Journal of KIIT. Vol. 24, No. 6, pp. 79-86, Jun. 30, 2026. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2026.24.6.79

구문 수준 임베딩 확장을 통한 멀티벡터 검색 성능 향상

윤 보 현*

Enhancing Multi-Vector Retrieval Performance through Phrase-Level Embedding Expansion

Bo-Hyun Yun*

요 약

멀티벡터 계열의 검색 모델은 용어 불일치를 완화하고 의미 기반의 정밀 검색이 가능하다. 토큰 표현 구조와 토큰 수준 상호작용의 모델링 방식은 검색 성능에 직접적인 영향을 미친다. 본 논문은 기존 ColBERT의 지연 상호작용 점수 계산방법을 유지한 채, 문서 표현을 구문 단위로 확장하는 방법을 제안한다. 인코더 출력의 유효 시퀀스에 슬라이딩 윈도우를 적용하고 각 윈도우를 풀링으로 요약한 phrase-token을 생성하여, 로컬 토큰 임베딩과 색인 단계에서 결합하는것이 특징이다. 제안법은 추가 파라미터 없이 멀티 토큰에 분산된 구문 의미를 보완하며, 문서당 소수의 추가 벡터만으로 성능향상을 기대할 수 있다. 실험 데이터는 한국어 위키피디아를 사용하였다. 실험 결과 본 방법은 R@1에서 베이스라인 성능인 0.671에서 0.705로 향상된 성능을 나타냈고, 다양한 지표 환경에서도 일관된 성능을 향상을 보였다. 아울러 phrase-token 생성 시 풀링 연산(Mean, Max, Attention) 간 비교를 통해 적합한 풀링 방법을 분석한다.

Abstract

Multi-vector retrieval models mitigate term mismatch and enable semantically precise retrieval. The design of token representations and token-level late interaction mechanisms directly affects retrieval performance. This paper proposes a method that extends document representations to the phrase level while preserving ColBERT's late interaction mechanism. Specifically, a sliding window is applied to the valid sequence of encoder outputs, and each window is summarized via pooling to generate phrase-tokens, which are then integrated with local token embeddings during the indexing stage. The proposed method enhances phrase-level semantics dispersed across multiple tokens without additional parameters, and improves retrieval performance with only a few extra vectors per document. Experimental data used Korean Wikipedia. Experimental results show that the proposed method improves Recall@1 from the baseline score of 0.671 to 0.705 and consistently enhances performance across various evaluation metrics. We further conduct a comparison of pooling operations(Mean, Max, Attention) for phrase token generation to determine the most effective configuration.

Keywords

information retrieval, deep learning, multimodal retrieval, question answering system

* 목원대학교 소프트웨어교양학부 교수
- ORCID: https://orcid.org/0000-0003-0544-3807

· Received: Mar. 24, 2026, Revised: Apr. 16, 2026, Accepted: Apr. 19, 2026

· Corresponding Author: Bo-Hyun Yun
35349 88, Doanbuk-ro, Seo-gu, Daejeon, Republic of Korea
Tel.: +82-42-829-7642, Email: ybh@mokwon.ac.kr

Ⅰ. 서 론

대규모 코퍼스에서 관련 단락을 정확하게 찾는 초기 단계 검색은 질의응답, 지식 증강 생성 시스템 등 다양한 응용의 핵심 구성 요소이다. 최근에는 멀티벡터 기반 표현이 단일 벡터보다 우수한 검색 성능을 보이며, 특히 ColBERT 계열은 질의-문서 간 토큰 단위 지연 상호작용을 통해 높은 정확도를 달성하였다. 그러나 ColBERT의 스코어 계산 방식은 질의의 각 토큰이 문서의 단일 토큰과만 매칭되도록 설계되어 있어, 멀티토큰 구문이나 엔티티가 여러 토큰으로 분산된 경우 이를 충분히 반영하기 어렵다. 기존 연구들은 이러한 한계를 완화하기 위해 표현 압축 및 효율성 개선(ColBERTv2), 희소 표현(SPLADE), 토큰 중요도 반영 등의 접근을 제안하였다. 그러나 이러한 방법들은 토큰 단위 상호작용 구조를 유지하고 있어 다중 토큰으로 구성된 의미 단위를 직접적으로 모델링하는 데에는 여전히 한계가 존재한다. 특히, 구문이나 엔티티와 같이 여러 토큰에 걸쳐 표현되는 의미는 개별 토큰 단위의 최대 유사도만으로는 충분히 반영되기 어렵다.

