AI Law agent

1. 목표 및 소개
    프로젝트 목표
    서비스 소개
        타겟 사용자
        주요 기능 및 범위
        성공 지표 및 목표
        잠재적 위험 대처
2. 데이터
    2-1. 데이터 수집
    2-2. 데이터 임베딩
        A. Chunking
        B. ID
        C. Content 구성과 Metadata
3. Agentic Workflow
    3-1. 질문 유효성 판단 노드
    3-2. 질문 유형 분류 노드
    3-3. 질문 전처리 노드
    3-4. RAG agent 노드
4. 평가 및 결과 정리
    토의
5. 문제점 및 개선안
    1차 목표 회고
        소스 데이터의 다양성 및 양 부족으로 인한 성능 저하
        RAG agent의 적용이 오히려 성능에 방해가 된 상황
    Appendix
    사용한 법령 데이터
    프롬프트 조절
    유효성 판단 노드의 토큰 통계량

1. 목표 및 소개

2025.07.02 프로젝트 시작

2025.07.06 프로젝트 1차 목표 달성

2025.07.17 1차 목표 정리 및 2차 목표 계획 수립

프로젝트 목표

이 프로젝트는 대한민국 공공 법률 데이터를 활용하여 법률 정보 검색 시스템과 법학 교육 지원 플랫폼을 결합한 AI 기반 솔루션을 개발하는 것을 목표로 한다.

주요 목적은 법률 지식이 부족한 일반 시민들에게 관련 법령 조문과 판례등의 법률 정보를 보기 쉽게 정리하여 제공하는데 있으며, 특히 변호사 상담 전 배경 지식을 쌓을 수 있는 무료 공공 서비스를 지향한다.

시스템은 사용자의 질문을 입력받아 관련 법령과 판례를 기반으로 답변을 제공하며, 초기 단계에서는 형법 분야에 초점을 맞춰 진행할 예정이다. 궁극적으로는 정확하고 신뢰할 수 있는 정보를 통해 법률 접근성을 높이는 데 기여하고자 한다.

이후에는 법학도를 대상으로 **개인화, 즉각적인 질의응답, 심층 분석, 최신성, 연관성 연결등의 부가적인 기능이 고려된** 법률 문제풀이를 제공할 예정이다.

프로젝트의 1차 목표는 몇가지의 법령을 기반으로하여 사용자에게 질문과 관련된 법령 조문을 정리하여 제공하는 것이다. 차후 **법률 데이터의 양 증가, 결과 요약문의 퀄리티 상승, 새로운 기능 추가** 등을 진행할 예정이다.

서비스 소개

타겟 사용자

- **일반 시민**: 일상생활 속 법률 문제 관련 참고 자료를 **"조문 + 판례" 요약 및 해당 링크를** 통해 쉽고 빠르게 확인하고자 하는 사용자
- **법학도**: 법학 시험(사법시험, 로스쿨 시험 등)에 대비하기 위해 **AI 생성 문제풀이** 기능을 필요로 하는 학습자

주요 기능 및 범위

- 시스템의 핵심 기능은 **사용자의 질문을 분석**하여 그것과 **가장 관련있는 법령 조문과 판례를 제공**하는 것이다.
- 형법 데이터를 우선적으로 다루며, **법령과 판례 데이터를 요약하여 제공**하고, 해당 데이터에 대한 **참조 링크를 함께 제시**할 예정이다.
- 초기 범위는 형법을 포함한 관련 법령 9개로 한정하며, 다국어 지원이나 추가 접근성 기능 및 다른 서비스는 포함하지 않는다.

성공 지표 및 목표

- 프로토타입의 성공은 적절한 법령 조문과 판례를 정확하게 가져오는 데 초점을 맞춘다.
- 무료 공공 서비스로서 운영되며, 법률 정보의 참고 용도로 활용되도록 설계한다.

