ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction

Stanford University

NAACL 2022

ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

Omar Khattab
Stanford University
okhattab@stanford.edu

Matei Zaharia
Stanford University
matei@cs.stanford.edu

ColBERTv2:
Effective and Efficient Retrieval via Lightweight Late Interaction

Keshav Santhanam*
Stanford University

Omar Khattab*
Stanford University

Jon Saad-Falcon
Georgia Institute of Technology

Christopher Potts
Stanford University

Matei Zaharia
Stanford University

Content

1. Introduction

- Neural information retrieval (IR) has quickly dominated the search landscape over the past 2-3 years
- 많은 IR methods 은 single-vector similarity paradigm 을 따름
    - A pretrained language model is used to encode each query and each document into a single high-dimensional vector
    - Relevance is modeled as a simple dot product between both vectors
- ColBERT 는 late interaction 으로 위 과정을 대체했음
    - Queries and documents are encoded into multi-vector representations

1. Introduction

- ColBERTv2 은 token-level computations 으로 decomposing relevance modeling 함으로써, late interaction 은 인코더의 burden 을 감소시킴
- 그러나, late interaction systems 은 single-vector models 보다 더 큰 space footprint 를 차지함
    - 왜냐하면, web-scale 과 같이 billions of small vector 를 저장해야 하기 때문에
- 차라리 single-vector에 새로운 supervision paradigms (for negative mining) 를 사용하거나, pretraining, 또는 distillation 등을 사용하는 후속 연구가 진행됨 => vanilla ColBERT 만큼의 성능 또는 그보다 높게 나옴
- ColBERTv2 는 lightweight token representation 으로 효율적인 저장 공간 사용과, denoised supervision 으로 within and outside domain 모두에서 SoTA 를 달성함
- 또한 LoTTE (Long-Tail Topic-stratified Evaluation) 라는 새로운 long-tail 를 위한 데이터셋 소개

1. Introduction

- 3 Contributions
  - Propose ColBERTv2, a retriever that combines denoised supervision and residual compression, leveraging the token-level decomposition of late interaction to achieve high robustness with a reduced space footprint
  - Introduce LoTTE, a new resource for out-of-domain evaluation of retrievers. LoTTE focuses on natural information-seeking queries over long-tail topics, an important yet understudied application space
  - Evaluate ColBERTv2 across a wide range of settings, establishing state-of-the-art quality within and outside the training domain

2. Background & Related Work

- Token-Decomposed Scoring in Neural IR
  - 기존의 IR 태스크에 대한 접근은 single high-dimensional vector 로 passages 를 encoding 하는 것
  - ColBERT 는 multi-vector embedding 을 사용하여 token-level 의 계산 수행
- Vector Compression for Neural IR
  - 최근 IR 태스크에서 compressing representation 연구 활발
  - 본 연구에서, residual compression 을 사용한 late-interaction retrieval and investigate compression 에 초점
  - ColBERTv2 is the first approach to use residual compression for scalable neural IR

2. Background & Related Work

- Improving the Quality of Single-Vector Representations
  - Multi-vector representations 을 압축해서 사용하는 것보다 supervision 의 세부 정보에 매우 민감한 single-vector model 을 사용하고자 함
  - Three directions
    - Distillation of more expressive architectures including explicit denoising
    - Hard negative sampling
    - Improved pretraining
- Out-of-Domain Evaluation in IR
  - Out-of-domain 데이터셋으로 BEIR 이 있지만, 이는 natural search task 보다는 semantic relatedness tasks 에 초점이 맞춰짐
  - 새로운 LoTTE, a new dataset for out-of-domain retrieval, natural search queries over long-tail topics 제안

3. ColBERTv2

- ColBERTv2
  - Improves the quality of multi-vector retrieval models (Supervision)
  - Reducing space footprint (Representation)

- ColBERTv2 Content
  - Modeling
  - Supervision
  - Representation
  - Indexing
  - Retrieval

3. ColBERTv2 : Modeling

ColBERT 피피티 : link

- ColBERTv2 adopts the late interaction architecture of ColBERT
- Queries, passages are independently encoded with BERT, and the output embeddings encoding each token are projected to a lower dimension
- Offline indexing
    - Corpus 의 모든 passage d 는 vector 로 encoding
- Search time
    - Query q 는 multi-vector representation 으로 encoding
    - MaxSim operation

3. ColBERTv2 : Supervision

- Neural retriever (e.g. ColBERT) 를 학습하기 위해 positive, negative passages 필요
- ColBERT 는 official <q, d+, d-> triples of MS MARCO 를 사용
  - A positive d+ is human-annotated
  - Each negative d- is sampled from unannotated BM25-retrieved passages
- **Simple, uniform supervision scheme 을 만들어서 ColBERTv2 에 적용하고자 함**
  - Selects challenging negatives (d-)
  - Avoid rewarding false positives
  - Penalizing false negatives
- 이를 위해서 Relevance-guided Supervision for OpenQA with ColBERT (Khattabl et al. (2021b)) 방식으로 ColBERT 모델 학습

Relevance-guided Supervision for OpenQA with ColBERT

- OpenQA 태스크
- OpenQA 태스크는 gold-labeled passages 가 부족하고, short answer strings 을 supervision signals 로 가지고 있음
- 그래서 본 연구에서 RGS (Relevance-guided Supervision) 를 통해 supervision <q, d+, d-> 를 생성
- 핵심 가정은 effective retrieval 은 더 정확하고 다양한 positives 를, 그리고 더 어렵고 사실적인 negatives 를 collect 할 것
- 또한, RGS 는 positives 와 negatives 를 training loop 완전히 밖에서 (outside, outer-loop) 에서 수행함
- As a results, we can flexibly sample positives and negatives from large depths and we can fine-tune the entire model without having to frequently re-index the corpus

Relevance-guided Supervision for OpenQA with ColBERT

Algorithm 1: Relevance-guided Supervision, given a weak heuristic H

Input: Training questions Q and corpus C
Input: Supervision heuristic H
Input: Retrieval model R_0, number of rounds n
Output: Stronger retriever R_n

1 **for** round $t \leftarrow 1$ ***to*** n **do**
2 Index corpus C for retrieval with R_t.
3 $Rank_t \leftarrow \{R_i.retrieve(q) : q \in Q\}$
4 $P_t = \{H.getPositives(r[0:k^+])[0:t]$
5 $: r \in Rank_t\}$
6 $N_t = \{H.getNegatives(r[0:k^-])$
7 $: r \in Rank_t\}$
8 $T_t = \{\langle q, p, n\rangle : q \in Q,$
9 $p \in P_t[q], n \in N_t[q]\}$
10 Train a new retriever R_{t+1} using triples T_t.
11 **end**
12 **return** *Final retriever* R_n

