

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

Abstract

Neural information retrieval (IR) has greatly advanced search and other knowledge-intensive language tasks. While many neural IR methods encode queries and documents into single-vector representations, late interaction models produce multi-vector representations at the granularity of each token and decompose relevance modeling into scalable token-level computations. This decomposition has been shown to make late interaction more effective, but it inflates the space footprint of these models by an order of magnitude. In this work, we introduce ColBERTv2, a retriever that couples an aggressive residual compression mechanism with a denoised supervision strategy to simultaneously improve the quality and space footprint of late interaction. We evaluate ColBERTv2 across a wide range of benchmarks, establishing state-of-the-art quality within and outside the training domain while reducing the space footprint of late interaction models by 6–10 \times .

1 Introduction

Neural information retrieval (IR) has quickly dominated the search landscape over the past 2–3 years, dramatically advancing not only passage and document search (Nogueira and Cho, 2019) but also many knowledge-intensive NLP tasks like open-domain question answering (Gua et al., 2020), multi-hop claim verification (Khattab et al., 2021a), and open-ended generation (Paranjape et al., 2022).

Many neural IR methods follow a *single-vector similarity* paradigm: a pretrained language model is used to encode each query and each document into a single high-dimensional vector, and relevance is modeled as a simple dot product between both vectors. An alternative is *late interaction*, introduced in ColBERT (Khattab and Zaharia, 2020), where queries and documents are encoded at a finer-granularity into multi-vector representations, and

relevance is estimated using rich yet scalable interactions between these two sets of vectors. ColBERT produces an embedding for every token in the query (and document) and models relevance as the sum of maximum similarities between each query vector and all vectors in the document.

By decomposing relevance modeling into token-level computations, late interaction aims to reduce the burden on the encoder: whereas single-vector models must capture complex query–document relationships within one dot product, late interaction encodes meaning at the level of tokens and delegates query–document matching to the interaction mechanism. This added expressivity comes at a cost: existing late interaction systems impose an order-of-magnitude larger *space footprint* than single-vector models, as they must store billions of small vectors for Web-scale collections. Considering this challenge, it might seem more fruitful to focus instead on addressing the fragility of single-vector models (Menon et al., 2022) by introducing new supervision paradigms for negative mining (Xiong et al., 2020), pretraining (Gao and Callan, 2021), and distillation (Qu et al., 2021). Indeed, recent single-vector models with highly-tuned supervision strategies (Ren et al., 2021b; Formal et al., 2021a) sometimes perform on-par or even better than “vanilla” late interaction models, and it is not necessarily clear whether late interaction architectures—with their fixed token-level inductive biases—admit similarly large gains from improved supervision.

In this work, we show that late interaction retrievers naturally produce lightweight token representations that are amenable to efficient storage off-the-shelf and that they can benefit drastically from denoised supervision. We couple those in **ColBERTv2**,¹ a new late-interaction retriever that employs a simple combination of distillation from

*Equal contribution.

¹Code, models, and LoTTE data are maintained at <https://github.com/stanford-futuredata/ColBERT>

a cross-encoder and hard-negative mining (§3.2) to boost quality beyond any existing method, and then uses a *residual compression* mechanism (§3.3) to reduce the space footprint of late interaction by 6–10 \times while preserving quality. As a result, ColBERTv2 establishes state-of-the-art retrieval quality both *within* and *outside* its training domain with a competitive space footprint with typical single-vector models.

When trained on MS MARCO Passage Ranking, ColBERTv2 achieves the highest MRR@10 of any standalone retriever. In addition to in-domain quality, we seek a retriever that generalizes “zero-shot” to domain-specific corpora and long-tail topics, ones that are often under-represented in large public training sets. To this end, we evaluate ColBERTv2 on a wide array of *out-of-domain* benchmarks. These include three Wikipedia Open-QA retrieval tests and 13 diverse retrieval and semantic-similarity tasks from BEIR (Thakur et al., 2021). In addition, we introduce a new benchmark, dubbed **LoTTE**, for Long-Tail Topic-stratified Evaluation for IR that features 12 domain-specific search tests, spanning StackExchange communities and using queries from GooAQ (Khashabi et al., 2021). LoTTE focuses on relatively long-tail topics in its passages, unlike the Open-QA tests and many of the BEIR tasks, and evaluates models on their capacity to answer natural search queries with a practical intent, unlike many of BEIR’s semantic-similarity tasks. On 22 of 28 out-of-domain tests, ColBERTv2 achieves the highest quality, outperforming the next best retriever by up to 8% relative gain, while using its compressed representations.

This work makes the following contributions:

1. We 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.
2. We 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.
3. We evaluate ColBERTv2 across a wide range of settings, establishing state-of-the-art quality within and outside the training domain.

2 Background & Related Work

2.1 Token-Decomposed Scoring in Neural IR

Many neural IR approaches encode passages as a single high-dimensional vector, trading off the higher quality of cross-encoders for improved efficiency and scalability (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2021). ColBERT’s (Khattab and Zaharia, 2020) late interaction paradigm addresses this tradeoff by computing multi-vector embeddings and using a scalable “MaxSim” operator for retrieval. Several other systems leverage multi-vector representations, including Poly-encoders (Humeau et al., 2020), PreTTR (MacAvaney et al., 2020), and MORES (Gao et al., 2020), but these target attention-based re-ranking as opposed to ColBERT’s scalable MaxSim end-to-end retrieval.

ME-BERT (Luan et al., 2021) generates token-level document embeddings similar to ColBERT, but retains a single embedding vector for queries. COIL (Gao et al., 2021) also generates token-level document embeddings, but the token interactions are restricted to lexical matching between query and document terms. uniCOIL (Lin and Ma, 2021) limits the token embedding vectors of COIL to a single dimension, reducing them to scalar weights that extend models like DeepCT (Dai and Callan, 2020) and DeepImpact (Mallia et al., 2021). To produce scalar weights, SPLADE (Formal et al., 2021b) and SPLADEv2 (Formal et al., 2021a) produce a sparse vocabulary-level vector that retains the term-level decomposition of late interaction while simplifying the storage into one dimension per token. The SPLADE family also piggybacks on the language modeling capacity acquired by BERT during pretraining. SPLADEv2 has been shown to be highly effective, within and across domains, and it is a central point of comparison in the experiments we report on in this paper.

2.2 Vector Compression for Neural IR

There has been a surge of recent interest in compressing representations for IR. Izacard et al. (2020) explore dimension reduction, product quantization (PQ), and passage filtering for single-vector retrievers. BPR (Yamada et al., 2021a) learns to directly hash embeddings to binary codes using a differentiable tanh function. JPQ (Zhan et al., 2021a) and its extension, RepCONC (Zhan et al., 2022), use PQ to compress embeddings, and jointly train the query encoder along with the centroids produced

by PQ via a ranking-oriented loss.

SDR (Cohen et al., 2021) uses an autoencoder to reduce the dimensionality of the contextual embeddings used for attention-based re-ranking and then applies a quantization scheme for further compression. DensePhrases (Lee et al., 2021a) is a system for Open-QA that relies on a multi-vector encoding of passages, though its search is conducted at the level of individual vectors and not aggregated with late interaction. Very recently, Lee et al. (2021b) propose a quantization-aware finetuning method based on PQ to reduce the space footprint of DensePhrases. While DensePhrases is effective at Open-QA, its retrieval quality—as measured by top-20 retrieval accuracy on NaturalQuestions and TriviaQA—is competitive with DPR (Karpukhin et al., 2020) and considerably less effective than ColBERT (Khattab et al., 2021b).