본 논문은 이러한 문제를 완화하기 위해 문서 표현을 구문 수준으로 확장하는 방법을 제안한다. 구체적으로, 백본 인코더 출력의 유효 시퀀스에 슬라이딩 윈도우를 적용하고 각 윈도우를 풀링하여 구문 벡터를 생성한다. 생성된 구문 벡터는 기존 토큰 임베딩과 동일한 수준으로 색인되어, 질의 토큰이 개별 토큰뿐 아니라 구문 단위 표현과도 매칭될 수 있도록 한다. 이를 통해 기존 지연 상호작용 구조를 유지하면서 다중 토큰 의미를 효과적으로 반영할 수 있다. 실험 결과, 제안한 방법은 기존 대비 R@1에서 최대 0.034의 성능 향상을 보였으며, 색인 대상 확장만으로도 검색 성능 개선이 가능함을 확인하였다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고, 3장에서는 제안 방법을 설명한다. 4장에서는 실험 환경과 결과를 제시하며, 5장에서는 결론을 제시한다.

Ⅱ. 관련 연구

딥러닝 기반 정보 검색에서는 질의와 문서를 독립적으로 임베딩한 뒤 유사도를 계산하는 듀얼 인코더 기반 밀집 검색(Dense retrieval)이 널리 사용되어 왔다. DPR[1]은 질의와 단락을 각각 임베딩한 후 내적 유사도를 기반으로 검색을 수행하여 전통적인 키워드 기반 검색 대비 우수한 성능을 보였다. 또한 PAIR[2]는 단락 중심의 유사도 관계를 명시적으로 모델링하여 듀얼 인코더의 표현력을 향상시키는 방법을 제안하였다. 그러나 이러한 단일 벡터 기반 접근은 질의와 문서의 다양한 의미를 하나의 벡터로 압축하기 때문에 세밀한 의미 표현이 손실되는 한계가 존재한다. 이러한 한계를 극복하기 위해 문서를 다수의 벡터로 표현하는 멀티벡터 기반 접근이 제안되었다. 대표적으로 ColBERT[3]는 문서를 토큰 단위로 표현하고, 질의-문서 간 토큰 단위 유사도를 계산하는 지연 상호작용(Late interaction) 구조를 도입하여 정확도와 효율성을 동시에 향상시켰다. COIL[4]은 토큰 수준 표현을 유지하면서 역색인 구조를 활용하여 어휘 일치 신호를 강화하는 접근을 제시하였다.

ColBERT 이후, 지연 상호작용 구조를 기반으로 다양한 후속 연구들이 제안되었다. ColBERTv2[5]는 잔차 기반 압축과 잡음 완화 학습을 도입하여 메모리 효율성과 검색 성능을 동시에 개선하였다. 또한 ColBERTer[6]는 문서 표현을 단어 단위로 압축하여 토큰 수를 줄이는 방식으로 효율성을 향상시키고자 하였다. 한편, SPLADE[7] 및 SLIM 계열 연구는 희소 표현(Sparse representation)을 활용하여 멀티벡터 표현의 효율성을 개선하는 방향을 제시하였다. 최근에는 Jina-ColBERT-v2[8]와 같이 다국어 및 장문 문서 처리를 지원하도록 확장된 모델도 제안되고 있다. 그러나 이러한 연구들은 주로 표현의 압축이나 효율성 개선에 초점을 맞추고 있으며, 토큰 단위 상호작용 구조 자체는 유지하고 있다. 그러나 ColBERT의 MaxSim 기반 점수 계산 방식은 각 질의 토큰이 문서 내 단일 토큰과의 최대 유사도만을 선택하는 구조를 갖는다. 이러한 토큰 단위 매칭(Token-wise matching)은 효율적인 계산을 가능하게 하지만, 다중 토큰으로 구성된 의미 단위를 직접적으로 모델링하는 데에는 한계가 있다. 예를 들어,

특정 구문이나 개체(Entity)의 의미는 여러 토큰에 분산되어 표현되며, 이를 개별 토큰의 최대 유사도로만 복원하기는 어렵다. 또한 토큰 간의 상호작용이나 순서 정보가 점수 계산에 직접 반영되지 않기 때문에 구성적 의미(Compositional semantics)를 충분히 반영하지 못하는 문제가 존재한다.