잠재적 위험 대처

변호사법을 고려하여 다음 두가지를 적용한다

- **경고문 필수 표기**: "※ 본 시스템은 법률 자문이 아닌 단순 참고용 정보를 제공합니다"
- **기능 제한**: 법령 해석·예측 기능 전면 배제

2. 데이터

사용자가 받아야할 법률 데이터는 항상 최신이어야 하므로 RAG를 활용하는 것이 효율적이다. RAG를 진행하기 위해 아래와 같이 데이터 수집과 임베딩을 진행했다.

2-1. 데이터 수집

형사법과 관련된 총 157개의 법령을 RAG를 위한 데이터 소스로 활용했다.

부칙을 제외하고 본칙만 파싱했다.

2-2. 데이터 임베딩

법령전문을 임베딩하는 것은 검색 과정에서의 품질 저하, 증강 및 생성 과정에서 토큰 낭비와 컨텍스트 오버로드의 발생으로 인한 정보 누락으로 이어질 수 있으므로 적절한 chunking이 필요하다.

A. Chunking

형법을 포함한 대한민국의 모든 법률은 [편, 장, 절, 조, 항 ...] 방식으로 구조화 되어 있다. 이 중 조문의 정의는 “법적으로 완결된 의미를 가지는 기본 단위”로 조문 단위의 chunking을 진행하면 RAG가 효과적으로 진행될 것이라 판단했다.

B. ID

법령 조문을 이름으로 찾기 쉽도록 document의 id를 법령과 조문번호의 조합으로 지정했다.

> ex) 형법 15조, 형사소송법 제109조의2

C. Content 구성과 Metadata

아래 그림 처럼 조문의 제목과 본문 만으로는 해당 조문을 다른 조문과 의미적으로 차별화 할 수 없어서 상위 계층의 장명 및 절명을 content에 추가하여 임베딩했다.

```
제196조(미수범)
제192조제2항, 제193조제2항과 전조의 미수범은 처벌한다.

제197조(예비, 음모)
제192조제2항, 제193조제2항 또는 제195조의 죄를 범할 목적으로
예비 또는 음모한 자는 2년 이하의 징역에 처한다

제212조(미수범)
제207조, 제208조와 전조의 미수범은 처벌한다.
```

Document 예시) 법령이름 + 조문 번호 + 장명 + 절명 + 조문명 + 조문 내용

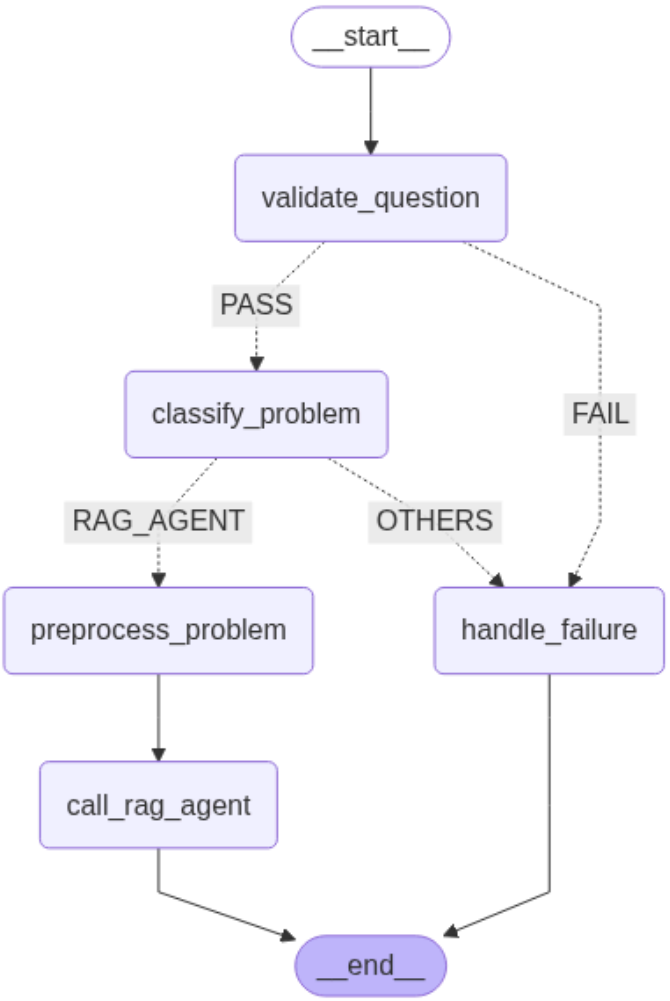
```
형법 제196조
먹는 물에 관한 죄<개정 2020. 12. 8.>
미수범
제192조
제2항
,
제193조
제2항
과 전조의 미수범은 처벌한다.
```