In this work, we focus on late-interaction retrieval and investigate compression using a residual compression approach that can be applied off-the-shelf to late interaction models, without special training. We show in Appendix A that ColBERT’s representations naturally lend themselves to residual compression. Techniques in the family of residual compression are well-studied (Barnes et al., 1996) and have previously been applied across several domains, including approximate nearest neighbor search (Wei et al., 2014; Ai et al., 2017), neural network parameter and activation quantization (Li et al., 2021b,a), and distributed deep learning (Chen et al., 2018; Liu et al., 2020). To the best of our knowledge, ColBERTv2 is the first approach to use residual compression for scalable neural IR.

2.3 Improving the Quality of Single-Vector Representations

Instead of compressing multi-vector representations as we do, much recent work has focused on improving the quality of single-vector models, which are often very sensitive to the specifics of supervision. This line of work can be decomposed into three directions: (1) distillation of more expressive architectures (Hofstätter et al., 2020; Lin et al., 2020) including explicit denoising (Qu et al., 2021; Ren et al., 2021b), (2) hard negative sampling (Xiong et al., 2020; Zhan et al., 2020a, 2021b), and (3) improved pretraining (Gao and Callan, 2021; Oğuz et al., 2021). We adopt similar techniques to (1) and (2) for ColBERTv2’s multi-vector representations (see §3.2).

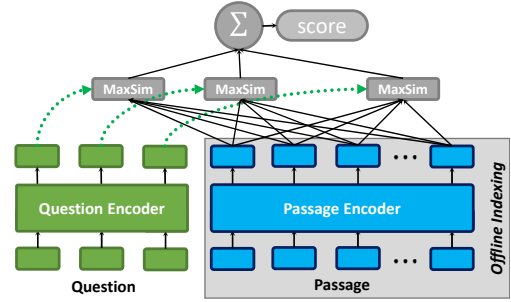


Figure 1: The late interaction architecture, given a query and a passage. Diagram from Khattab et al. (2021b) with permission.

2.4 Out-of-Domain Evaluation in IR

Recent progress in retrieval has mostly focused on large-data evaluation, where many tens of thousands of annotated training queries are associated with the test domain, as in MS MARCO or Natural Questions (Kwiatkowski et al., 2019). In these benchmarks, queries tend to reflect high-popularity topics like movies and athletes in Wikipedia. In practice, user-facing IR and QA applications often pertain to domain-specific corpora, for which little to no training data is available and whose topics are under-represented in large public collections.

This out-of-domain regime has received recent attention with the BEIR (Thakur et al., 2021) benchmark. BEIR combines several existing datasets into a heterogeneous suite for “zero-shot IR” tasks, spanning bio-medical, financial, and scientific domains. While the BEIR datasets provide a useful testbed, many capture broad semantic relatedness tasks—like citations, counter arguments, or duplicate questions—instead of natural search tasks, or else they focus on high-popularity entities like those in Wikipedia. In §4, we introduce LoTTE, a new dataset for out-of-domain retrieval, exhibiting natural search queries over long-tail topics.

3 ColBERTv2

We now introduce ColBERTv2, which improves the quality of multi-vector retrieval models (§3.2) while reducing their space footprint (§3.3).

3.1 Modeling

ColBERTv2 adopts the late interaction architecture of ColBERT, depicted in Figure 1. Queries and passages are independently encoded with BERT (Devlin et al., 2019), and the output embeddings encoding each token are projected to a lower dimension. During offline indexing, every passage d in the corpus is encoded into a set of vectors, and these

vectors are stored. At search time, the query q is encoded into a multi-vector representation, and its similarity to a passage d is computed as the summation of query-side “MaxSim” operations, namely, the largest cosine similarity between each query token embedding and all passage token embeddings:

$$S_{q,d} = \sum_{i=1}^N \max_{j=1}^M Q_i \cdot D_j^T \quad (1)$$

where Q is a matrix encoding the query with N vectors and D encodes the passage with M vectors. The intuition of this architecture is to align each query token with the most contextually relevant passage token, quantify these matches, and combine the partial scores across the query. We refer to Khattab and Zaharia (2020) for a more detailed treatment of late interaction.

3.2 Supervision

Training a neural retriever typically requires *positive* and *negative* passages for each query in the training set. Khattab and Zaharia (2020) train ColBERT using the official $\langle q, d^+, d^- \rangle$ triples of MS MARCO. For each query, a positive d^+ is human-annotated, and each negative d^- is sampled from unannotated BM25-retrieved passages.

Subsequent work has identified several weaknesses in this standard supervision approach (see §2.3). Our goal is to adopt a simple, uniform supervision scheme that selects challenging negatives and avoids rewarding false positives or penalizing false negatives. To this end, we start with a ColBERT model trained with triples as in Khattab et al. (2021b), using this to index the training passages with ColBERTv2 compression.

For each training query, we retrieve the top- k passages. We feed each of those query–passage pairs into a cross-encoder reranker. We use a 22M-parameter MiniLM (Wang et al., 2020) cross-encoder trained with distillation by Thakur et al. (2021).² This small model has been shown to exhibit very strong performance while being relatively efficient for inference, making it suitable for distillation.

We then collect w -way tuples consisting of a query, a highly-ranked passage (or labeled positive), and one or more lower-ranked passages. In this work, we use $w = 64$ passages per example. Like RocketQAv2 (Ren et al., 2021b), we use a

KL-Divergence loss to distill the cross-encoder’s scores into the ColBERT architecture. We use KL-Divergence as ColBERT produces scores (i.e., the sum of cosine similarities) with a restricted scale, which may not align directly with the output scores of the cross-encoder. We also employ in-batch negatives per GPU, where a cross-entropy loss is applied to the positive score of each query against all passages corresponding to other queries in the same batch. We repeat this procedure once to refresh the index and thus the sampled negatives.

Denoted training with hard negatives has been positioned in recent work as ways to bridge the gap between single-vector and interaction-based models, including late interaction architectures like ColBERT. Our results in §5 reveal that such supervision can improve multi-vector models dramatically, resulting in state-of-the-art retrieval quality.

3.3 Representation

We hypothesize that the ColBERT vectors cluster into regions that capture highly-specific token semantics. We test this hypothesis in Appendix A, where evidence suggests that vectors corresponding to each sense of a word cluster closely, with only minor variation due to context. We exploit this regularity with a *residual* representation that dramatically reduces the space footprint of late interaction models, completely *off-the-shelf* without 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 $\tilde{v} = C_t + \tilde{r}$.

To encode \tilde{r} , we quantize every dimension of r into one or two bits. In principle, our b -bit encoding of n -dimensional vectors needs $\lceil \log |C| \rceil + bn$ bits per vector. In practice, with $n = 128$, we use four bytes to capture up to 2^{32} centroids and 16 or 32 bytes (for $b = 1$ or $b = 2$) to encode the residual. This total of 20 or 36 bytes per vector contrasts with ColBERT’s use of 256-byte vector encodings at 16-bit precision. While many alternatives can be explored for compression, we find that this simple encoding largely preserves model quality, while considerably lowering storage costs against typical 32- or 16-bit precision used by existing late interaction systems.