한편, 검색 모델의 표현력을 향상시키기 위한 사전학습 기반 접근도 제안되었다. MAE 계열[9-11]은 마스킹된 입력을 복원하는 방식으로 검색 지향 표현을 학습하며, 최근에는 디코더를 제거하고 인코더 중심으로 표현을 학습하는 방식[12]도 연구되었다. 또한 실제 검색 시스템에서는 1단계 검색 이후 재순위화 과정을 통해 성능을 향상시키며, 크로스 인코더 기반 방법[13]이나 문서 구조를 활용한 재순위화 기법[14][15]이 널리 사용된다. 그러나 이러한 접근은 계산 비용이 증가하거나 별도의 후처리 단계를 필요로 한다는 한계가 있다.

본 연구는 이러한 한계를 완화하기 위해, 기존 ColBERT의 지연 상호작용 구조를 유지하면서 문서 표현을 구문 수준으로 확장하는 방법을 제안한다. 슬라이딩 윈도우 기반 풀링을 통해 생성된 phrase-level 임베딩을 문서 토큰 집합에 추가함으로써, 질의 토큰이 개별 토큰뿐 아니라 구문 단위 표현과도 매칭될 수 있도록 하여 다중 토큰 의미를 효과적으로 반영한다.

III. 제안 방법

3.1 멀티벡터 기반 표현 모델 구조

ColBERT는 질의-문서 간 유사도를 토큰 단위로 계산하는 지연 상호작용 모델이다. 백본 인코더로부터 컨텍스트 임베딩을 얻은 뒤, 선형 레이어와 L2 정규화를 거쳐 질문과 문서를 표현하는 최종 임베딩을 생성한다. $Q \in R^{|q| \times d}$와 $L \in R^{T \times d}$는 질문 임베딩과 문서 임베딩 결과를 의미한다.

검색시 점수는 질의 토큰별 최대 유사도 합(Token-wise MaxSim)으로 정의하며, 수식은 식 (1)과 같다.

$$s(q,d) = \sum_{i=1}^{|q|} \max_{i \le j \le T} (Q_i, L_j) \tag{1}$$

그러나, ColBERT의 검색 스코어 계산 방법은 질의 토큰 Q_i는 문서의 단일 토큰 L_j만 매칭이 된다. 이때 다음과 같은 한계가 존재한다. 첫 번째는 구절·멀티토큰 엔티티의 분산 문제이다. "전자상거래 반품 정책"과 같은 구문은 의미가 여러 토큰에 분할되어 있다. 각 토큰은 개별 의미만 담고, 구절 전체의 의미는 토큰의 최대 유사도 값을 찾는 방법으로 복원되기 어렵다. 두 번째는 구성적 의미(Compositionality) 손실이 발생한다. 토큰 간 상호작용이 점수식에 직접 모델링되지 않아, 토큰 순서에 대한 고려나 동시 출현과 같은 시그널이 약하다. 이러한 제약을 완화하기 위해 본 연구에서는 문서 내 구문 정보를 소수의 요약 벡터로 응축해 기존 토큰 집합에 저비용으로 주입하는 방법을 제안한다.

3.2 제안방법

본 절에서는 ColBERT의 스코어 계산 방법인 지연 상호작용 연산을 변경하지 않고, 문서 표현을 구문 수준으로 확장하는 방법을 제안한다. 핵심 아이디어는 인코더의 컨텍스트 임베딩 위에서 윈도우 요약 연산으로 소수의 phrase-token을 생성하고, 이를 기존의 문서 토큰 집합 L에 결합하는 것이고, 그림 1을 통해 확인할 수 있다. 우선 백본 인코더 출력 $H \in R^{T \times h}$에 대해, 스페셜 토큰인 [CLS], 문서 마커인 [D], [SEP] 및 패딩 토큰을 제거하는 바이너리 마스크 $M \in \{0,1\}^T$를 적용하여, 유효 시퀀스 $C = \{H_i | M_i = 1\} \in R^{L \times h}$를 구한다. 이후 C를 기반으로 슬라이딩 윈도우를 구성하고, 각 윈도우를 하나의 phrase-token 형태의 구문 임베딩으로 변환하여 활용한다. 식 (2)는 이러한 윈도우들의 집합을 나타낸다.