- RGS takes as input a set of training question Q, a corpus of passages C, and an initial weak retrieval model (e.g., BM25)
- For every question q in the training set, we retrieve R's top-k results (Line 3)
- Take as positives (Line 4) the highest-ranked t (e.g., t = 3) passages that contain q's short answer, restricted to the top-k+ (e.g., k+ = 20) results
- Every passage in the top-k− (k− >> k+) that does not contain the short answer is treated as a negative (Line 6)
- We train a new retriever with these examples (Line 10), and repeat until all rounds are complete

Relevance-guided Supervision for OpenQA with ColBERT

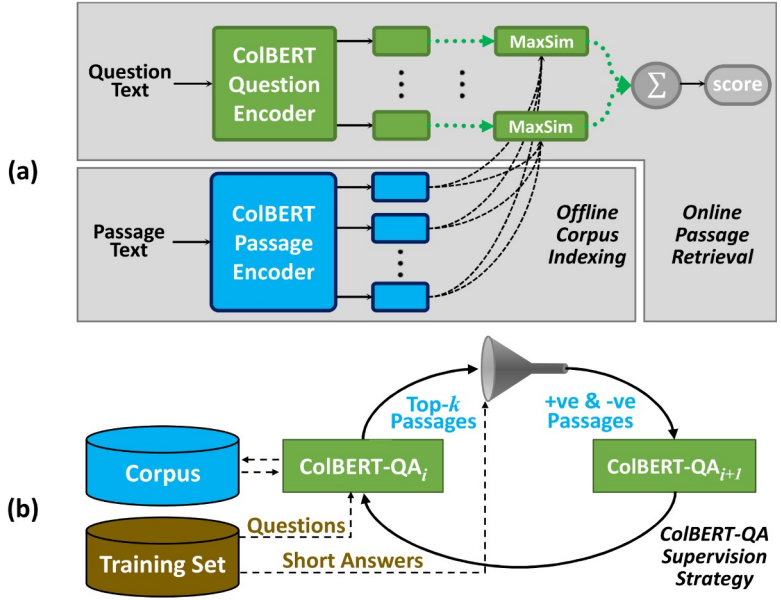


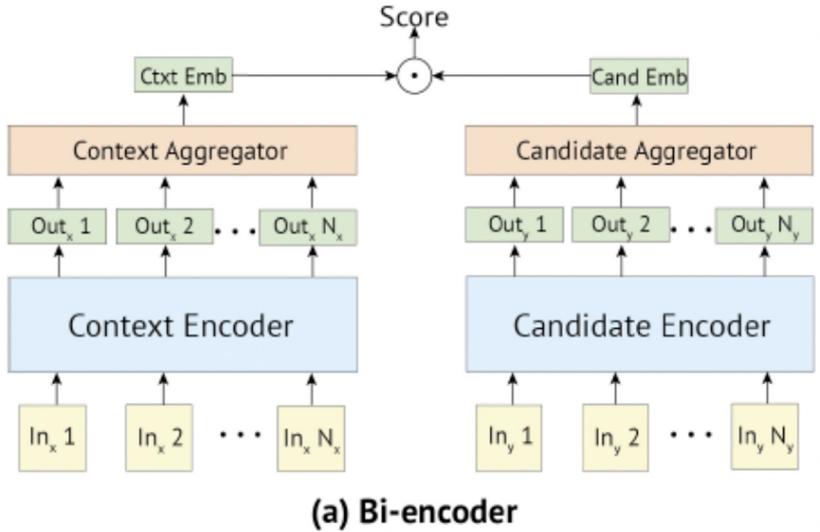
Figure 1: Sub-figure (a) depicts the ColBERT retrieval model (Khattab and Zaharia, 2020). ColBERT encodes questions and passages into multiple embeddings and allows those to interact via a scalable maximum-similarity (MaxSim) mechanism. Sub-figure (b) illustrates our proposed ColBERT-QA training strategy: we use an existing retrieval model to collect the top-k passages for every training question and, with a simple heuristic, sort these passages into positive (+ve) and negative (–ve) examples, using those to train another, more effective retriever. This process is applied thrice, and the resulting ColBERT-QA is used in the OpenQA pipeline.

3. ColBERTv2 : Supervision

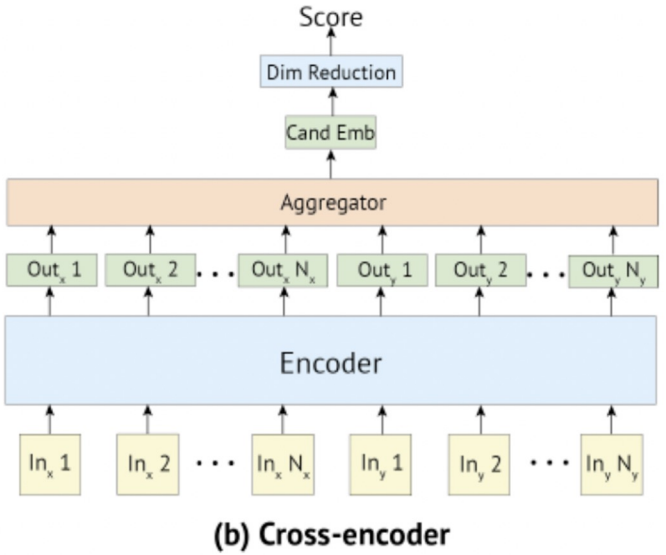
- 이를 위해서 Relevance-guided Supervision for OpenQA with ColBERT (Khattabl et al. (2021b)) 방식으로 ColBERT 모델 학습
- Each training query 에 대해, top-k passages retrieve 수행
- Top-k passages 의 query-passage pairs 를 cross-encoder reranker (22M-param MiniLM cross-encoder 모델) 에 입력하여, score 에 따라 d+, d- 선정. **즉, <q, d+, d-> 생성**
- **최종적으로, w-way tuples (w=64) consisting of a query, a highly-ranked passage (or labeled positive), and one or more lower-ranked passages**
    - 즉, <q, d+, d-_{0}>, <q, d+, d-_{1}>, … , <q, d+, d-_{63}>

MiniLM

이것!