This centroid-based encoding can be considered a natural extension of product quantization to *multi-*

²<https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2>

vector representations. Product quantization (Gray, 1984; Jegou et al., 2010) compresses a single vector by splitting it into small sub-vectors and encoding each of them using an ID within a codebook. In our approach, each representation is already a matrix that is naturally divided into a number of small vectors (one per token). We encode each vector using its nearest centroid plus a residual. Refer to Appendix B for tests of the impact of compression on retrieval quality and a comparison with a baseline compression method for ColBERT akin to BPR (Yamada et al., 2021b).

3.4 Indexing

Given a corpus of passages, the indexing stage precomputes all passage embeddings and organizes their representations to support fast nearest-neighbor search. ColBERTv2 divides indexing into three stages, described below.

Centroid Selection. In the first stage, ColBERTv2 selects a set of cluster centroids C . These are embeddings that ColBERTv2 uses to support residual encoding (§3.3) and also for nearest-neighbor search (§3.5). Standardly, we find that setting $|C|$ proportionally to the square root of $n_{\text{embeddings}}$ in the corpus works well empirically.³ Khattab and Zaharia (2020) only clustered the vectors after computing the representations of all passages, but doing so requires storing them uncompressed. To reduce memory consumption, we apply k -means clustering to the embeddings produced by invoking our BERT encoder over only a sample of all passages, proportional to the square root of the collection size, an approach we found to perform well in practice.

Passage Encoding. Having selected the centroids, we encode every passage in the corpus. This entails invoking the BERT encoder and compressing the output embeddings as described in §3.3, assigning each embedding to the nearest centroid and computing a quantized residual. Once a chunk of passages is encoded, the *compressed* representations are saved to disk.

Index Inversion. To support fast nearest-neighbor search, we group the embedding IDs that correspond to each centroid together, and save this *inverted list* to disk. At search time, this allows us to quickly find token-level embeddings similar to those in a query.

³We round down to the nearest power of two larger than $16 \times \sqrt{n_{\text{embeddings}}}$, inspired by FAISS (Johnson et al., 2019).

3.5 Retrieval

Given a query representation Q , retrieval starts with candidate generation. For every vector Q_i in the query, the nearest $n_{\text{probe}} \geq 1$ centroids are found. Using the inverted list, ColBERTv2 identifies the passage embeddings close to these centroids, decompresses them, and computes their cosine similarity with every query vector. The scores are then grouped by passage ID for each query vector, and scores corresponding to the same passage are max-reduced. This allows ColBERTv2 to conduct an approximate “MaxSim” operation per query vector. This computes a lower-bound on the true MaxSim (§3.1) using the embeddings identified via the inverted list, which resembles the approximation explored for scoring by Macdonald and Tonellotto (2021) but is applied for candidate generation.

These lower bounds are summed across the query tokens, and the top-scoring $n_{\text{candidate}}$ candidate passages based on these approximate scores are selected for ranking, which loads the complete set of embeddings of each passage, and conducts the same scoring function using all embeddings per document following Equation 1. The result passages are then sorted by score and returned.

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

We introduce **LoTTE** (pronounced latte), a new dataset for **Long-Tail Topic-stratified Evaluation** for IR. To complement the out-of-domain tests of BEIR (Thakur et al., 2021), as motivated in §2.4, LoTTE focuses on *natural user queries* that pertain to *long-tail topics*, ones that might not be covered by an entity-centric knowledge base like Wikipedia. LoTTE consists of 12 test sets, each with 500–2000 queries and 100k–2M passages.

The test sets are explicitly divided by topic, and each test set is accompanied by a validation set of *related but disjoint* queries and passages. We elect to make the passage texts disjoint to encourage more realistic out-of-domain transfer tests, allowing for minimal development on related but distinct topics. The test (and dev) sets include a “pooled” setting. In the pooled setting, the passages and queries are aggregated across all test (or dev) topics to evaluate out-of-domain retrieval across a larger and more diverse corpus.

Table 1 outlines the composition of LoTTE. We derive the topics and passage corpora from the *answer posts* across various StackExchange fo-

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.

forums. StackExchange is a set of question-and-answer communities that target individual topics (e.g., “physics” or “bicycling”). We gather forums from five overarching domains: writing, recreation, science, technology, and lifestyle. To evaluate retrievers, we collect *Search* and *Forum* queries, each of which is associated with one or more target answer posts in its corpus. Example queries, and short snippets from posts that answer them in the corpora, are shown in Table 2.

Search Queries. We collect search queries from GooAQ (Khashabi et al., 2021), a recent dataset of Google search-autocomplete queries and their answer boxes, which we filter for queries whose answers link to a specific StackExchange post. As Khashabi et al. (2021) hypothesize, Google Search likely maps these natural queries to their answers by relying on a wide variety of signals for relevance, including expert annotations, user clicks, and hyperlinks as well as specialized QA components for various question types *with access to the post title and question body*. Using those annotations as ground truth, we evaluate the models on their capacity for retrieval using *only* free text of the answer posts (i.e., no hyperlinks or user clicks, question title or body, etc.), posing a significant challenge for IR and NLP systems trained only on public datasets.

Forum Queries. We collect the forum queries by extracting post titles from the StackExchange communities to use as queries and collect their corresponding answer posts as targets. We select questions in order of their popularity and sample questions according to the proportional contribution of individual communities within each topic.

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.

These queries tend to have a wider variety than the “search” queries, while the search queries may exhibit more natural patterns. Table 3 compares a random samples of search and forum queries. It can be seen that search queries tend to be brief, knowledge-based questions with direct answers, whereas forum queries tend to reflect more open-ended questions. Both query sets target topics that exceed the scope of a general-purpose knowledge repository such as Wikipedia.

For search as well as forum queries, the resulting evaluation set consists of a query and a target set of StackExchange answer posts (in particular, the answer posts from the target StackExchange page). Similar to evaluation in the Open-QA literature (Karpukhin et al., 2020; Khattab et al.,

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).

2021b), we evaluate retrieval quality by computing the success@5 (S@5) metric. Specifically, we award a point to the system for each query where it finds an accepted or upvoted (score ≥ 1) answer from the target page in the top-5 hits.

Appendix D reports on the breakdown of constituent communities per topic, the construction procedure of LoTTE as well as licensing considerations, and relevant statistics. Figures 5 and 6 quantitatively compare the search and forum queries.

5 Evaluation

We now evaluate ColBERTv2 on passage retrieval tasks, testing its quality within the training domain (§5.1) as well as outside the training domain in zero-shot settings (§5.2). Unless otherwise stated, we compress ColBERTv2 embeddings to $b = 2$ bits per dimension in our evaluation.

5.1 In-Domain Retrieval Quality

Similar to related work, we train for IR tasks on MS MARCO Passage Ranking (Nguyen et al., 2016). Within the training domain, our development-set results are shown in Table 4, comparing ColBERTv2 with vanilla ColBERT as well as state-of-the-art single-vector systems.

While ColBERT outperforms single-vector systems like RepBERT, ANCE, and even TAS-B, improvements in supervision such as distillation from cross-encoders enable systems like SPLADEv2,

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

Table 4: In-domain performance on the development set of MS MARCO Passage Ranking as well the “Local Eval” test set described by Khattab and Zaharia (2020). Dev-set results for baseline systems are from their respective papers: Zhan et al. (2020b), Xiong et al. (2020) for DPR and ANCE, Zhan et al. (2020a), Khattab and Zaharia (2020), Hofstätter et al. (2021), Gao and Callan (2021), Ren et al. (2021a), Formal et al. (2021a), and Ren et al. (2021b).