$$W = \left\{ X = C_{t:t+w} \in R^{w \times h} \,|\, t \in \{1, 1+s, 1+2s, \ldots\}, t+w-1 \le l \right\} \tag{2}$$

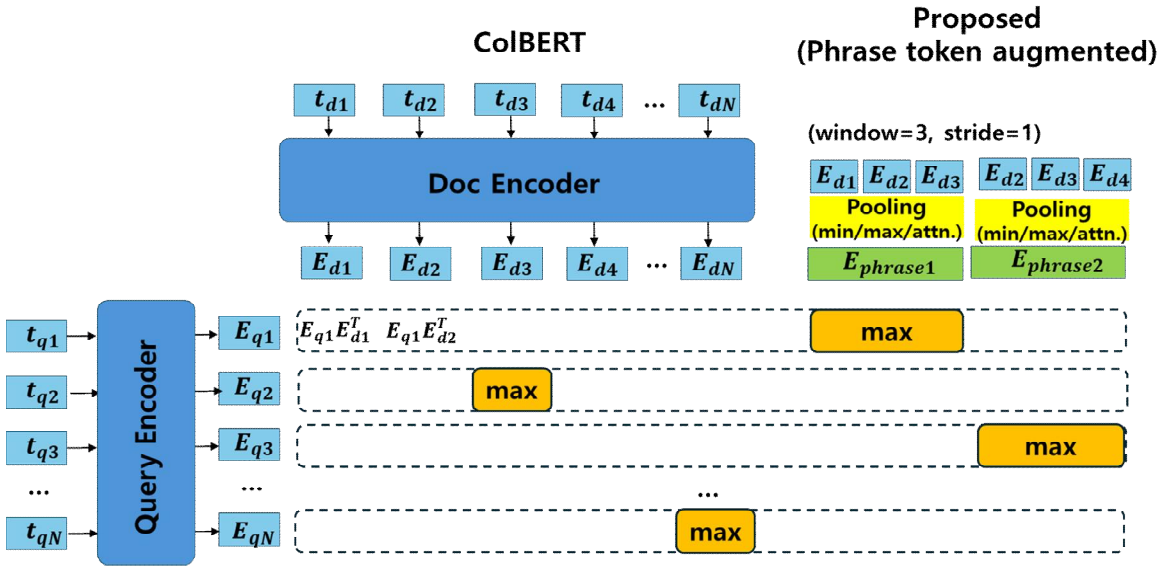


그림 1. 지연 상호작용 구조를 유지하면서, 풀링을 통해 생성된 구문 수준 임베딩을 문서 표현에 추가하여, 질의 토큰은 개별 토큰뿐 아니라 구문 단위 표현과도 MaxSim 연산

Fig. 1. Maintaining the late interaction structure, phrase-level embeddings from pooling are added to the document, enabling query tokens to match both token- and phrase-level representations

W는 슬라이딩 윈도우로 잘라 낸 모든 윈도우 조각들의 집합을 의미하고, X는 집합 W의 개별 윈도우 행렬이다. w는 윈도우 길이, s는 스트라이드, l은 유효 시퀀스의 길이를 의미한다.

우리는 각 윈도우 X를 구문 수준의 phrase-token으로 요약하기 위해 다수 개의 토큰을 하나의 임베딩으로 투사하는 경량 풀링 연산을 수행하였고, 최적의 결합 방법을 산정하기 위해 아래와 같이 세 가지 변형 실험을 수행하였다.