(a) Bi-encoder



(b) Cross-encoder

MiniLM

- Pre-trained LM 은 성능은 좋은데 파라미터가 너무 크고, latency, capacity 문제가 있음
- MiniLM
  - Simple and effective approach to compress large Transformer based pre-trained models, termed as deep self-attention distillation

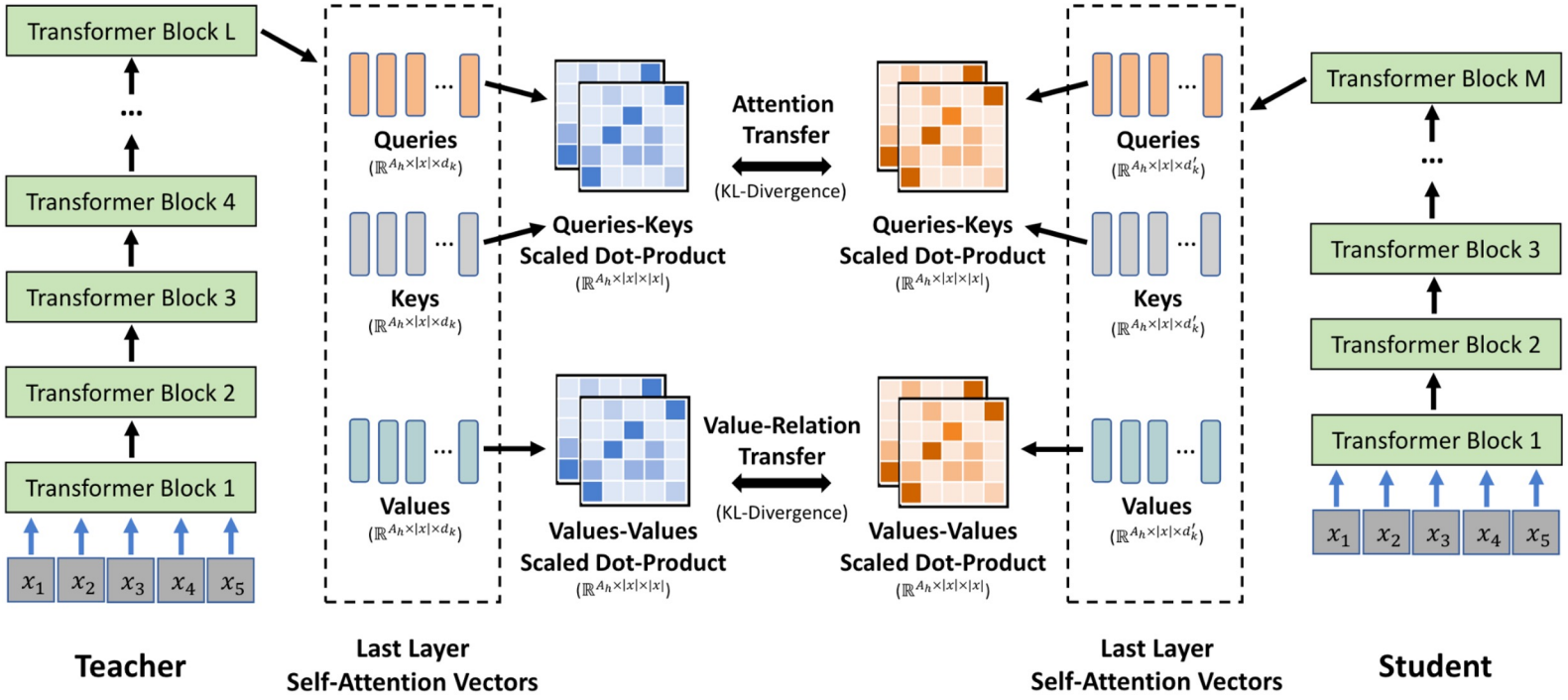


Figure 1. Overview of Deep Self-Attention Distillation. The student is trained by deeply mimicking the self-attention behavior of the last Transformer layer of the teacher. In addition to the self-attention distributions, we introduce the self-attention value-relation transfer to help the student achieve a deeper mimicry. Our student models are named as MINILM.

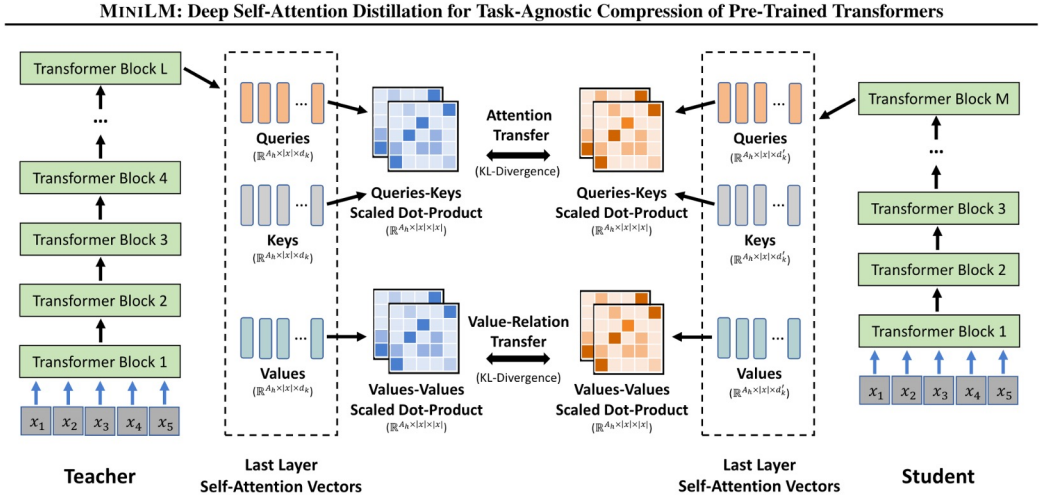


Figure 1. Overview of Deep Self-Attention Distillation. The student is trained by deeply mimicking the self-attention behavior of the last Transformer layer of the teacher. In addition to the self-attention distributions, we introduce the self-attention value-relation transfer to help the student achieve a deeper mimicry. Our student models are named as MINILM.

$$\mathcal{L} = \mathcal{L}_{\text{AT}} + \mathcal{L}_{\text{VR}}$$

$$\mathcal{L}_{\text{AT}} = \frac{1}{A_h|x|}\sum_{a=1}^{A_h}\sum_{t=1}^{|x|} D_{KL}(\mathbf{A}_{L,a,t}^{T} \parallel \mathbf{A}_{M,a,t}^{S}) \quad (6)$$

$$\mathbf{VR}_{L,a}^{T} = \text{softmax}(\frac{\mathbf{V}_{L,a}^{T}\mathbf{V}_{L,a}^{T\intercal}}{\sqrt{d_k}}) \quad (7)$$

$$\mathbf{VR}_{M,a}^{S} = \text{softmax}(\frac{\mathbf{V}_{M,a}^{S}\mathbf{V}_{M,a}^{S\intercal}}{\sqrt{d_k'}}) \quad (8)$$

$$\mathcal{L}_{\text{VR}} = \frac{1}{A_h|x|}\sum_{a=1}^{A_h}\sum_{t=1}^{|x|} D_{KL}(\mathbf{VR}_{L,a,t}^{T} \parallel \mathbf{VR}_{M,a,t}^{S}) \quad (9)$$

