형법 법령 검색 성능 개선 : 도메인 특화 데이터셋과 Sentence-level BERT 활용 연구

김재헌

November 24, 2025

Abstract

대한민국 형법 문항을 쿼리(query)로, 사건 개요 문장을 패시지(passage)로 하는 문장 검색 문제를 해결한다. 해당 문제를 LLM으로 해결하기 위해 모델이 학습할 데이터를 생성 및 가공하고, 기존 모델에 새로운 모듈을 추가하여 구조화된 쿼리에 대한 문장 검색 성능 개선을 연구한다.

1 서론

『Government at a glance 2021』(OECD, 2021) 자료에 따르면, 시민 1,000명을 대상으로 사법부에 대한 신뢰도를 조사한 설문(14.3. Citizen confidence in the judiciary system and the courts, 2010 and 2020)에서 한국의 2020년도 사법부 신뢰도는 22%였다. OECD 평균은 대략 57%였으며, 콜롬비아 27%, 슬로바키아 19% 등의 수치를 고하면 저조한 수치임을 알 수 있다. 이러한 사법부에 대한 불신의 원인 중 하나를 형사소송에서 대중이 기대하는 처벌 강도와 실제 판례에서의 처벌 강도 사이의 간극 때문으로 보고, '일반적으로 기대되는 처벌 강도'를 법에 익숙하지 않은 일반인에게도 합리적으로 설명할 수 있다면 사법부에 대한 신뢰도가 회복되리라 판단했다.

따라서 본 과제에서는 사건 경위 문장이 제시되었을 때 이에 적용되는 법령을 자동으로 찾아줄 수 있는 모델을 구축하는 것을 목표로 하였으며, 이를 위해 대한민국 형법 문서를 바탕으로 학습 데이터셋을 구축하고, BGE-M3 모델 구조를 일부 변형시켜 검색 성능 개선을 도모하였다. [1]

2 관련연구

2.1 BGE-M3(Chen et al., 2024)

BGE-M3는 BAAI(Beijing Academy of Artificial Intelligence)와 University of Science and Technology of China에서 제안한 텍스트 임베딩모델이다. M3는 Multi-Linguality, Multi-Functionality, 그리고 Multi-Granularity를 의미한다. Multi-Linguality는 다국어 및 언어 간 교차 검색을, Multi-Functionality는 Dense Retrieval, Sparse Retrieval, 그리고 Multi-vector Retrieval 세 가지 검색방식을 지원

[1] 코드 및 데이터: https://github.com/RGB234/capstone-2025-spring

함을, 그리고 Multi-Granularity는 최대 8,192 토큰까지 다양한 입력 길이 처리가 가능함을 의미한다.

- **Dense Retrieval**

BGE-M3의 인코더인 XLM-RoBERTa의 마지막 은닉층에서 출력되는 토큰 임베딩들을 바탕으로 문장 임베딩을 생성한다. 여기서는 첫 번째 토큰인 [CLS] 토큰 임베딩을 선택하여 문장 임베딩으로 사용한다 (Chen et al., 2024). XLM-RoBERTa의 마지막 은닉층을 H라고 한다면 쿼리 q의 임베딩 벡터 e_q와 패시지 p의 임베딩 벡터 e_p, 그리고 둘의 내적인 dense score는 다음의 수식으로 표현된다:

$$e_q = \text{norm}(H_q[0]), \quad e_p = \text{norm}(H_p[0]), \quad s_{\text{dense}} \leftarrow \langle e_q, e_p \rangle. \tag{1}$$

- **Sparse(Lexical) Retrieval**

Sparse Retrieval에서는 토큰의 임베딩들을 선형 레이어(sparse linear)에 통과시켜 스칼라값인 가중치로 변환시킨다. 그리고 쿼리와 패시지에 공통으로 존재하는 토큰 가중치로 Joint importance를 계산해 Sparse score로 사용한다. 만약 한 문장에 동일한 토큰이 여러 번 등장하면 그 가중치 중 최댓값을 해당 토큰의 가중치로 선택하는 방식을 사용하였다 (Chen et al., 2024).

$$w_{q_t} \leftarrow \text{ReLU}(W_{lex}^T H_q[i]), \quad w_{p_t} \leftarrow \text{ReLU}(W_{lex}^T H_p[i]), \quad W_{lex} \in \mathbb{R}^{d \times 1}. \tag{2}$$

$$s_{\text{lex}} \leftarrow \sum_{t \in q \cap p} \left(w_{q_t} * w_{p_t} \right). \tag{3}$$

q_t는 쿼리 내의 임의의 토큰 t를 의미한다.