첫 번째는 Mean pooling 방법이다. 윈도우 내 토큰들을 균등 가중치로 평균하여 구문 전체를 요약하는 방법이다. 특정 토큰의 큰 변화가 평균에 완만하게 반영되어 노이즈에 안정적이고 단순하지만, 일관성을 가질수 있는 방법이다. 수식은 식 (3)과 같다.

$$\Pi_{mean}(X) = \frac{1}{w}\sum_{k=1}^{w} X_k \tag{3}$$

두 번째는 Max pooling 방법이다. 차원별로 큰 활성화 값만 사용하기 때문에 선택적 요약에 가깝다. 구문 내 핵심 토큰이 분명한 경우 효율성이 높으며, ColBERT의 유사도 계산인 MaxSim과 상호보완적인 연산이 가능하다. 수식은 식 (4)와 같다.

$$\left(\Pi_{\max}(X)\right)_d = \max_{1 \le k \le w} (X_k)_d, \; d = 1,..,h \tag{4}$$

세 번째는 Attention pooling 방법이다. 윈도우 내의 토큰들의 내용 유사도에 따라 가중 분포를 산출하고, 이에 기반한 가중 평균으로 구문 벡터를 생성하는 방법이다. 앞선 두 가지 Mean/Max pooling 접근과 달리 연속 스펙트럼을 형성하기 때문에 도메인 적응력이 높으며, 수식은 식 (5)와 같다.

$$\Pi_{attn}(X) = \sum_{k=1}^{w} a_k X_k \tag{5}$$
$$a = softmax\left(\frac{Xq^T}{\sqrt{h}}\right)$$
$$q = \frac{1}{w}\sum_{k=1}^{w} X_k$$

세 방식의 출력 벡터는 공유하는 선형 레이어와 L2 정규화를 거쳐 문서 토큰과 동일 임베딩 공간에서 비교 실험이 가능하다. 우리는 추가적인 모델의 구조 변경 없이 색인 단계의 결합만으로 구문 수준의 토큰을 추가하였으며, 다음 장에서 제안 방법의 우수성을 평가한다.

Ⅳ. 평가 결과

4.1 실험 환경

코퍼스는 한국어 위키피디아를 사용하였다. 위키피디아의 각 페이지의 소제목 및 줄 바꿈을 기준으로 전처리하였고, 전체 8,2347,666건의 단락 컬렉션을 구성하였다.

학습용 질의-긍정 단락 데이터는 단락을 임의 추출한 뒤 Qwen2.5-14B-Instruct모델에 프롬프트를 부여하여 해당 단락을 정답으로 하는 질문-단락 쌍을 생성하였고, 샘플 검수를 통해 부적합한 데이터는 제외하였다. 최종적으로 질의-긍정 단락 쌍 30,000건과 각 질문에 대해 컬렉션에서 무작위로 선택한 부정 단락을 4개씩 선택하여 총 120,000건의 부정 데이터를 구성함으로써, 1:4 비율의 긍정-부정 학습 데이터 셋을 구성하였다.

학습시 사용된 백본 인코더는 다국어 BERT모델인 bert-base-multilingual-cased를 사용하였고, 인코더 출력 결과를 선형 레이어와 L2 정규화를 통해 128차원의 임베딩을 생성하였다. 문서는 특수 토큰으로 사용되는 [D], [CLS], [SEP]를 포함하여 최대 512토큰을 사용하였으며, 색인시에는 특수 토큰과 패딩 및 구두점은 바이너리 마스크로 제외하였다.

Phrase-token 연산에 사용되는 윈도우 하이퍼 파라미터는 식 (6)과 같다.

$$(w,s,K_{\max}) \in \{(10,5,24),(20,10,24),(30,15,24),(40,20,24)\} \quad (6)$$

여기서 w는 윈도우 길이, s는 스트라이드, $K_{\max}$는 문서당 phrase-token 최대 개수를 의미하고, 본 실험에서는 최대 윈도우 길이일 때를 고려하여 24개로 설정하였다. 생성된 구문 수준 벡터는 로컬 토큰 임베딩과 결합하여 색인하여 사용하였다.

본 실험에서는 phrase-token이 인접 구간의 문맥 정보를 중첩하여 반영하도록 하기 위해 스트라이드를 윈도우 길이의 절반으로 설정하였다. 따라서 본 결과는 윈도우 길이와 스트라이드의 영향을 완전히 분리한 분석이라기보다는, 해당 설정에서의 비교 결과로 해석하는 것이 적절하다. 또한 본 연구에서 phrase-token은 언어학적 구와 정확히 일치하는 단위라기보다는, 연속된 토큰 구간으로 구성된 국소 의미 단위를 의미한다.