PAIR, and RocketQAv2 to achieve higher quality than vanilla ColBERT. These supervision gains challenge the value of fine-grained late interaction, and it is not inherently clear whether the stronger inductive biases of ColBERT-like models permit it to accept similar gains under distillation, especially when using compressed representations. Despite this, we find that with denoised supervision and residual compression, ColBERTv2 achieves the highest quality across all systems. As we discuss in §5.3, it exhibits space footprint competitive with these single-vector models and much lower than vanilla ColBERT.

Besides the official dev set, we evaluated ColBERTv2, SPLADEv2, and RocketQAv2 on the “Local Eval” test set described by Khattab and Zaharia (2020) for MS MARCO, which consists of 5000 queries disjoint with the training and the official dev sets. These queries are obtained from labeled 50k queries that are provided in the official MS MARCO Passage Ranking task as additional validation data.⁴ On this test set, ColBERTv2 obtains 40.8% MRR@10, considerably outperforming the baselines, including RocketQAv2 which makes use of document titles in addition to the passage text unlike the other systems.

⁴These are sampled from delta between qrels.dev.tsv and qrels.dev.small.tsv on <https://microsoft.github.io/msmarco/Datasets>. We refer to Khattab and Zaharia (2020) for details. All our query IDs will be made public to aid reproducibility.

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. \(2021a\)](#).

5.2 Out-of-Domain Retrieval Quality

Next, we evaluate ColBERTv2 outside the training domain using BEIR ([Thakur et al., 2021](#)), Wikipedia Open QA retrieval as in [Khattab et al. \(2021b\)](#), and LoTTE. We compare against a wide range of recent and state-of-the-art retrieval systems from the literature.

BEIR. We start with BEIR, reporting the quality of models that do not incorporate distillation from cross-encoders, namely, ColBERT ([Khattab and Zaharia, 2020](#)), DPR-MARCO ([Xin et al., 2021](#)), ANCE ([Xiong et al., 2020](#)), and MoDIR ([Xin et al., 2021](#)), as well as models that do utilize distillation, namely, TAS-B ([Hofstätter et al., 2021](#)), SPLADEv2 ([Formal et al., 2021a](#)), and also RocketQAv2, which we test ourselves using the official checkpoint trained on MS MARCO. We divide the table into “search” (i.e., natural queries and questions) and “semantic relatedness” (e.g., citation-relatedness and claim verification) tasks to reflect the nature of queries in each dataset.⁵

Table 5a reports results with the official nDCG@10 metric. Among the models with-

out distillation, we see that the vanilla ColBERT model outperforms the single-vector systems DPR, ANCE, and MoDIR across all but three tasks. ColBERT often outpaces all three systems by large margins and, in fact, outperforms the TAS-B model, which utilizes distillation, on most datasets. Shifting our attention to models with distillation, we see a similar pattern: while distillation-based models are generally stronger than their vanilla counterparts, the models that decompose scoring into term-level interactions, ColBERTv2 and SPLADEv2, are almost always the strongest.

Looking more closely into the comparison between SPLADEv2 and ColBERTv2, we see that ColBERTv2 has an advantage on six benchmarks and ties SPLADEv2 on two, with the largest improvements attained on NQ, TREC-COVID, and FiQA-2018, all of which feature natural search queries. On the other hand, SPLADEv2 has the lead on five benchmarks, displaying the largest gains on Climate-FEVER (C-FEVER) and HotPotQA. In C-FEVER, the input queries are sentences making climate-related claims and, as a result, do not reflect the typical characteristics of search queries. In HotPotQA, queries are written by crowdworkers who have access to the target pas-

⁵Following [Formal et al. \(2021a\)](#), we conduct our evaluation using the publicly-available datasets in BEIR. Refer to §E for details.

sages. This is known to lead to artificial lexical bias (Lee et al., 2019), where crowdworkers copy terms from the passages into their questions as in the Open-SQuAD benchmark.

Wikipedia Open QA. As a further test of out-of-domain generalization, we evaluate the MS MARCO-trained ColBERTv2, SPLADEv2, and vanilla ColBERT on retrieval for open-domain question answering, similar to the out-of-domain setting of Khattab et al. (2021b). We report Success@5 (sometimes referred to as Recall@5), which is the percentage of questions whose short answer string overlaps with one or more of the top-5 passages. For the queries, we use the development set questions of the open-domain versions (Lee et al., 2019; Karpukhin et al., 2020) of Natural Questions (NQ; Kwiatkowski et al. 2019), TriviaQA (TQ; Joshi et al. 2017), and SQuAD (Rajpurkar et al., 2016) datasets in Table 5b. As a baseline, we include the BM25 (Robertson et al., 1995) results using the Anserini (Yang et al., 2018a) toolkit. We observe that ColBERTv2 outperforms BM25, vanilla ColBERT, and SPLADEv2 across the three query sets, with improvements of up to 4.6 points over SPLADEv2.

LoTTE. Next, we analyze performance on the LoTTE test benchmark, which focuses on natural queries over long-tail topics and exhibits a different annotation pattern to the datasets in the previous OOD evaluations. In particular, LoTTE uses automatic Google rankings (for the “search” queries) and organic StackExchange question–answer pairs (for “forum” queries), complimenting the pooling-based annotation of datasets like TREC-COVID (in BEIR) and the answer overlap metrics of Open-QA retrieval. We report Success@5 for each corpus on both search queries and forum queries.

Overall, we see that ANCE and vanilla ColBERT outperform BM25 on all topics, and that the three methods using distillation are generally the strongest. Similar to the Wikipedia-OpenQA results, we find that ColBERTv2 outperforms the baselines across all topics for both query types, improving upon SPLADEv2 and RocketQAv2 by up to 3.7 and 8.1 points, respectively. Considering the baselines, we observe that while RocketQAv2 tends to have a slight advantage over SPLADEv2 on the “search” queries, SPLADEv2 is considerably more effective on the “forum” tests. We hypothesize that the search queries, obtained from Google (through GooAQ) are more similar to MS

MARCO than the forum queries and, as a result, the latter stresses generalization more heavily, rewarding term-decomposed models like SPLADEv2 and ColBERTv2.

5.3 Efficiency

ColBERTv2’s residual compression approach significantly reduces index sizes compared to vanilla ColBERT. Whereas ColBERT requires 154 GiB to store the index for MS MARCO, ColBERTv2 only requires 16 GiB or 25 GiB when compressing embeddings to 1 or 2 bit(s) per dimension, respectively, resulting in compression ratios of 6–10 \times . This storage figure includes 4.5 GiB for storing the inverted list.

This matches the storage for a typical single-vector model on MS MARCO, with 4-byte lossless floating-point storage for one 768-dimensional vector for each of the 9M passages amounting to a little over 25 GiBs. In practice, the storage for a single-vector model could be even larger when using a nearest-neighbor index like HNSW for fast search. Conversely, single-vector representations could be themselves compressed very aggressively (Zhan et al., 2021a, 2022), though often exacerbating the loss in quality relative to late interaction methods like ColBERTv2.

We discuss the impact of our compression method on search quality in Appendix B and present query latency results on the order of 50–250 milliseconds per query in Appendix C.