MiniLM

- Task
  - Extractive Question Answering (추출형 질의응답) : 문서에 대해서 질문을 제시하고 문서 자체에 존재하는 텍스트 범위(spans of text)를 해당 질문에 대한 답변으로 추출하는 작업

3. ColBERTv2 : Supervision

- Like RocketQAv2, we use a KL-Divergence loss to distill the cross-encoder's scores into the ColBERT architecture

RocketQAv2:
A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking

- Passage retrieval 과 passage re-ranking 두 개의 key procedures 는 ranking relevant information 을 찾는데 중요한 역할을 함
- 위의 두 개 key procedures 는 최종 성능에 영향을 많이 미치기 때문에, 두 개를 jointly optimize 하여 성능을 향상시키고자 함
- 본 연구에서, dense passage retrieval 과 passage reranking 을 위한 novel joint training approach 제안
- 주요 contribution : **dynamic listwise distillation**
    - The retriever and the re-ranker can be adaptively improved according to each other's relevance information

RocketQAv2:
A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking

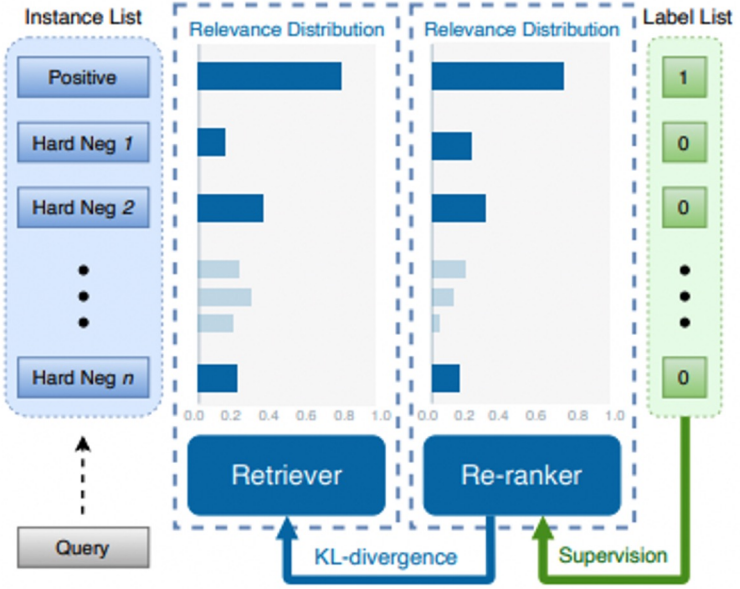


Figure 1: The illustration of dynamic listwise distillation in our approach.

- Query 에 해당하는 candidate passages (instance list) 선정
- Query-document 에 대해 1) dual-encoder-based retriever 와 2) cross-encoder-based retriever 각각 두 개의 score 계산
- 각 normalize 함

ranker, respectively. Then, we normalize them in a listwise way to obtain the corresponding relevance distributions over candidate passages:

$$\tilde{s}_{\mathrm{de}}(q,p) = \frac{e^{s_{\mathrm{de}}(q,p)}}{\sum_{p' \in \mathcal{P}_q} e^{s_{\mathrm{de}}(q,p')}}, \quad (2)$$

$$\tilde{s}_{\mathrm{ce}}(q,p) = \frac{e^{s_{\mathrm{ce}}(q,p)}}{\sum_{p' \in \mathcal{P}_q} e^{s_{\mathrm{ce}}(q,p')}}. \quad (3)$$

RocketQAv2:
A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking

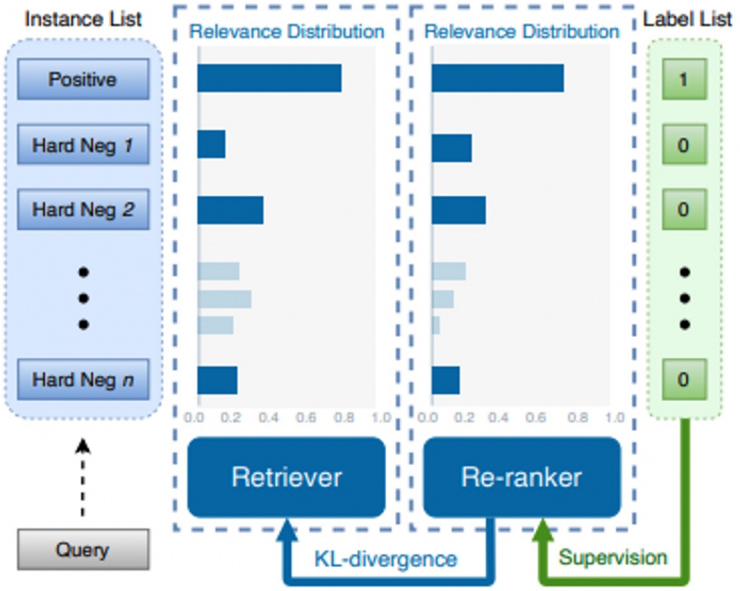


Figure 1: The illustration of dynamic listwise distillation in our approach.

- The main idea is to **adaptively reduce the difference between the two distributions from the retriever and the re-ranker so as to mutually improve each other**
- Adaptively mutual improvement 하기 위해 두 relevance distributions 간 KL-divergence 를 줄이도록 학습

$$\mathcal{L}_{\mathrm{KL}} = \sum_{q \in \mathcal{Q}, p \in \mathcal{P}_q} \tilde{s}_{\mathrm{de}}(q,p) \cdot \log \frac{\tilde{s}_{\mathrm{de}}(q,p)}{\tilde{s}_{\mathrm{ce}}(q,p)}. \quad (4)$$