3. Agentic Workflow

Naive RAG를 포함한 간단한 체인 구조만으로 로직을 진행하기에는 아래의 장애물들이 있어 불가능했다. 이를 해결하기 위해 LangGraph와 Agent를 활용하여 LLM workflow를 구성했다. Workflow의 테스트는 KMMLU의 Criminal-Law 데이터로 진행했다.

1. 유효하지 않은 입력 처리 - 일상 질문 및 답변이 불가능한 질문 등
2. 법률 질문 유형별 분기 처리 - 법률 및 판례 제시 유형 or 이론 및 방법론적 판단 등
3. 조문의 특성 - 다른 조문을 참조하는 조문의 존재
4. 조문의 특성으로 인한 단계적 RAG의 필요성

전체 workflow는 아래 그림과 같다.



3-1. 질문 유효성 판단 노드

유효하지 않은 입력을 처리하기 위한 노드로 주어진 입력이 법률과 관련된 질문인지, 해결 가능한 질문인지를 판별한다.

판별에 **평균 680토큰이 소모**된다. KMMLU 데이터 중 **유효하지 않은 입력**들만 모아 이후의 **workflow를 진행했을 때 평균 8343토큰이 소모**된다. 계산하면, **12번의 입력 중 1개의 입력(전체의 약 8%)이 법률 질문이 아니거나 해결 불가능한 질문**이어야지 이 노드가 **효율적**일 것이다.

3-2. 질문 유형 분류 노드

같은 법률 도메인이더라도 질문의 유형마다 필요한 추론의 과정이 다르다. 따라서 이후 유형에 맞는 로직을 적용하기 위해 질문의 유형을 먼저 라벨링한다.

KMMLU를 기반으로 문제의 유형을 구분했을 때, 범죄의 원인에 관한 이론을 다루는 유형과 법령 및 판례를 기반으로 추론해야하는 유형 두가지로 구분이 가능했다.

이론)

29. 교정학 및 형사정책의 연구방법에 대한 설명으로

A) 범죄(공식) 통계표 분석 방법은 ... 유용하며, 실
B) 참여 관찰 방법은 ... 연구결과를 객관화할 수 있
C) 실험적 연구방법은 ... 유용하며, 인간을 대상으로
D) 사례조사 방법은 ... 개인의 정보 획득을 바탕으로

법령 해석)

4. 명예훼손죄에 대한 설명으로 옳지 않은 것은?

A) ... 행위가 형법 제310조의 위법성 조각사유에 해당.
B) 언론매체가 ... 허위임을 충분히 인식하면서도 이를
C) 개인 블로그의 비공개 대화방에서 ... 타인의 명예를
D) 정보통신망을 이용한 명예훼손의 경우 범죄 종료 ...

3-3. 질문 전처리 노드

법적 상황을 올바르게 판단하기 위해서는 **행위의 주체**가 명확해야하고, **모호한 표현**이 없어야 하며, **소유격 관계**가 분명해야한다. 따라서 사용자의 질문에서 앞서 말한 부분을 적용하는 노드를 구현했다.

KMMLU 평가 데이터 기준 평균적으로 **1356개의 토큰이 사용**된다.

예시) 윗부분이 적용 전이고 아래부분이 적용 후다. 각 행위나 사물에 대해서 **주체가 명시**되도록 처리되었다.

甲은 처乙과 부부싸움을 하다가 화가 나서 폭행의 고의로 乙의 가슴을 세게 밀쳤고,
乙은 그 충격으로 사망... 범행을 은폐하기 위하여 탁자에 불을 붙인 후 ...
형사책임에 대한 설명으로 옳지 않은 것은?