6 Conclusion

We introduced ColBERTv2, a retriever that advances the quality and space efficiency of multi-vector representations. We hypothesized that cluster centroids capture context-aware semantics of the token-level representations and proposed a residual representation that leverages these patterns to dramatically reduce the footprint of multi-vector systems *off-the-shelf*. We then explored improved supervision for multi-vector retrieval and found that their quality improves considerably upon distillation from a cross-encoder system. The proposed ColBERTv2 considerably outperforms existing retrievers in within-domain and out-of-domain evaluations, which we conducted extensively across 28 datasets, establishing state-of-the-art quality while exhibiting competitive space footprint.

Acknowledgements

This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, and VMware—as well as Cisco, SAP, Virtusa, and the NSF under CAREER grant CNS-1651570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Broader Impact & Ethical Considerations

This work is primarily an effort toward retrieval models that generalize better while performing reasonably efficiently in terms of space consumption. Strong out-of-the-box generalization to small domain-specific applications can serve many users in practice, particularly where training data is not available. Moreover, retrieval holds significant promise for many downstream NLP tasks, as it can help make language models smaller and thus more efficient (i.e., by decoupling knowledge from computation), more transparent (i.e., by allowing users to check the sources the model relied on when making a claim or prediction), and easier to update (i.e., by allowing developers to replace or add documents to the corpus without retraining the model) (Guu et al., 2020; Borgeaud et al., 2021; Khattab et al., 2021a). Nonetheless, such work poses risks in terms of misuse, particularly toward misinformation, as retrieval can surface results that are relevant yet inaccurate, depending on the contents of a corpus. Moreover, generalization from training on a large-scale dataset can propagate the biases of that dataset well beyond its typical reach to new domains and applications.

While our contributions have made ColBERT’s late interaction more efficient at storage costs, large-scale distillation with hard negatives increases system complexity and accordingly increases training cost, when compared with the straightforward training paradigm of the original ColBERT model. While ColBERTv2 is efficient in terms of latency and storage at inference time, we suspect that under extreme resource constraints, simpler model designs like SPLADEv2 or RocketQAv2 could lend themselves to easier-to-optimize environments. We leave low-level systems optimizations of all systems to future work. Another worthwhile dimension for future exploration of tradeoffs is re-ranking architectures over various systems with

cross-encoders, which are known to be expensive yet precise due to their highly expressive capacity.

Research Limitations

While we evaluate ColBERTv2 on a wide range of tests, all of our benchmarks are in English and, in line with related work, our out-of-domain tests evaluate models that are trained on MS MARCO. We expect our approach to work effectively for other languages and when all models are trained using other, smaller training set (e.g., NaturalQuestions), but we leave such tests to future work.

We have observed consistent gains for ColBERTv2 against existing state-of-the-art systems across many diverse settings. Despite this, almost all IR datasets contain false negatives (i.e., relevant but unlabeled passages) and thus some caution is needed in interpreting any individual result. Nonetheless, we intentionally sought out benchmarks with dissimilar annotation biases: for instance, TREC-COVID (in BEIR) annotates the pool of documents retrieved by the systems submitted at the time of the competition, LoTTE uses automatic Google rankings (for “search” queries) and StackExchange question–answer pairs (for “forum” queries), and the Open-QA tests rely on passage–answer overlap for factoid questions. ColBERTv2 performed well in all of these settings. We discuss other issues pertinent to LoTTE in Appendix §D.

We have compared with a wide range of strong baselines—including sparse retrieval and single-vector models—and found reliable patterns across tests. However, we caution that empirical trends can change as innovations are introduced to each of these families of models and that it can be difficult to ensure exact apple-to-apple comparisons across families of models, since each of them calls for different sophisticated tuning strategies. We thus primarily used results and models from the rich recent literature on these problems, with models like RocketQAv2 and SPLADEv2.

On the representational side, we focus on reducing the storage cost using residual compression, achieving strong gains in reducing footprint while largely preserving quality. Nonetheless, we have not exhausted the space of more sophisticated optimizations possible, and we would expect more sophisticated forms of residual compression and composing our approach with dropping tokens (Zhou and Devlin, 2021) to open up possibilities for further reductions in space footprint.

References

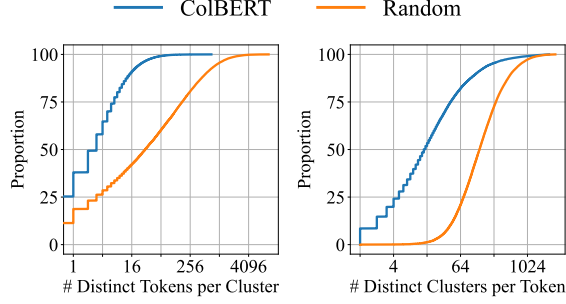
[Stack Exchange Data Dump](#).

- Liefu Ai, Junqing Yu, Zebin Wu, Yunfeng He, and Tao Guan. 2017. Optimized Residual Vector Quantization for Efficient Approximate Nearest Neighbor Search. *Multimedia Systems*, 23(2):169–181.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The semantic web*, pages 722–735. Springer.
- Christopher F Barnes, Syed A Rizvi, and Nasser M Nasrabadi. 1996. Advances in Residual Vector Quantization: A Review. *IEEE transactions on image processing*, 5(2):226–262.
- Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, et al. 2020. Overview of touché 2020: Argument Retrieval. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 384–395. Springer.
- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2021. [Improving language models by retrieving from trillions of tokens](#). *arXiv preprint arXiv:2112.04426*.
- Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-text Learning to Rank Dataset for Medical Information Retrieval. In *European Conference on Information Retrieval*, pages 716–722. Springer.
- Chia-Yu Chen, Jungwook Choi, Daniel Brand, Ankur Agrawal, Wei Zhang, and Kailash Gopalakrishnan. 2018. [Adacomp : Adaptive residual gradient compression for data-parallel distributed training](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2827–2835. AAAI Press.
- Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. [SPECTER: Document-level representation learning using citation-informed transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.
- Nachshon Cohen, Amit Portnoy, Besnik Fetahu, and Amir Ingber. 2021. [SDR: Efficient Neural Re-ranking using Succinct Document Representation](#). *arXiv preprint arXiv:2110.02065*.
- Zhuyun Dai and Jamie Callan. 2020. [Context-aware term weighting for first stage passage retrieval](#). In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1533–1536. ACM.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. [CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims](#). *arXiv preprint arXiv:2012.00614*.
- Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. [SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval](#). *arXiv preprint arXiv:2109.10086*.
- Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. [SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.
- Luyu Gao and Jamie Callan. 2021. [Unsupervised corpus aware language model pre-training for dense passage retrieval](#). *arXiv preprint arXiv:2108.05540*.
- Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. [Modularized transformer-based ranking framework](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4180–4190, Online. Association for Computational Linguistics.
- Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. [COIL: Revisit exact lexical match in information retrieval with contextualized inverted list](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, Online. Association for Computational Linguistics.
- Robert Gray. 1984. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Realm: Retrieval-augmented language model pre-training](#). *arXiv preprint arXiv:2002.08909*.

- Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. [Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation](#). *arXiv preprint arXiv:2010.02666*.
- Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. [Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling](#). *arXiv preprint arXiv:2104.06967*.
- Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. [Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. 2020. [A memory efficient baseline for open domain question answering](#). *arXiv preprint arXiv:2012.15156*.
- Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128.
- Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. [HoVer: A dataset for many-hop fact extraction and claim verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. [GooAQ: Open Question Answering with Diverse Answer Types](#). *arXiv preprint arXiv:2104.08727*.
- Omar Khattab, Christopher Potts, and Matei Zaharia. 2021a. [Baleen: Robust Multi-Hop Reasoning at Scale via Condensed Retrieval](#). In *Thirty-Fifth Conference on Neural Information Processing Systems*.
- Omar Khattab, Christopher Potts, and Matei Zaharia. 2021b. [Relevance-guided supervision for openqa with ColBERT](#). *Transactions of the Association for Computational Linguistics*, 9:929–944.
- Omar Khattab and Matei Zaharia. 2020. [Colbert: Efficient and effective passage search via contextualized late interaction over BERT](#). In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021a. [Learning dense representations of phrases at scale](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.
- Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021b. [Phrase retrieval learns passage retrieval, too](#). *arXiv preprint arXiv:2109.08133*.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- Yue Li, Wenrui Ding, Chunlei Liu, Baochang Zhang, and Guodong Guo. 2021a. [TRQ: Ternary Neural Networks With Residual Quantization](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8538–8546.
- Zefan Li, Bingbing Ni, Teng Li, Xiaokang Yang, Wenjun Zhang, and Wen Gao. 2021b. [Residual Quantization for Low Bit-width Neural Networks](#). *IEEE Transactions on Multimedia*.
- Jimmy Lin and Xueguang Ma. 2021. [A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques](#). *arXiv preprint arXiv:2106.14807*.

- Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. [Distilling Dense Representations for Ranking using Tightly-Coupled Teachers](#). *arXiv preprint arXiv:2010.11386*.
- Xiaorui Liu, Yao Li, Jiliang Tang, and Ming Yan. 2020. [A double residual compression algorithm for efficient distributed learning](#). In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 133–143. PMLR.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.
- Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. [Efficient document re-ranking for transformers by precomputing term representations](#). In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 49–58. ACM.
- Craig Macdonald and Nicola Tonellotto. 2021. On approximate nearest neighbour selection for multi-stage dense retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3318–3322.
- Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW’18 Open Challenge: Financial Opinion Mining and Question Answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1941–1942.
- Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1723–1727.
- Aditya Krishna Menon, Sadeep Jayasumana, Seungyeon Kim, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2022. [In defense of dual-encoders for neural ranking](#).
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. [MS MARCO: A human-generated MACHINE reading COMprehension dataset](#). *arXiv preprint arXiv:1611.09268*.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. [Passage Re-ranking with BERT](#). *arXiv preprint arXiv:1901.04085*.
- Barlas Oğuz, Kushal Lakhota, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. [Domain-matched Pre-training Tasks for Dense Retrieval](#). *arXiv preprint arXiv:2107.13602*.
- Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. 2022. [Hindsight: Posterior-guided training of retrievers for improved open-ended generation](#). In *International Conference on Learning Representations*.
- Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. [RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. [PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183, Online. Association for Computational Linguistics.
- Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. [RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking](#). *arXiv preprint arXiv:2110.07367*.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *NIST Special Publication*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. [BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models](#). *arXiv preprint arXiv:2104.08663*.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

- Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection. In *ACM SIGIR Forum*, volume 54, pages 1–12. ACM New York, NY, USA.
- Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. [Retrieval of the best counterargument without prior topic knowledge](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or fiction: Verifying scientific claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. [MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers](#). *arXiv preprint arXiv:2002.10957*.
- Benchang Wei, Tao Guan, and Junqing Yu. 2014. Projected Residual Vector Quantization for ANN Search. *IEEE multimedia*, 21(3):41–51.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul N Bennett. 2021. [Zero-Shot Dense Retrieval with Momentum Adversarial Domain Invariant Representations](#). *arXiv preprint arXiv:2110.07581*.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.
- Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021a. [Efficient passage retrieval with hashing for open-domain question answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online. Association for Computational Linguistics.
- Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021b. [Efficient passage retrieval with hashing for open-domain question answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online. Association for Computational Linguistics.
- Peilin Yang, Hui Fang, and Jimmy Lin. 2018a. Anserini: Reproducible ranking baselines using lucene. *Journal of Data and Information Quality (JDIQ)*, 10(4):1–20.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021a. Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2487–2496.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021b. Optimizing Dense Retrieval Model Training with Hard Negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2022. [Learning discrete representations via constrained clustering for effective and efficient dense retrieval](#). In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22*, page 1328–1336. Association for Computing Machinery.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020a. [Learning to retrieve: How to train a dense retrieval model effectively and efficiently](#). *arXiv preprint arXiv:2010.10469*.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020b. [Repbert: Contextualized text embeddings for first-stage retrieval](#). *arXiv preprint arXiv:2006.15498*.
- Giulio Zhou and Jacob Devlin. 2021. [Multi-vector attention models for deep re-ranking](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5452–5456, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.



(a) Number of distinct tokens appearing in each cluster. (b) Number of distinct clusters each token appears in.

Figure 2: Empirical CDFs analyzing semantic properties of MS MARCO token-level embeddings both encoded by ColBERT and randomly generated. The embeddings are partitioned into 2^{18} clusters and correspond to roughly 27,000 distinct tokens.

A Analysis of ColBERT’s Semantic Space

ColBERT (Khattab and Zaharia, 2020) decomposes representations and similarity computation at the token level. Because of this compositional architecture, we hypothesize that ColBERT exhibits a “lightweight” semantic space: without any special re-training, vectors corresponding to each sense of a word would cluster very closely, with only minor variation due to context.

If this hypothesis is true, we would expect the embeddings corresponding to each token in the vocabulary to localize in only a small number of regions in the embedding space, corresponding to the contextual “senses” of the token. To validate this hypothesis, we analyze the ColBERT embeddings corresponding to the tokens in the MS MARCO Passage Ranking (Nguyen et al., 2016) collection: we perform k -means clustering on the nearly 600M embeddings—corresponding to 27,000 unique tokens—into $k = 2^{18}$ clusters. As a baseline, we repeat this clustering with random embeddings but keep the true distribution of tokens. Figure 2 presents empirical cumulative distribution function (eCDF) plots representing the number of distinct non-stopword tokens appearing in each cluster (2a) and the number of distinct clusters in which each token appears (2b).⁶ Most tokens appear in a very small fraction of the number of centroids: in particular, we see that roughly 90% of clusters have ≤ 16 distinct tokens with

⁶We rank tokens by number of clusters they appear in and designate the top-1% (under 300) as stopwords.

the ColBERT embeddings, whereas less than 50% of clusters have ≤ 16 distinct tokens with the random embeddings. This suggests that the centroids effectively map the ColBERT semantic space.

Table 6 presents examples to highlight the semantic space captured by the centroids. The most frequently appearing tokens in cluster #917 relate to photography; these include, for example, ‘photos’ and ‘photographs’. If we then examine the additional clusters in which these tokens appear, we find that there is substantial semantic overlap between these new clusters (e.g., Photos-Photo, Photo-Image-Picture) and cluster #917. We observe a similar effect with tokens appearing in cluster #216932, comprising tornado-related terms.

This analysis indicates that cluster centroids can summarize the ColBERT representations with high precision. In §3.3, we propose a residual compression mechanism that uses these centroids along with minor refinements at the dimension level to efficiently encode late-interaction vectors.