- **Multi-vector Retrieval**

Dense retrieval의 확장 버전이다. 출력 행렬 H_q와 H_p를 바탕으로 Colbert-embedding을 얻고, late interaction을 수행하여 Multi-vector score를 구한다:

$$E_q = \text{norm}(W_{\text{mul}}^T H_q), \quad E_p = \text{norm}(W_{\text{mul}}^T H_p). \tag{4}$$

$$s_{\text{mul}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \max_{1 \le j \le M} \left(E_q[i] \cdot E_p^{\text{T}}[j] \right). \tag{5}$$

$W_{\text{mul}} \in \mathbb{R}^{d \times d}$ 는 학습 가능한 투영행렬이며, N과 M은 각각 쿼리와 패시지의 길이이다.

2.2 A Sentence-level Hierarchical BERT Model (HBM)(Lu et al., 2021)

Limited labelled data에서의 문서 분류 문제를 푸는 것이 목적인 모델이다. Token-level RoBERTa 인코더 위에 Sentence-level BERT 인코더를 쌓고, 마지막 출력층으로 문서 라벨 예측을 위한 prediction layer가 있는 구조이다.

우선 RoBERTa 인코더로부터 얻어낸 문장 임베딩 행렬인 $\mathcal{D} = (e_1, e_2, ...e_m)$를 Sentence-level BERT 인코더의 입력으로 사용하여 출력 행렬 $\mathcal{Z}$를 얻는다. 그리고 $\mathcal{Z}$를 FFNN에 통과시켜 중간문서표현 $\mathcal{S} \in \mathbb{R}^{1\times d_e}$를 만든다. 해당 FFNN의 내부 구조는 $Z \in \mathbb{R}^{m\times d_e}$에 mean pooling을 적용하고(i.e. $\mathrm{Avg}(\mathrm{Z}) \in \mathbb{R}^{1\times d_e}$) 선형 변환($\mathrm{W}^t \in \mathbb{R}^{d_e\times d_e}$)한 후 활성함수 하이퍼볼릭탄젠트 함수를 적용하는 구조이다:

$$S = \mathrm{Tanh}\big(\mathrm{Avg}(\mathcal{Z}) \times W^t\big) \tag{6}$$

마지막으로 $\mathcal{S}$를 prediction layer에 통과시키면 최종적으로 문서 라벨의 예측값이 출력된다. prediction layer는 linear layer 한 개와 softmax head로 구성된다. 즉, prediction layer에서 각 클래스의 raw score는 다음의 수식으로 표현된다:

$$[t_1, t_2, ...t_y] = S \times W \tag{7}$$

$W \in \mathbb{R}^{d_e\times y}$는 linear layer의 가중치 행렬이며, t_1, t_2, ... t_y는 softmax head를 통과하기 전 각 클래스 0^{th}, ... y^{th}의 raw score이다.

3 데이터셋 준비

3.1 파인튜닝 데이터셋

- **Manual Query Reformulation based on Domain Knowledge**

쿼리는 시행일자 2025년 3월 18일 형법 (법률 제20795호)에서 제2편 제1장부터 제42장까지의 내용을 담았다. 법률 문서 특성상 한 조항에서 다른 조항을 준용하거나 참조하는 경우가 많은데, 해당 준용 표현들은 그 자체로는 의미가 온전하지 않으므로 준용하는 조항의 내용을 추가하거나 혹은 표현을 대체하는 방식으로 수정된 문장을 학습 쿼리로 사용하여 검색 성능 향상을 도모하였다 (Mandal et al., 2017).

예를 들어, "단체 또는 다중의 위력을 보이거나 위험한 물건을 휴대하여 제136조, 제138조와 제140조 내지 전조의 죄를 범한 때에는 각 조에 정한 형의 2분의 1까지 가중한다"라는 조항(제144조 제1항)에 대해 "단체 또는 다중의 위력을 보이거나 위험한 물건을 휴대하여 제136조 공무집행방해 제138조 법정 또는 국회회의장 모욕과 제140조 내지 전조의 공무상 비밀표시 무효, 부동산 강제집행 효용 침해, 공용서류 등의 무효 공용물의 파괴, 공무상 보관물의 무효에 관한 죄를 범한 때에는 각 조에 정한 형의 2분의 1까지 가중한다"라는 식으로 내용을 추가하였다.

이렇게 외부 참조로 인한 정보의 공백을 사람이 직접 메우는 과정에서 미수범, 예비음모, 병과, 상습범 등 형의 경중을 결정짓는 내용은 중요도가 낮다고 판단하고, 데이터 처리 과정에 소요되는 시간을 단축하기 위해 학습 쿼리에서 제외하였다. 죄의 성립을 규정하는 조항을 위주로 작업하여 학습 쿼리에 사용한 셈이며, 그 결과 총 363개의 쿼리(363개 조문)를 학습에 사용하였다.