RocketQAv2:
A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking

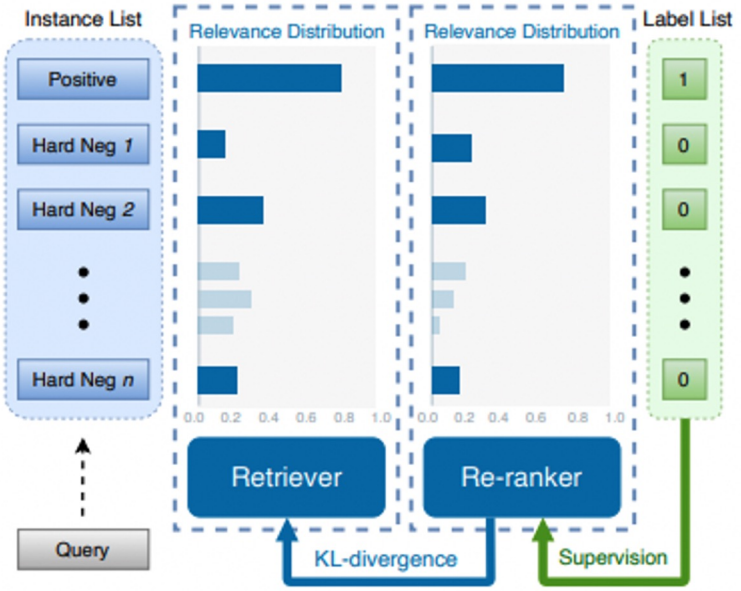


Figure 1: The illustration of dynamic listwise distillation in our approach.

Additionally, we provide ground-truth guidance for the joint training. Specifically, we also adopt a cross-entropy loss for the re-ranker based on passages in $\mathcal{P}_q$ with supervised information:

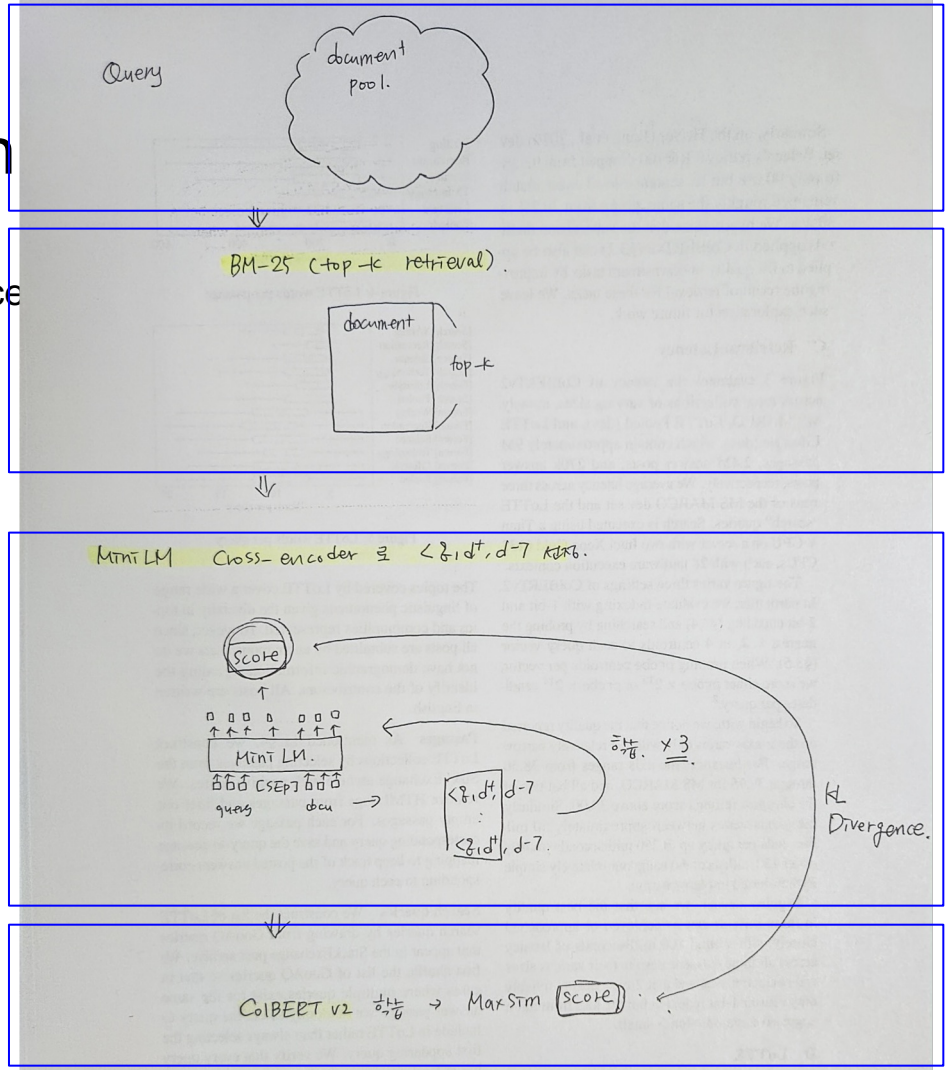
$$\mathcal{L}_{\text{sup}} = -\frac{1}{N} \sum_{q \in \mathcal{Q}, p^+} \log \frac{e^{s_{\text{ce}}(q,p^+)}}{e^{s_{\text{ce}}(q,p^+)} + \sum_{p^-} e^{s_{\text{ce}}(q,p^-)}}, \tag{5}$$

where N is the number of training instances, and p^+ and p^- denote the positive passage and negative passage in $\mathcal{P}_q$, respectively. We combine the KL-divergence loss and the supervised cross-entropy loss defined in Eq. (4) and Eq. (5) to obtain the final loss function:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{sup}}. \tag{6}$$

3. ColBERTv2 : Supervision

- Like RocketQAv2, we use a KL-Divergence ColBERT architecture



3. ColBERTv2 : Supervision

- Like RocketQAv2, we use a KL-Divergence loss to distill the cross-encoder's scores into the ColBERT architecture
- KL-Divergence 를 사용한 이유
    - ColBERT 는 제한된 스케일로 score 를 생성하기 때문에 (i.e. the sum of cosine similarities), cross-encoder 의 score 와 align 을 맞추기 위해 사용함
- In-batch negatives per GPU 사용

- 위의 denoised training with hard negatives 방식으로 single-vector 모델과 interaction-based model 의 성능 간극을 보완했음

[Recap] ColBERTv2

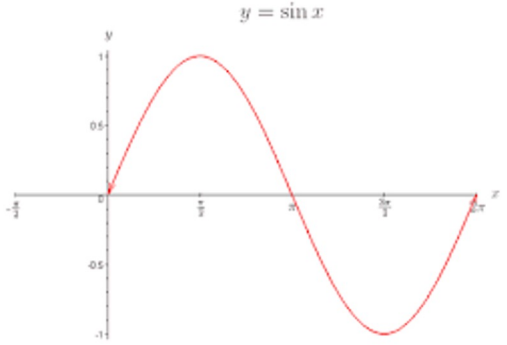
- ColBERTv2
  - Improves the quality of multi-vector retrieval models (Supervision)
  - **Reducing space footprint (Representation)**