A) 처乙에 대해서는 폭행치사죄, 집에 대해서는 방화죄가 성립한다.
B) 만약 살인의 고의로 처乙을 실신케 한 후 집에 방화하여 ...
C) ... 허위의 보험금 지급 청구서 작성행위는 ...
보험회사에 그 보험금 지급 청구서를 ...
D) 범행을 은폐하기 위하여 탁자에 불을 ... 증거인멸죄가 성립한다.

A) 甲이 처乙에 대해서 폭행치사죄가 성립하고, 甲의 집에 대해서 방화죄가 성립한다.
B) 만약 甲이 살인의 고의로 처乙을 실신시킨 후 甲의 집에 방화하여 ...
C) ... 甲이 자신의 명의로 허위의 보험금 지급 청구서를 작성하는 행위는 ...
甲이 보험회사에 자신의 보험금 지급 청구서를 ...
D) 자신의 범행을 은폐하기 위하여 甲이 탁자에 ... 甲에게 증거인멸죄가 성립한다

3-4. RAG agent 노드

아래와 같이 **법령 조문** 중 **다른 조문을 참조하는 경우** 해당 조문을 지칭하는 정보를 추출해내어 참조된 조문을 찾아갈 필요가 있다. 심지어, 참조의 방식이 **"전조"같이 자연어로 되어 있는 경우**도 고려해야한다.

제196조(미수범)
제192조제2항, 제193조제2항과 전조의 미수범은 처벌한다.

RAG를 진행함에 있어 단순히 검색, 증강, 생성 순으로 진행하면 되는 것이 아니라, **검색과 증강 단계 사이에 참조 관계를 풀어가면서 필요한 정보를 추가적으로 탐색**하는 과정이 요구되고 이 과정이 **맥락을 지속적으로 유지한 채로 진행**되어야 하기 때문에 Agent를 활용하기로 결정했다.

Agent가 활용하는 tool은 아래 두가지이다.

1. Retrieve tool
    a. 법령의 제목을 필터링 조건으로 사용하여 semantic search로 조문을 검색한다
2. 조문 탐색 tool

a. “법령 조문번호” 형태의 id로 저장되어 있는 조문의 id를 통하여 조문을 탐색한다

b. 한번에 여러개의 조문을 찾아 올 수 있다.

Agent의 행동이 recursion limit을 넘어갈 때를 대비하기 위해 RAG agent를 호출하는 노드에서 오류 발생 시 오류가 나기 전까지의 대화기록을 모두 입력으로 받아 답변을 생성해내는 로직을 추가했다.

4. 평가 및 결과 정리

평가는 KMMLU의 Criminal-Law 테스트 데이터셋(200 row)로 진행했다. 입력은 형법 관련 4지 선다형 문제로 최종 평가는 Agentic workflow의 결과물을 context로 하여 OpenAI의 **gpt-4o-mini**가 주어진 4지 선다형 문제를 얼마나 잘 맞추는지 그 정확도를 기반으로 진행했다.