- **Synthetic Passage Generation via LLMs**

학습 쿼리의 가공이 끝나고 난 후 각 쿼리마다 대응되는 양성 샘플은 GPT-5-mini 모델로 생성한 문장을 사용하였다. 공소사실 양식의 문장 20개를 출력하도록 프롬프트를 작성하여 요청한 뒤, 응답 문장 10개는 학습 데이터셋에, 나머지 10개는 검증 데이터셋으로 나눠 저장하였다. Bonifacio et al. (2022)에서는 LLM이 생성한 텍스트 중 저품질의 노이즈 데이터를 Reranker를 바탕으로 제거하였지만, 본 연구에서는 사용하지 않은 방식이다.

음성 샘플은 네거티브 샘플링 방식을 사용하여, 쿼리와의 유사도 상위 10위부터 210위까지 총 200개 중 15개 샘플을 랜덤 추출해 음성 샘플로 사용했다. 유사도를 구하는 과정에서 사용한 인코더 모델은 BAAI/bge-m3 모델을 사용하였다.

3.2 테스트 데이터셋

테스트 데이터셋은 KLAID 데이터셋중 일부를 선택하여 사용했다. 해당 데이터셋은 사건 경위와 그에 대응하는 법령 목록으로 행 데이터가 이루어져 있다. 대응되는 법령 목록에는 형법 외에도 도로교통법, 교통사고처리특례법 등이 있으나 법령 목록에 오직 형법만을 포함하는 행 데이터만 선택하였다.

그렇게 쿼리(법령)는 총 65가지이며, 쿼리당 양성 샘플은 10개가 되도록 랜덤 추출하였다. 쿼리는 학습 데이터셋에 있는 쿼리가 62개, 나머지 3개는 파인튜닝 데이터셋에 없는 쿼리이다. 언급한 3개의 쿼리는 파인튜닝 데이터셋 쿼리를 가공하는 과정에서 제외된 조항들로 형법 제264조, 제285조, 제332조이다.

4 모델 커스터마이징

4.1 Sentence-level attention for queries (BGEM3-SAQ)

BGE-M3(Chen et al., 2024)커스터마이징을 위해 세 retrieval score(dense score, sparse score, multi-vector score)에 sentence attention score를 하나 더 추가하였다. 해당 점수는 기존 모델 구조에서 추가한 Sentence-level BERT를 활용하여 계산된다.

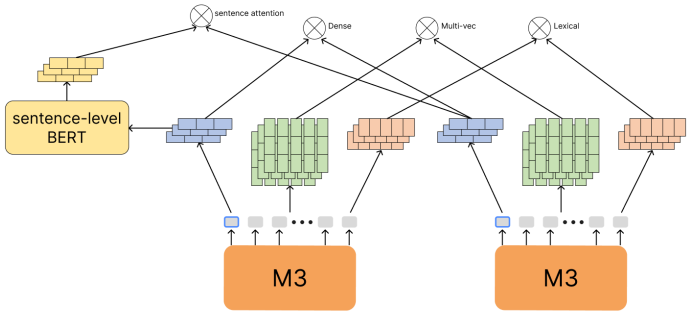

Figure 1: BGEM3-SAQ

- **Sentence-level BERT**

XLM-RoBERTa의 출력인 쿼리 임베딩 행렬 $\mathcal{D}_q = (e_{q_1}, e_{q_2}, ..., e_{q_m})$을 Sentence-level BERT 인코더에 통과시켜 얻은 출력 행렬 $\mathcal{Z}_q$와 $\mathcal{D}_p$를 내적하여 sentence attention score를 구한다.

$$e_q = \text{norm}\left(H_q[0]\right), \quad \mathcal{D}_q = \left(e_{q_1}, e_{q_2}, ..., e_{q_m}\right). \tag{8}$$

$$\mathcal{Z}_q = \text{norm}\left(H'_{D_q}\right), \quad let\ e'_q \in \mathcal{Z}_q, \quad s_{\text{sent}} \leftarrow \langle e'_q, e_p \rangle. \tag{9}$$

H은 BGE-M3 인코더의 마지막 은닉층, H'은 Sentence-level BERT 인코더의 마지막 은닉층이다. $\mathcal{D}_q$를 Sentence-level BERT의 입력으로 사용하기 위해 두 인코더 (XLM-RoBERTa, Sentence-level BERT)의 모델 사이즈는 동일하다.