검색 성능 평가는 2,000개의 질문을 사용하여 평가하였으며, 평가 방법으로는 Recall@TopK 지표를 사용하여 검색 성능을 비교 평가하였다.

4.2 실험 결과

표 1은 기존 베이스라인 모델인 ColBERT 검색의 성능과 제안한 세 가지 phrase-token의 풀링 방식들을 비교한 결과이다. 볼드체는 실험 환경에서 가장 높은 성능을 나타내며, 밑줄이 있는 성능은 두 번째로 높은 성능을 의미한다.

R@1 기준으로 baseline은 0.6710인 반면, mean, max, attention 풀링의 최대 성능은 각각 0.6945, 0.7050, 0.6875를 나타냈다. 성능 개선 폭은 +0.0165~+0.0340로, phrase-token을 색인 수준에서 추가하는 것만으로도 상위 랭크의 검색 품질이 유의미하게 향상됨을 확인하였다.

표 1. 풀링 방식 및 윈도우/스트라이드 설정에 따른 검색 성능 비교
Table 1. Comparison of retrieval performance across pooling methods and window–stride configurations

Pooling method	Baseline	Mean pooling				Max pooling				Attention pooling			
w(window), s(stride)	–	w=10, s=5	w=20, s=10	w=30, s=15	w=40, s=20	w=10, s=5	w=20, s=10	w=30, s=15	w=40, s=20	w=10, s=5	w=20, s=10	w=30, s=15	w=40, s=20
R@1	0.6710	0.6785	0.6730	0.6875	0.6945	0.6955	0.6870	0.6955	**0.7050**	0.6745	0.6645	0.6875	0.6875
R@5	0.8685	0.8665	0.8635	0.8655	0.8675	0.8710	0.8745	0.8780	**0.8785**	0.8655	0.8650	0.8700	0.8695
R@50	0.9460	0.9515	0.9520	0.9520	0.9520	0.9550	0.9545	**0.9565**	0.9555	0.9535	0.9530	0.9515	0.9525
R@100	0.9575	0.9625	0.9650	0.9650	0.9645	0.9650	0.9660	0.9650	0.9650	0.9650	**0.9665**	0.9660	0.9650

R@5에서는 mean이 소폭 하락하였으나, max와 attention은 각각 0.8785, 0.8700로 개선된 것을 확인하였다. R@1과 R@5는 사용자 관점에서 검색 순위 품질을 반영하는 지표로, 다중 토큰에 분산된 구문 정보를 소수의 phrase-token으로 응축하여 검색 성능 향상에 효과적인 것을 확인할 수 있었다. 개별 풀링 방법에 따라 성능 분석 결과는 아래와 같다.

Mean 풀링은 윈도우 크기가 증가될수록 안정성 면에서 일관된 성능 향상을 보였다. R@1 최고 성능은 0.6945로 베이스라인보다 높은 성능을 보였지만, R@5에서는 향상 폭이 제한적이거나 일부 설정에서는 미세하게 하락하였다.

Max 풀링은 전반적으로 가장 안정적이고 높은 성능 향상을 보였다. 윈도우 크기가 40일때 0.7050으로 최고 성능을 나타냈고, R@5에서도 0.8785로 최대 개선을 보였다. 차원별 최대치 선택이 phrase-token의 핵심 단서를 효과적으로 포착하며, 유사도 계산인 MaxSim과 구조적으로도 맞는 방법으로 분석되었다.