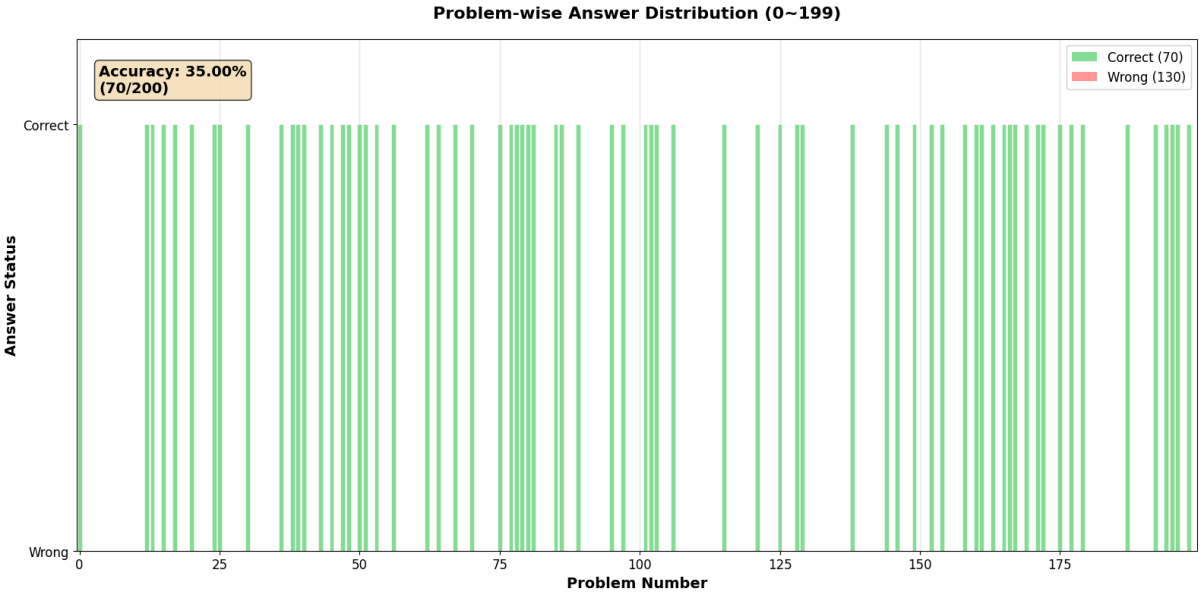
버전	입력 구성	정확도	정답/전체	주요 특징
Base	문제 + 선택지	**35.0%**	70/200	순수 LLM, 추가 정보 없이 진행
v1	문제 + 선택지 + 형법 조문 5개	**35.0%**	70/200	Naive RAG 진행
v2	문제 + 선택지 + RAG agent 결과물	**27.5%**	55/200	Agentic RAG 진행
v3	문제 + 전처리된 선택지 + RAG agent 결과물	**31.0%**	62/200	Agentic RAG + 선택지 전처리
v4	문제 + 전처리된 선택지 + RAG agent 결과물	**35.5%**	71/200	Agentic RAG + 선택지 전처리 + 유형 분류 추가

토의

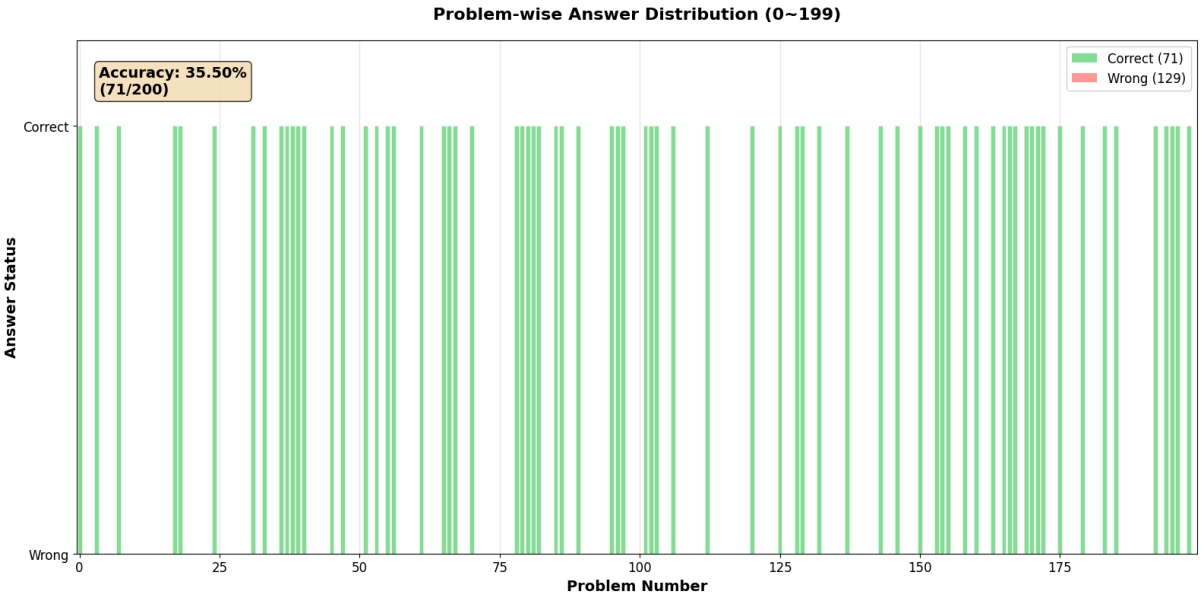
아무런 추가적인 정보 없이 문제와 선택지만 입력한 결과(Base)와 Agentic RAG를 추가 노드와 함께 적용시킨 결과(v4)가 0.5% 정확도 차이로 마치 아무런 차이가 없어 보인다. 하지만, 정답 분포를 보게 되면 Base와 v4가 총 정답 수는 비슷할지언정 어떤 문제를 맞추었는지에 대해서는 차이가 있는 것을 확인할 수 있다. 정답 70개 중 약 37%(26개)가 서로 상이했으며, 이는 적어도 26개의 입력에 대해서 본 서비스가 유용한 답변을 제시했다고 볼 수 있다.