- **Loss Function for Self-Knowledge Distillation**

Loss function은 BGE-M3(Chen et al., 2024)의 방식을 그대로 따르되, score가 하나 더 늘었기 때문에 이를 처리하기 위한 부분만 일부 수정하여 적용하였다.

각 점수들의 가중합을 integrated score로 사용한다:

$$s_{\text{inter}} \leftarrow w_1 \cdot s_{\text{dense}} + w_2 \cdot s_{\text{lex}} + w_3 \cdot s_{\text{mul}} + w_4 \cdot s_{\text{sent}} \tag{10}$$

손실함수로는 infoNCE Loss를 사용한다:

$$\mathcal{L}_{s(\cdot)} = -\log \frac{\exp\left(\frac{s(q,p^*)}{\tau}\right)}{\sum_{p \in \{p^*, p'\}} \exp\left(\frac{s(q,p)}{\tau}\right)} \tag{11}$$

p^*와 p'는 각각 쿼리 q에 대한 양성샘플과 음성샘플이다. $s_*(\cdot)$는 $s_{\text{dense}}(\cdot)$, $s_{\text{lex}}(\cdot)$, $s_{\text{mul}}(\cdot)$, $s_{\text{sent}}(\cdot)$ 중 임의의 함수이다.

그리고 $\mathcal{L}_*$의 가중합으로 self-knowledge distillation이 적용되지않은 loss인 $\mathcal{L}$를 구한다:

$$\mathcal{L} \leftarrow \frac{1}{5}(\lambda_1 \cdot \mathcal{L}_{\text{dense}} + \lambda_2 \cdot \mathcal{L}_{\text{lex}} + \lambda_3 \cdot \mathcal{L}_{\text{mul}} + \lambda_4 \cdot \mathcal{L}_{\text{sent}} + \lambda_5 \cdot \mathcal{L}_{\text{inter}}) \tag{12}$$

teacher score로 이전에 구했던 integrated score s_{inter}를 사용해 self-knowledge distillation이 적용된 손실인 $\mathcal{L}'_*$를 계산한다:

$$\mathcal{L}'_* \leftarrow -\,\text{softmax}(s_{\text{inter}}) \times \log\left(\text{softmax}(s_*)\right) \tag{13}$$

$\mathcal{L}'_*$의 가중합으로 $\mathcal{L}'$를 계산한다:

$$\mathcal{L}' \leftarrow \frac{1}{4}(\lambda_1 \cdot \mathcal{L}'_{\text{dense}} + \lambda_2 \cdot \mathcal{L}'_{\text{lex}} + \lambda_3 \cdot \mathcal{L}'_{\text{mul}} + \lambda_4 \cdot \mathcal{L}'_{\text{sent}}) \tag{14}$$

마지막으로, $\mathcal{L}$과 $\mathcal{L}'$의 선형결합으로 최종 손실함수 L_{final}를 얻는다:

$$\mathcal{L}_{\text{final}} = (\mathcal{L} + \mathcal{L}')/2 \tag{15}$$

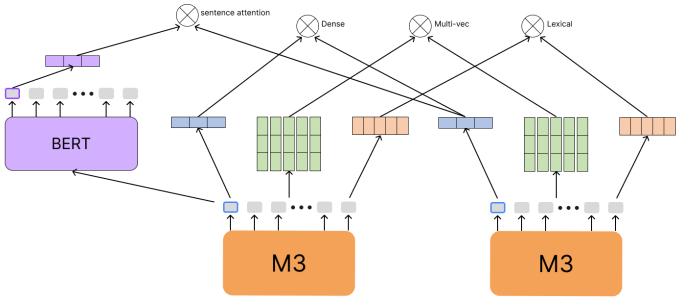

Figure 2: BGEM3-Ctrl

4.2 대조군 모델 (BGEM3-Ctrl)

SAQ 방식과 단순히 인코더 층을 더 추가시킨 방식의 성능을 비교하고자 하였고, 이를 위한 대조군 모델로 BGEM3-Ctrl를 준비하였다. Sentence-level BERT를 BERT로 대체한 것 외에는 BGEM3-SAQ와 동일하다. 쿼리들의 M3 인코더 last hidden state가 인코더 층을 추가적으로 더 통과하도록 구성하였다. 추가된 BERT의 last hidden state에서의 [CLS]토큰벡터를 e'_q라 할 때, sentence attention score는 e'_q와 e_p의 내적으로 계산한다:

$$e_p = \text{norm}(H_p[0]), \quad H'_q = BERT(H_q). \tag{16}$$

$$e'_q = \text{norm}(H'_q[0]), \quad s_{sent} \leftarrow \langle e'_q, e_p. \rangle. \tag{17}$$