Attention 풀링은 R@1 및 R@5 기준에서 baseline 대비 개선 폭이 제한적이거나 일부 설정에서 성능 변동이 나타났다. 이는 윈도우 내 토큰들에 대해 가중 평균을 수행하는 특성으로 인해, 특정 핵심 토큰의 강한 신호가 상대적으로 희석될 수 있기 때문으로 해석된다. 반면, 윈도우 크기 20에서는 국소 문맥과 구문 수준 의미를 비교적 균형 있게 반영할 수 있어 상대적으로 우수한 성능을 보였다. 이는 Attention pooling이 윈도우 범위에 민감하게 반응하는 특성을 가지며, 윈도우가 지나치게 작을 경우 phrase-token의 의미 확장 효과가 제한되고, 반대로 지나치게 클 경우 불필요한 토큰까지 함께 반영되어 중요도 분포가 분산되기 때문으로 볼 수 있다. 따라서 본 실험에서 관찰된 Attention pooling의 성능 변동은 가중 평균 기반 표현의 특성과 윈도우 크기 설정에 대한 민감도에서 기인한 것으로 분석된다.

전반적인 실험 결과를 종합하면, phrase-token의 성능 향상은 기존 토큰 단위 매칭에서 분산되어 반영되던 다중 토큰 의미를 phrase-token을 통해 하나의 표현으로 집약함으로써, MaxSim 연산에서 baseline 대비 더 높은 유사도를 갖는 매칭 후보를 제공하기 때문으로 해석된다. 또한 제안 방법은 phrase-token을 추가함에 따라 문서당 벡터 수가 증가하지만, 생성 개수를 제한함으로써 추가 비용은 선형적 수준으로 유지된다. 본 실험은 윈도우 크기와 pooling 방식에 따른 성능 변화를 통해 phrase-token의 효과를 간접적으로 분석한 결과이며, 보다 세밀한 분석 실험은 향후 연구에서 추가적으로 수행할 필요가 있다.

V. 결론 및 향후 과제

본 연구에서는 ColBERT 계열 모델의 한계인 토큰 단위 매칭 제약을 완화하기 위해, 문서 표현을 구문 수준으로 확장하는 방법을 제안하였다. 인코더 출력의 유효 시퀀스에 슬라이딩 윈도우를 적용하고, 각 윈도우를 다양한 풀링 연산으로 요약한 phrase-token을 생성하여 기존 토큰 임베딩과 함께 색인에 활용하였다. 실험 결과, 제안한 방법은 모델 구조나 학습 절차의 변경 없이도 단순한 색인 확장만으로 R@1 기준 최대 0.034의 성능 향상을 보였다. 특히, Max 풀링 기반 phrase-token이 상위 랭크의 검색 품질 개선에 가장 효과적이었으며, Mean 풀링은 안정적 성능을, Attention 풀링은 완만한 가중 분포가 필요한 경우에 이점을 보였다. 이러한 결과는 멀티 토큰에 분산된 구문 의미를 저비용으로 보완함으로써, 토큰 단위 지연 상호작용 기반 검색 모델에서도 구문 수준 의미 표현이 성능 향상에 기여할 수 있음을 입증하였다.

향후 연구에서는 phrase-token의 동적 선택 전략이나 학습 가능한 가중 풀링 구조를 결합하여, 질의와 문서 간 의미적 정합성을 더욱 정교하게 반영하는 방향으로 확장할 수 있을 것이다. 또한 본 연구에서는 Recall@TopK 지표를 중심으로 성능을 평가하였으나, 향후에는 MRR, nDCG 등 다양한 순위 기반 평가 지표를 추가적으로 도입하여, 풀링 방식에 따른 성능 특성을 보다 다각적으로 분석할 계획이다.

References

[1] V. Karpukhin, B. Oguz, S. Min, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense Passage Retrieval for Open-Domain Question Answering", Proc. of the

EMNLP 2020, Online, pp. 6769-6781. Nov. 2020. https://doi.org/10.48550/arXiv.2004.04906.

[2] R. Ren, S. Lv, Y. Qu, J. Liu, W. Zhao, Q. She, H. Wu, H. Wang, and J. Wen, "PAIR: Leveraging Passage-Centric Similarity Relation for Improving Dense Passage Retrieval", Proc. of the ACL-IJCNLP 2021, Bangkok, Thailand, pp. 2173-2183, Jan. 2021. https://doi.org/10.18653/v1/2021.findings-acl.191.

[3] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT", Proc. of the SIGIR 2020, Xi'an, China, pp. 39-48, Jul. 2020. https://doi.org/10.1145/3397271.3401075.