B Impact of Compression

Our residual compression approach (§3.3) preserves approximately the same quality as the uncompressed embeddings. In particular, when applied to a vanilla ColBERT model on MS MARCO whose MRR@10 is 36.2% and Recall@50 is 82.1%, the quality of the model with 2-bit compression is 36.2% MRR@10 and 82.3% Recall@50. With 1-bit compression, the model achieves 35.5% MRR@10 and 81.6% Recall@50.⁷

We also tested the residual compression approach on late-interaction retrievers that conduct downstream tasks, namely, ColBERT-QA (Khattab et al., 2021b) for the NaturalQuestions open-domain QA task, and Baleen (Khattab et al., 2021a) for multi-hop reasoning on HoVer for claim verification. On the NQ dev set, ColBERT-QA’s success@5 (success@20) dropped only marginally from 75.3% (84.3%) to 74.3% (84.2%) and its downstream Open-QA answer exact match dropped from 47.9% to 47.7%, when using 2-bit compression for retrieval and using the same checkpoints of ColBERT-QA otherwise.

⁷We contrast this with an early implementation of compression for ColBERT, which used binary representations as in BPR (Yamada et al., 2021a) without residual centroids, and achieves 34.8% (35.7%) MRR@10 and 80.5% (81.8%) Recall@50 with 1-bit (2-bit) binarization. Like the original ColBERT, this form of compression relied on a separate FAISS index for candidate generation.

Cluster ID	Most Common Tokens	Most Common Clusters Per Token	
		Token	Clusters
917	'photos', 'photo', 'pictures', 'photographs', 'images', 'photography', 'photograph'	'photos'	Photos-Photo, Photos-Pictures-Photo
		'photo'	Photo-Image-Picture, Photo-Picture-Photograph, Photo-Picture-Photography
		'pictures'	Pictures-Picture-Images, Picture-Pictures-Artists, Pictures-Photo-Picture
216932	'tornado', 'tornadoes', 'storm', 'hurricane', 'storms'	'tornado'	Tornado-Hurricane-Storm, Tornadoes-Tornado-Blizzard
		'tornadoes'	Tornadoes-Tornado-Storms, Tornadoes-Tornado-Blizzard, Tornado-Hurricane-Storm
		'storm'	Storm-Storms, Storm-Storms-Weather, Storm-Storms-Tempest

Table 6: Examples of clusters taken from all MS MARCO passages. We present the tokens that appear most frequently in the selected clusters as well as additional clusters the top tokens appear in.

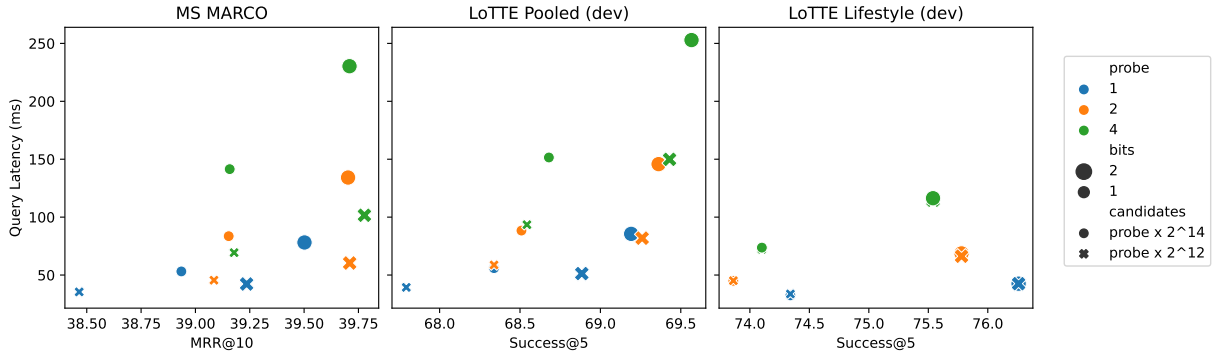


Figure 3: Latency vs. retrieval quality with varying parameter configurations for three datasets of different collection sizes. We sweep a range of values for the number of centroids per vector (probe), the number of bits used for residual compression, and the number of candidates. Note that retrieval quality is measured in MRR@10 for MS MARCO and Success@5 for LoTTE datasets. Results toward the bottom right corner (higher quality, lower latency) are best.

Similarly, on the HoVer (Jiang et al., 2020) dev set, Baleen’s retrieval R@100 dropped from 92.2% to only 90.6% but its sentence-level exact match remained roughly the same, going from 39.2% to 39.4%. We hypothesize that the supervision methods applied in ColBERTv2 (§3.2) can also be applied to lift quality in downstream tasks by improving the recall of retrieval for these tasks. We leave such exploration for future work.

C Retrieval Latency

Figure 3 evaluates the latency of ColBERTv2 across three collections of varying sizes, namely, MS MARCO, LoTTE Pooled (dev), and LoTTE Lifestyle (dev), which contain approximately 9M passages, 2.4M answer posts, and 270k answer posts, respectively. We average latency across three runs of the MS MARCO dev set and the LoTTE “search” queries. Search is executed using a Titan V GPU on a server with two Intel Xeon Gold 6132 CPUs, each with 28 hardware execution contexts.

The figure varies three settings of ColBERTv2. In particular, we evaluate indexing with 1-bit and 2-bit encoding (§3.4) and searching by probing the nearest 1, 2, or 4 centroids to each query vector (§3.5). When probing probe centroids per vector, we score either $\text{probe} \times 2^{12}$ or $\text{probe} \times 2^{14}$ candidates per query.⁸

To begin with, we notice that the quality reported on the x -axis varies only within a relatively narrow range. For instance, the axis ranges from 38.50 through 39.75 for MS MARCO, and all but two of the cheapest settings score above 39.00. Similarly, the y -axis varies between approximately 50 milliseconds per query up to 250 milliseconds (mostly under 150 milliseconds) using our relatively simple Python-based implementation.

Digging deeper, we see that the best quality in these metrics can be achieved or approached closely with around 100 milliseconds of latency across all three datasets, despite their various sizes and characteristics, and that 2-bit indexing reliably outperforms 1-bit indexing but the loss from more aggressive compression is small.

D LoTTE

Domain coverage Table 9 presents the full distribution of communities in the LoTTE dev dataset.

⁸These settings are selected based on preliminary exploration of these parameters, which indicated that performance for larger probe values tends to require scoring a larger number of candidates.

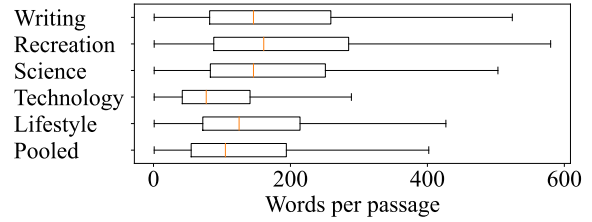


Figure 4: LoTTE words per passage

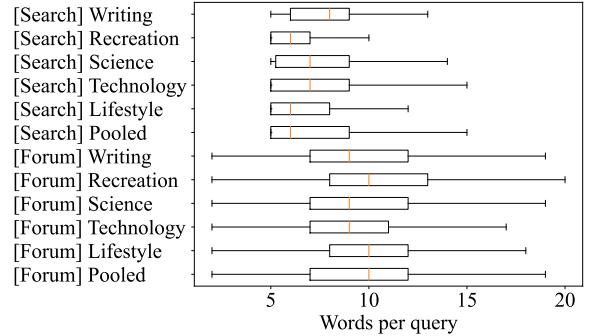


Figure 5: LoTTE words per query

The topics covered by LoTTE cover a wide range of linguistic phenomena given the diversity in topics and communities represented. However, since all posts are submitted by anonymous users we do not have demographic information regarding the identity of the contributors. All posts are written in English.