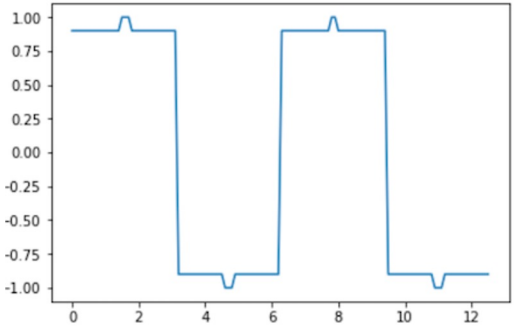
- ColBERTv2 Content
  - Modeling
  - Supervision
  - **Representation**
  - Indexing
  - Retrieval

3. ColBERTv2 : Representation

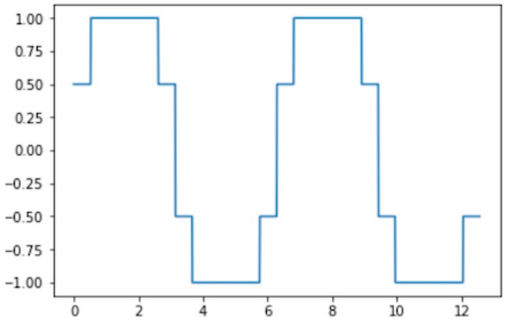
- Residual representation
  - Dramatically reduces the space footprint of late interaction models
  - Completely off-the-shelf without architectural or training changes
- ColBERTv2 는 각 벡터 v 를 1) index of its closest centroid Ct, 2) quantized vector ~r (tilda_r) 로 표기
  - r = v - Ct
  - ~r = quantized vector r
- At search time, use the centroid index t and residual ~r recover an approximate **~v = Ct + ~r**

Quantization의 쉬운 예제





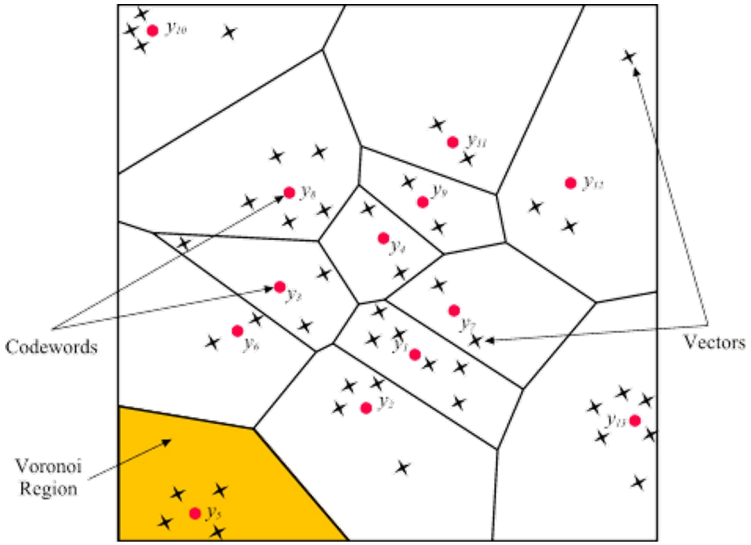
{1, 0.9, -0.9, -1}



{1, 0.5, -0.5, -1}

- y = sin(x) 그래프는 [-1, 1] 사이의 무한한 실수로 표현됨 (첫 번째)
- 이를 2bit 로 데이터를 표현해야 한다고 하면, 총 4개의 데이터로 표현할 수 있음 (두 번째, 세 번째)

Quantization의 쉬운 예제



03. Vector Quantization 정리

- **정의** (정보통신기술용어해설)
  - 연속적으로 샘플링된 진폭값들을 그룹핑하여, 이 그룹단위를 몇개의 대표값으로 양자화
    - 일련의 표본들을 특성화시킨 몇개의 대표값(n-tuple,순서쌍)으로 양자화
- **쉬운 정의**
  - N개의 특징 벡터 집합 $\mathbf{x}$를 K개의 특징 벡터 집합 $\mathbf{y}$로 mapping 하는 것
  - 예를 들어, 특징 벡터 집합이 아래와 같다고 하면,
    - $\mathbf{x}$ = {유재석, G-Idle,이정재 ,싸이, 아이유, 마동석, 강호동}
    - $\mathbf{y}$ = {가수, 영화배우, 개그맨}
  - 양자화 연산자 $f(*)$를 통해서 $\mathbf{y} = f(\mathbf{x})$와 같이 mapping하면 Vector Quantization 결과는 아래와 같음
    - 가수 = {G-Idle, 싸이, 아이유}
    - 영화배우 = {이정재, 마동석}
    - 개그맨 = {유재석, 강호동}
  - 이때 $\mathbf{y}$의 각 원소(i.e., 특징)들은 `Codeword` 또는 `Cluster` 라고 불리며, $\mathbf{y}$를 `Codebook` 이라고 함

따라서, 음성인식에서의 VQ-Wav2Vec이나 Wav2Vec 2.0에서 행해지는 Vector Quantization이란, 학습 가능한 Embedding Matrix를 Codebook으로 사용하여, Encoder를 거친 특징 벡터 $\mathbf{z}$를 Gumbel Softmax 또는 K-means 클러스터링 등으로 Codebook 내 Codeword에 mapping시키는 과정이라고 이해할 수 있겠다.

3. ColBERTv2 : Representation

- Residual representation
  - Dramatically reduces the space footprint of late interaction models

architectural or training changes. Given a set of centroids C, ColBERTv2 encodes each vector v as the index of its closest centroid C_t and a *quantized* vector $\tilde{r}$ that approximates the residual $r = v - C_t$. At search time, we use the centroid index t and residual $\tilde{r}$ recover an approximate $\tilde{v} = C_t + \tilde{r}$.