그리고, 본 서비스의 답변이 정답을 맞추는데 방해가 된 경우는 RAG agent의 도움없이 간단한 배경지식만으로도 답변할 수 있는 케이스를 아직 우리 서비스가 분류해내고 있지 못하다는 증거이며 추가적인 전처리 과정이나 분류 과정이 필요하다고 볼 수 있다.

Base 분포)



v4 분포)



5. 문제점 및 개선안

1차 목표까지 진행했을 때 파악한 문제점은 다음과 같다.

1차 목표 회고

소스 데이터의 다양성 및 양 부족으로 인한 성능 저하

핵심 문제점은 다음과 같다.

- 형법 포함 총 9개의 조문 만으로는 다양한 사례를 충분히 커버하지 못한다.
- KMMLU의 성능이 특히 낮은 이유는 문제 중 상당수가 특정 상황에서 조문들이 활용되는 예시가 있는 판례가 필요하기 때문이다.
- 게다가, 부칙에 대한 이해 부족으로 부칙을 소스 데이터로서 활용하지 못했다.

개선 계획은 다음과 같다

1. 국가 법령 정보 OpenAPI를 활용하여 다른 법령 및 판례 데이터 확보
2. 판례 데이터 중 판례 요지를 기반으로 document화 후 임베딩 진행
3. 부칙 데이터 또한 그에 맞는 chunking 전략으로 본칙과 함께 document화 후 임베딩
4. 판례를 Retrieve하는 tool을 구현하여 RAG agent에 추가

RAG agent의 적용이 오히려 성능에 방해가 된 상황

핵심 문제점은 다음으로 파악된다.

- 형법 추론의 기본은 구성요건-위법성-책임 이라는 3단계 구조를 중심으로 접근해야 하는 것이지만, 프롬프트에 적용되지 않았다.
- 이외에도 추가적으로 쟁점 정리, 죄수론, 공범, 판례 및 학설 검토 등의 복잡한 추론 과정이 포함되어야하는 경우가 있음에도 이를 적용할 시스템이 미비했다.
- 복잡한 추론 과정을 단일 agent가 처리하게 됨으로써 복잡도 증가에 따른 전체 context의 증가로 Agent의 성능이 하락했다.

결과적으로 복잡한 로직을 요구하는 형법 추론 task에서 **기초적인 프롬프트 및 workflow**로는 적절한 정보를 가지고 있더라도 높은 성능을 기대하기 어렵다.

개선 계획은 다음과 같다.

- Multi Agent system 도입을 통한 극복
- 각 추론 단계(구성요건-위법성-책임)을 담당하는 전문 agent들로 구성된 시스템을 구축한다.
- 각 전문 agent들이 실제 전문가의 추론 과정을 모사할 수 있도록 프롬프틀 설계하고 tool을 개발하여 적용한다.