5 모델 평가

각 모델을 파인튜닝하는 과정에서 사용된 사전학습된 모델은 dragonkue/bge-m3-ko를 사용하였다. 모델 테스트 결과1를 봤을 때, BGEM3-SAQ가 모든 평가지표에서 근소하게 세 모델 중 가장 우위를 점하였다. SAQ 모델이 기본 M3 모델에 비해 인코딩 층의 수가 더 많다는 점을 생각하면 당연한 결과라고 생각된다. 반면, 추가된 인코더 레이어 갯수가 8개로 SAQ와 동일하나 추가된 방식이 다른 Ctrl 모델의 지표를 보면 전반적으로 봤을 때, 기본 M3 모델보다는 근소하게 우위이고, SAQ 모델보다는 근소하게 열세이다. 다만 MRR@10상에서는 Ctrl 모델이 세 모델 중 가장 떨어지는 수치를 보여주고 있다. 결론적으로 해당 실험 결과는 문장 단위에서의 셀프 어탠션을 쿼리에 수행하여 생성한 임베딩을 학습에 활용하는 방식이(SAQ) 단순히 쿼리 인코더 임베딩 층의 수를 늘리는 방식보다(Ctrl) 더 효과적일 수 있음을 보여주고 있다.

6 한계

모델 학습에 사용한 쿼리(형법 조항)들은 '제-조의 예에 따른다.', '전항의 죄를 상습적으로 범한'등의 외부 참조 표현에서 발생하는 정보의 공백을 일일이 사람이 채워

Table 1: Evaluation Results

We use dragonkue/bge-m3-ko as a shared pre-trained encoder across all models.

Model	**NDCG@10**	**MAP@10**	**Recall@10**	**MRR@10**
BGEM3(pre-trained)	0.387	0.263	0.362	0.582
BGEM3	0.598	0.460	0.545	0.845
SAQ	0.609	0.472	0.555	0.870
Ctrl	0.597	0.467	0.554	0.800

넣는 방식을 통해 만들어졌다. 이 과정에서 사람의 재량적인 판단에 의존하여, 가급적 본래 의미를 훼손하지 않는 선에서 데이터를 가공하는 방식을 사용하고 있기에 쿼리를 가공하는 사람의 법리적 이해도에 따라 쿼리의 데이터 품질 또한 달라진. 그리고 쿼리 가공에 있어 사람의 개입이 필수적인 방식이기에, 법령이 개정되거나 추가될 때마다 모델에 반영시키는 과정이 번거롭다.

패시지의 경우 사람이 쿼리와 연관된 문장을 분류하는 것이 아닌, 기계가 문장을 생성하고 생성한 문장을 패시지로 사용하는 방식을 사용하였다. 패시지 문장의 경우 "피고인은 위험물질이 담긴 공업용 드럼통을 안전장치 없이 가열하다 드럼통이 터져 인근 상가 건물 및 차량에 심각한 재산상 피해를 입혔다."처럼 핵심 내용을 벗어난 부가적인 정보가 비교적 없어 실사용 데이터와는 다소 차이가 있다. 가령 KLAID에서는 "피고인은 2018. 8. 3. 06:30경 경기 파주시 B에 있는 'C 주점' 앞길에서, 폭행 사건이 발생했다는 내용의 112신고를 받고 출동한 파주경찰서 D 파출소 소속 순경..."처럼 부가적인 정보가 오히려 핵심적인 정보보다 더 많다. 또한 Synthetic passage 중 노이즈 데이터를 거르지 않고 그대로 사용했기 때문에 해당 부분도 개선의 여지가 충분하다 (Bonifacio et al., 2022).

References

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Data augmentation for information retrieval using large language models, 2022. URL `https://arxiv.org/abs/2202.05144`.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. URL `https://arxiv.org/abs/2402.03216`.

Jinghui Lu, Maeve Henchion, Ivan Bacher, and Brian Mac Namee. A sentence-level hierarchical bert model for document classification with limited labelled data, 2021. URL `https://arxiv.org/abs/2106.06738`.

Arpan Mandal, Kripabandhu Ghosh, Arnab Bhattacharya, Arindam Pal, and Saptarshi Ghosh. Overview of the fire 2017 irled track: Information retrieval from legal documents. In *FIRE (Working Notes)*, pages 63–68, 2017.

OECD. *Government at a Glance 2021*. OECD Publishing, Paris, 2021. doi: 10.1787/1c258f55-en. URL https://doi.org/10.1787/1c258f55-en.