3. ColBERTv2 : Representation

- ~r 을 encoding 하기 위해서 r 의 모든 dimension 을 one or two bits 로 quantize
- b-bit encoding of n-dimensional vectors needs *log* |*C*| + *bn* bits per vector
    - n=128, four bytes to capture up to 2^32 centroids, 16 or 32 bytes to encode the residual
    - ColBERT 가 16 bit precision 에서 256 bytes vector encoding 을 사용하는데 반해, 20 or 36 bytes 를 사용했음
- 다른 compression 대안들 보다도 위 방법이 model quality 를 유지하면서 storage costs 를 효과적으로 줄여줌을 확인했음
- We encode each vector using its nearest centroid plus a residual

- 위의 centroid-based encoding 방식은 기존 single vector 에서 compression 하기 위해 사용한 product quantization 방식을 multi vector 로 확장한 것으로 볼 수 있음

[Recap] ColBERTv2

- ColBERTv2
  - Improves the quality of multi-vector retrieval models (Supervision)
  - Reducing space footprint (Representation)

- ColBERTv2 Content
  - Modeling
  - Supervision
  - Representation
  - **Indexing**
  - Retrieval

3. ColBERTv2 : Indexing

- 인덱싱을 3가지 스테이지로 나눠 수행
- **1. Centroid Selection**
  - ColBERTv2 는 set of cluster centroids C 를 선택
  - 보통 |C|의 크기 는 임베딩 크기의 square root 에 비례하여 선택
  - 메모리 소비를 줄이기 위해서 k-means clustering 기법으로 centroids C 선정
- **2. Passage Encoding**
  - Centroids 를 선택한 후, 모든 passage 를 압축하여 인코딩
  - Passages 의 chunk 가 인코딩 된 후, compressed representation 들은 disk 에 저장
- **3. Index Inversion**
  - Fast nearest-neighbor search 를 위해서, 각 centroid 에 속하는 embedding IDs 를 그룹핑
    => save this ***inverted list*** to disk
  - At search time, *inverted list* 를 사용해 query 와 유사한 token-level embeddings 를 찾도록 함

[Recap] ColBERTv2

- ColBERTv2
  - Improves the quality of multi-vector retrieval models (Supervision)
  - Reducing space footprint (Representation)

- ColBERTv2 Content
  - Modeling
  - Supervision
  - Representation
  - Indexing
  - **Retrieval**

3. ColBERTv2 : Retrieval

- Query representation Q 가 주어졌을 때, retrieval 을 위해 candidate generation 먼저 수행
- vector Qi 에 대해서, 가까운 centroids n_probe 개를 찾음
- Inverted list 를 사용해서 가까운 centroids 에 해당하는 passage embeddings 를 찾음
- 위의 passage embeddings 을 decompress 한 다음, Qi 와의 cosine similarity 계산 (MaxSim)
- **각 문서의 모든 passage embedding 을 사용한 ColBERT 와는 다르게, centroid 에 해당하는 passage embedding 만을 사용하여 passage ID 의 score 계산**
- 이는 ColBERTv2 가 query 별로 approximate ‘MaxSim’ 을 계산하도록 함
- 최종적으로 top-scoring 의 n_candidate 는 ranking 의 후보로 선정되어 각 passage 에 해당하는 모든 embeddings 값을 로드하게 됨 => 이 후 과정은 ColBERT 와 동일

4. LoTTE: Long-Tail, Cross-Domain Retrieval Evaluation

- 논문에서 소개한 데이터셋, Long-Tail Topic-stratified Evaluation (pronounced latte)
  - Out-of-domain tests 문제를 완화하기 위해서
- LoTTE 는 natural user queries that pertain to long-tail topics 에 초점을 맞춤 (Wikipedia 같은 entity-centric knowledge 에 속하지 않은 데이터만)
- Test sets 는 topic 으로 구분되며, 별개의 validation set 존재

Topic	Question Set	Dev			Test		
		# Questions	# Passages	Subtopics	# Questions	# Passages	Subtopics
Writing	Search Forum	497 2003	277k	ESL, Linguistics, Worldbuilding	1071 2000	200k	English
Recreation	Search Forum	563 2002	263k	Sci-Fi, RPGs, Photography	924 2002	167k	Gaming, Anime, Movies
Science	Search Forum	538 2013	344k	Chemistry, Statistics, Academia	617 2017	1.694M	Math, Physics, Biology
Technology	Search Forum	916 2003	1.276M	Web Apps, Ubuntu, SysAdmin	596 2004	639k	Apple, Android, UNIX, Security
Lifestyle	Search Forum	417 2076	269k	DIY, Music, Bicycles, Car Maintenance	661 2002	119k	Cooking, Sports, Travel
Pooled	Search Forum	2931 10097	2.4M	All of the above	3869 10025	2.8M	All of the above

Table 1: Composition of LoTTE showing topics, question sets, and a sample of corresponding subtopics. Search Queries are taken from GooAQ, while Forum Queries are taken directly from the StackExchange archive. The pooled datasets combine the questions and passages from each of the subtopics.

4. LoTTE: Long-Tail, Cross-Domain Retrieval Evaluation

- Search Queries
  - Collect search queries from GooQA, a recent dataset of Google search-autocomplete queries and their answer boxes
  - **Filter for queries whose answers link to a specific StackExchange post**
- Forum Queries
  - **Extracting post titles from the StackExchange communities to use as queries and collect their corresponding answer posts as targets**
  - Select questions in order of their popularity and sample questions according to the proportional contribution of individual communities within each topic
- Search queries 는 좀 더 짧고, knowledge-based questions with direct answers 인 반면, forum queries 는 좀 더 open-ended questions 을 포함하고 있음

4. LoTTE: Long-Tail, Cross-Domain Retrieval Evaluation

Q: *what is the difference between root and stem in linguistics?* **A:** A root is **the form to which derivational affixes are added** to form a stem. A stem is **the form to which inflectional affixes are added** to form a word.
Q: *are there any airbenders left?* **A:** the Fire Nation had wiped out all Airbenders while Aang was frozen. **Tenzin and his 3 children are the only Airbenders left in Korra's time.**
Q: *Why are there two Hydrogen atoms on some periodic tables?* **A:** some periodic tables show hydrogen in both places **to emphasize that hydrogen isn't really a member of the first group or the seventh group.**
Q: *How can cache be that fast?* **A:** the cache memory sits right next to the CPU on the same die (chip), **it is made using SRAM which is much, much faster than the DRAM.**

Table 2: Examples of queries and shortened snippets of answer passages from LoTTE. The first two examples show "search" queries, whereas the last two are "forum" queries. Snippets are shortened for presentation.

Q: *what is xerror in rpart?* **Q:** *is sub question one word?* **Q:** *how to open a garage door without making noise?* **Q:** *is docx and dotx the same?* **Q:** *are upvotes and downvotes anonymous?* **Q:** *what is the difference between descriptive essay and narrative essay?* **Q:** *how to change default user profile in chrome?* **Q:** *does autohotkey need to be installed?* **Q:** *how do you tag someone on facebook with a youtube video?* **Q:** *has mjolnir ever been broken?*
Q: *Snoopy can balance on an edge atop his doghouse. Is any reason given for this?* **Q:** *How many Ents were at the Entmoot?* **Q:** *What does a hexagonal sun tell us about the camera lens/sensor?* **Q:** *Should I simply ignore it if authors assume that Im male in their response to my review of their article?* **Q:** *Why is the 2s orbital lower in energy than the 2p orbital when the electrons in 2s are usually farther from the nucleus?* **Q:** *Are there reasons to use colour filters with digital cameras?* **Q:** *How does the current know how much to flow, before having seen the resistor?* **Q:** *What is the difference between Fact and Truth?* **Q:** *hAs a DM, how can I handle my Druid spying on everything with Wild shape as a spider?* **Q:** *What does 1x1 convolution mean in a neural network?*

Table 3: Comparison of a random sample of search queries (top) vs. forum queries (bottom).

