Prompt for Zero-shot Document Image Question Answering", arXiv:2306.00526, 2023

[9] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, Lidong Zhou, "LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding", arXiv:2012.14740, 2022

[10] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, "LayoutLM: Pre-training of Text and Layout for Document Image Understanding", arXiv:1912.13318, 2020

[11] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, Furu Wei, "LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking", arXiv:2204.08387, 2022

[12] Zhuosheng Zhang, Aston Zhang, Mu Li, Alex Smola, "Automatic Chain of Thought Prompting in Large Language Models", arXiv:2210.03493, 2022