[4] L. Gao, Z. Dai, and J. Callan, "COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List", Proc. of the NAACL 2021, Online, pp. 3030-3042, Jun. 2021. https://doi.org/10.18653/v1/2021.naacl-main.241.

[5] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia, "ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction", Proc. of the NAACL 2022, Seattle, United States, pp. 3715-3734, Jul. 2022. https://doi.org/10.18653/v1/2022.naacl-main.272.

[6] S. Hofstätter, O. Khattab, S. Althammer, M. Sertkan, and A. Hanbury, "Introducing Neural Bag of Whole-Words with ColBERTer: Contextualized Late Interactions using Enhanced Reduction", arXiv preprint, arXiv:2203.13088, Mar. 2022. https://doi.org/10.48550/arXiv.2203.13088.

[7] T. Formal, B. Piwowarski, and S. Clinchant, "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking", Proc. of the SIGIR 2021, Online, pp. 2288-2292, Jul. 2021. https://doi.org/10.1145/3404835.3463098.

[8] R. Jha, B. Wang, M. Günther, G. Mastrapas, S. Sturua, I. Mohr, A. Koukounas, M. K. Akram, N. Wang, and H. Xiao, "Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever", Proc. of the MRL 2024, Miami, Florida, USA, pp. 159-166, Nov. 2024. https://doi.org/10.48550/arXiv.2408.16672.

[9] Shuqi Lu, et al., "Less is More: Pretrain a Strong Siamese Encoder for Dense Text Retrieval Using a Weak Decoder", Proc. of the EMNLP 2021, Punta Cana, Dominican Republic, pp. 2780-2791, Nov. 2021. https://doi.org/10.18653/v1/2021.emnlp-main.220.

[10] S. Xiao, Z. Liu, Y. Shao, and Z. Cao, "RetroMAE: Pre-training Retrieval-oriented Transformers via Masked Auto-Encoder", Proc. of the EMNLP 2022, Abu Dhabi, United Arab Emirates, pp. 538-548, Dec. 2022. https://doi.org/10.18653/v1/2022.emnlp-main.35.

[11] Z. Liu, S. Xiao, Y. Shao, and Z. Cao, "RetroMAE-2: Duplex Masked Auto-Encoder For Pre-Training Retrieval-Oriented Language Models", Proc. of the ACL 2023, Toronto, Canada, pp. 2635-2648, Jul. 2023. https://doi.org/10.48550/arXiv.2305.02564.

[12] G. Ma, X. Wu, Z. Lin, and S. Hu, "Drop your Decoder: Pre-training with Bag-of-Word Prediction for Dense Passage Retrieval", Proc. of the SIGIR 2024, Washington D.C., USA, pp. 1818-1827, Jul. 2024. https://doi.org/10.1145/3626772.3657792.

[13] R. Nogueira and K. Cho, "Passage Re-ranking with BERT", arXiv preprint arXiv:1901.04085, Apr. 2020. https://doi.org/10.48550/arXiv.1901.04085.

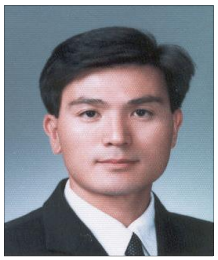
[14] L. Canjia, Y. Andrew, M. Sean, H. Ben, and S. Yingfei, "PARADE: Passage Representation Aggregation for Document Reranking", arXiv preprint arXiv:2008.09093, Jun. 2021. https://doi.org/10.48550/arXiv.2008.09093.

[15] Y. Zhang, D. Long, G. Xu, and P. Xie, "HLATR: Enhance Multi-stage Text Retrieval with Hybrid List Aware Transformer Reranking", arXiv preprint arXiv:2205.10569, May 2022. https://doi.org/10.48550/arXiv.2205.10569.

http://dx.doi.org/10.14801/jkiit.2026.24.6.87

저자소개

윤 보 현 (Bo-Hyun Yun)



1999년 8월 : 고려대학교 컴퓨터학과(이학박사)
1999년 9월 ~ 2002년 2월 : 한국전자통신 연구원 선임연구원 (팀장)
2003년 3월 ~ 현재 : 목원대학교 SW교양학부 교수
관심분야 : 인공지능, 소프트웨어 교육, 정보검색