5. Evaluation

- Evaluate on passage retrieval tasks, testing its quality within the training domain as well as outside the training domain in zero-shot settings
- ColBERTv2 embeddings to b=2 bits per dimension in our evaluation

5. Evaluation

- In-domain Retrieval Quality
  - Train 데이터셋, Test 데이터셋 : MS MARCO Passage Ranking 데이터셋
  - Single-vector systems, ColBERT, ColBERTv2 와 비교 (w/ w/o distillation)

Method	Official Dev (7k)			Local Eval (5k)		
	MRR@10	R@50	R@1k	MRR@10	R@50	R@1k
Models without Distillation or Special Pretraining						
RepBERT	30.4	-	94.3	-	-	-
DPR	31.1	-	95.2	-	-	-
ANCE	33.0	-	95.9	-	-	-
LTRe	34.1	-	96.2	-	-	-
ColBERT	36.0	82.9	96.8	36.7	-	-
Models with Distillation or Special Pretraining						
TAS-B	34.7	-	97.8	-	-	-
SPLADEv2	36.8	-	97.9	37.9	84.9	98.0
PAIR	37.9	86.4	98.2	-	-	-
coCondenser	38.2	-	**98.4**	-	-	-
RocketQAv2	38.8	86.2	98.1	39.8	85.8	97.9
ColBERTv2	**39.7**	**86.8**	**98.4**	**40.8**	**86.3**	**98.3**

5. Evaluation

- Out-domain Retrieval Quality
  - Train 데이터셋 : MS MARCO Passage Ranking 데이터셋
  - Test 데이터셋
    - BEIR
    - Wikipedia Open QA
    - LoTTE

Corpus	Models without Distillation				Models with Distillation			
	ColBERT	DPR-M	ANCE	MoDIR	TAS-B	RocketQAv2	SPLADEv2	ColBERTv2
BEIR Search Tasks (nDCG@10)								
DBPedia	39.2	23.6	28.1	28.4	38.4	35.6	43.5	**44.6**
FiQA	31.7	27.5	29.5	29.6	30.0	30.2	33.6	**35.6**
NQ	52.4	39.8	44.6	44.2	46.3	50.5	52.1	**56.2**
HotpotQA	59.3	37.1	45.6	46.2	58.4	53.3	**68.4**	66.7
NFCorpus	30.5	20.8	23.7	24.4	31.9	29.3	33.4	**33.8**
T-COVID	67.7	56.1	65.4	67.6	48.1	67.5	71.0	**73.8**
Touché (v2)	-	-	-	-	-	24.7	**27.2**	26.3
BEIR Semantic Relatedness Tasks (nDCG@10)								
ArguAna	23.3	41.4	41.5	41.8	42.7	45.1	**47.9**	46.3
C-FEVER	18.4	17.6	19.8	20.6	22.8	18.0	**23.5**	17.6
FEVER	77.1	58.9	66.9	68.0	70.0	67.6	**78.6**	**78.5**
Quora	85.4	84.2	85.2	**85.6**	83.5	74.9	83.8	85.2
SCIDOCS	14.5	10.8	12.2	12.4	14.9	13.1	**15.8**	15.4
SciFact	67.1	47.8	50.7	50.2	64.3	56.8	**69.3**	**69.3**

(a)

Corpus	ColBERT	BM25	ANCE	RocketQAv2	SPLADEv2	ColBERTv2
OOD Wikipedia Open QA (Success@5)						
NQ-dev	65.7	44.6	-	-	65.6	**68.9**
TQ-dev	72.6	67.6	-	-	74.7	**76.7**
SQuAD-dev	60.0	50.6	-	-	60.4	**65.0**
LoTTE Search Test Queries (Success@5)						
Writing	74.7	60.3	74.4	78.0	77.1	**80.1**
Recreation	68.5	56.5	64.7	72.1	69.0	**72.3**
Science	53.6	32.7	53.6	55.3	55.4	**56.7**
Technology	61.9	41.8	59.6	63.4	62.4	**66.1**
Lifestyle	80.2	63.8	82.3	82.1	82.3	**84.7**
Pooled	67.3	48.3	66.4	69.8	68.9	**71.6**
LoTTE Forum Test Queries (Success@5)						
Writing	71.0	64.0	68.8	71.5	73.0	**76.3**
Recreation	65.6	55.4	63.8	65.7	67.1	**70.8**
Science	41.8	37.1	36.5	38.0	43.7	**46.1**
Technology	48.5	39.4	46.8	47.3	50.8	**53.6**
Lifestyle	73.0	60.6	73.1	73.7	74.0	**76.9**
Pooled	58.2	47.2	55.7	57.7	60.1	**63.4**

(b)

Table 5: Zero-shot evaluation results. Sub-table (a) reports results on BEIR and sub-table (b) reports results on the Wikipedia Open QA and the test sets of the LoTTE benchmark. On BEIR, we test ColBERTv2 and RocketQAv2 and copy the results for ANCE, TAS-B, and ColBERT from Thakur et al. (2021), for MoDIR and DPR-MSMARCO (DPR-M) from Xin et al. (2021), and for SPLADEv2 from Formal et al. (2021a).

5. Evaluation

- Efficiency
  - ColBERTv2’s residual compression approach significantly reduces index sizes compared to vanilla ColBERT
  - ColBERT 는 MS MARCO 의 index 를 저장하기 위해 154GiB 가 필요했는데, ColBERTv2 는 16GiB 또는 25GiB 가 필요함
  - 또한 single-vector 모델을 사용할 때, 최소 25GiB 가 필요함, HNSW 와 같은 fast search nearest-neighbor index 를 사용하면 저장 공간이 더 늘어남

6. Conclusion

- ColBERTv2, a retriever that advances the quality and space efficiency of multi-vector representations
  - Residual representation 으로 multi-vector footprint 를 감소시킴
  - Supervision 을 통해 quality 향상
- 제안한 ColBERTv2 는 within-domain, out-of-domain 모두에서 retrievers 성능이 뛰어남

Thank You