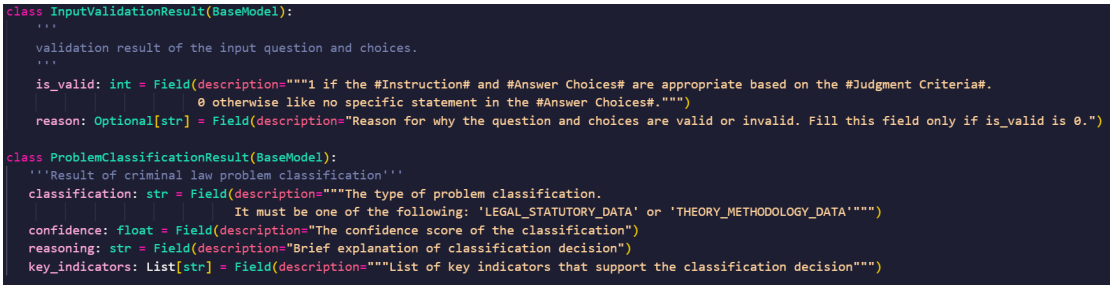
Appendix

사용한 법령 데이터

```
{
   "형법": "https://www.law.go.kr/법령/형법",
   "형사소송법": "https://www.law.go.kr/법령/형사소송법",
   "폭력행위등처벌에관한법률": "https://www.law.go.kr/법령/폭력행위등처벌에관한법률",
   "부정수표단속법": "https://www.law.go.kr/법령/부정수표단속법",
   "도로교통법": "https://www.law.go.kr/법령/도로교통법",
   "특정범죄가중처벌등에관한법률": "https://www.law.go.kr/법령/특정범죄가중처벌등에관한법률",
   "마약류불법거래방지에관한특례법": "https://www.law.go.kr/법령/마약류불법거래방지에관한특례법/",
   "소송촉진등에관한특례법": "https://www.law.go.kr/법령/소송촉진등에관한특례법",
   "벌금미납자의사회봉사집행에관한특례법": "https://www.law.go.kr/법령/벌금미납자의사회봉사집행에관한특례법"
}
```


프롬프트 조절

Agentic workflow 내 모든 노드에서 프롬프트로 특정 task에 대한 지시를 내릴때 추가적으로 **답변의 이유를 제시해 달라고 요청**했다. 이런 식으로 진행한 이유는 reason부분을 답변에 추가하는 것이 복잡한 작업의 성능을 어느정도 올려주기 때문이기도 하지만, 내가 의도하지 않은 대답을 모델이 출력했을 때 **reason을 기반으로 프롬프트를 첨삭하기 위해서가 가장 핵심적인 이유**다.



```python
class InputValidationResult(BaseModel):
    '''
    validation result of the input question and choices.
    '''
    is_valid: int = Field(description="""1 if the #Instruction# and #Answer Choices# are appropriate based on the #Judgment Criteria#.
                          0 otherwise like no specific statement in the #Answer Choices#.""")
    reason: Optional[str] = Field(description="Reason for why the question and choices are valid or invalid. Fill this field only if is_valid is 0.")

class ProblemClassificationResult(BaseModel):
    '''Result of criminal law problem classification'''
    classification: str = Field(description="""The type of problem classification.
                                It must be one of the following: 'LEGAL_STATUTORY_DATA' or 'THEORY_METHODOLOGY_DATA'""")
    confidence: float = Field(description="The confidence score of the classification")
    reasoning: str = Field(description="Brief explanation of classification decision")
    key_indicators: List[str] = Field(description="""List of key indicators that support the classification decision""")
```

유효성 판단 노드의 토큰 통계량

토큰 사용량

유효하지 않은 입력의 토큰 사용량

- 편차가 매우 크다.
- 토큰 사용량은 다음 3개의 유형으로 그룹화가 가능하다.
  - RAG agent 에서 tool 사용없이 지나가는 경우(1200 토큰)
  - RAG agent에서 tool을 몇번 사용하는 경우(4000~6000 토큰)
  - RAG agent 에서 recursion limit 까지 동작을 계속하는 경우(46007 토큰)

```
데이터 개수: 10

평균 (Mean): 8343.30
중앙값 (Median): 3865.5
최대값 (Max): 46007
최소값 (Min): 1200
범위 (Range): 44807
표준편차 (Std Dev): 13379.15
```