Passages As mentioned in §4, we construct LoTTE collections by selecting passages from the StackExchange archive with positive scores. We remove HTML tags from passages and filter out empty passages. For each passage we record its corresponding query and save the query-to-passage mapping to keep track of the posted answers corresponding to each query.

Search queries We construct the list of LoTTE search queries by drawing from GooAQ queries that appear in the StackExchange post archive. We first shuffle the list of GooAQ queries so that in cases where multiple queries exist for the same answer passage we randomly select the query to include in LoTTE rather than always selecting the first appearing query. We verify that every query has at least one corresponding answer passage.

Forum queries For each LoTTE topic and its constituent communities we first compute the fraction of the total queries attributed to each individual community. We then use this distribution to construct a truncated query set by selecting the

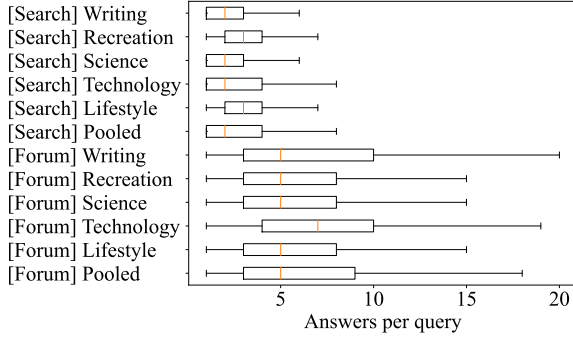


Figure 6: LoTTE answers per query

Corpus	ColBERT	BM25	ANCE	RocketQAv2	SPLADEv2	ColBERTv2
LoTTE Search Dev Queries (Success@5)						
Writing	76.3	47.3	75.7	79.5	78.9	81.7
Recreation	71.8	56.3	66.1	73.0	70.7	76.0
Science	71.7	52.2	66.9	67.7	73.4	74.2
Technology	52.8	35.8	55.7	54.3	56.3	59.3
Lifestyle	73.1	54.4	69.8	72.4	71.2	75.8
Pooled	65.4	45.6	63.7	66.4	67.0	69.3
LoTTE Forum Dev Queries (Success@5)						
Writing	75.5	66.2	74.4	75.5	78.1	80.8
Recreation	69.1	56.6	65.9	69.0	68.9	71.8
Science	58.2	51.3	56.3	56.7	59.9	62.6
Technology	39.6	30.7	38.8	39.9	42.1	45.0
Lifestyle	61.1	48.2	61.8	62.0	61.8	65.8
Pooled	59.1	47.8	57.4	58.9	60.6	63.7

Table 7: Zero-shot evaluation results on the dev sets of the LoTTE benchmark.

highest ranked queries from each community as determined by 1) the query scores and 2) the query view counts. We only use queries which have an accepted answer. We ensure that each community contributes at least 50 queries to the truncated set whenever possible. We set the overall size of the truncated set to be 2000 queries, though note that the total can exceed this due to rounding and/or the minimum per-community query count. We remove all quotation marks and HTML tags.

Statistics Figure 4 plots the number of words per passage in each LoTTE dev corpus. Figures 5 and 6 plot the number of words and number of corresponding answer passages respectively per query, split across search and forum queries.

Dev Results Table 7 presents out-of-domain evaluation results on the LoTTE dev queries. Continuing the trend we observed in 5, ColBERTv2 consistently outperforms all other models we tested.

Licensing and Anonymity The original Stack-Exchange post archive is licensed under a Creative Commons BY-SA 4.0 license (sta). Personal data is removed from the archive before being uploaded, though all posts are public; when we release LoTTE publicly we will include URLs to the original posts for proper attribution as required by the license. The GooAQ dataset is licensed under an Apache license, version 2.0 (Khashabi et al., 2021). We will also release LoTTE with a CC BY-SA 4.0 license. The search queries can be used for non-commercial research purposes only as per the GooAQ license.

E Datasets in BEIR

Table 8 lists the BEIR datasets we used in our evaluation, including their respective license information as well as the numbers of documents as well as the number of test set queries. We refer to Thakur et al. (2021) for a more detailed description of each of the datasets.

Our Touché evaluation uses an updated version of the data in BEIR, which we use for evaluating the models we run (i.e., ColBERTv2 and RocketQAv2) as well as SPLADEv2.

Dataset	License	# Passages	# Test Queries
ArguAna (Wachsmuth et al., 2018)	CC BY 4.0	8674	1406
Climate-Fever (Diggelmann et al., 2020)	Not reported	5416593	1535
DBPedia (Auer et al., 2007)	CC BY-SA 3.0	4635922	400
FEVER (Thorne et al., 2018)	CC BY-SA 3.0		
FiQA-2018 (Maia et al., 2018)	Not reported	57638	648
HotpotQA (Yang et al., 2018b)	CC BY-SA 4.0	5233329	7405
NFCorpus (Boteva et al., 2016)	Not reported	3633	323
NQ (Kwiatkowski et al., 2019)	CC BY-SA 3.0	2681468	3452
SCIDOCS (Cohan et al., 2020)	GNU General Public License v3.0	25657	1000
SciFact (Wadden et al., 2020)	CC BY-NC 2.0	5183	300
Quora	Not reported	522931	10000
Touché-2020 (Bondarenko et al., 2020)	CC BY 4.0	382545	49
TREC-COVID (Voorhees et al., 2021)	Dataset License Agreement	171332	50

Table 8: BEIR dataset information.

We also tested on the Open-QA benchmarks NQ, TQ, and SQuAD, each of which has approximately 9k dev-set questions and multi-hop HoVer, whose development set has 4k claims. In the compression evaluation §B, we used models trained in-domain on NQ and HoVer, whose training sets contain 79k and 18k queries, respectively.

F Implementation & Hyperparameters

We implement ColBERTv2 using Python 3.7, PyTorch 1.9, and HuggingFace Transformers 4.10 (Wolf et al., 2020), extending the original implementation of ColBERT by Khattab and Zaharia (2020). We use FAISS 1.7 (Johnson et al., 2019) for

k -means clustering,⁹ though unlike ColBERT we do not use it for nearest-neighbor search. Instead, we implement our candidate generation mechanism (§3.5) using PyTorch primitives in Python.

We conducted our experiments on an internal cluster, typically using up to four 12GB Titan V GPUs for each of the inference tasks (e.g., indexing, computing distillation scores, and retrieval) and four 80GB A100 GPUs for training, though GPUs with smaller RAM can be used via gradient accumulation. Using this infrastructure, computing the distillation scores takes under a day, training a 64-way model on MS MARCO for 400,000 steps takes around five days, and indexing takes approximately two hours. We very roughly estimate an upper bound total of 20 GPU-months for all experimentation, development, and evaluation performed for this work over a period of several months.

Like ColBERT, our encoder is a bert-base-uncased model that is shared between the query and passage encoders and which has 110M parameters. We retain the default vector dimension suggested by Khattab and Zaharia (2020) and used in subsequent work, namely, $d=128$. For the experiments reported in this paper, we train on MS MARCO training set. We use simple defaults with limited manual exploration on the official development set for the learning rate (10^{-5}), batch size (32 examples), and warm up (for 20,000 steps) with linear decay.

Hyperparameters corresponding to retrieval are explored in §C. We default to $\text{probe} = 2$, but use $\text{probe} = 4$ on the largest datasets, namely, MS MARCO and Wikipedia. By default we set $\text{candidates} = \text{probe} * 2^{12}$, but for Wikipedia we set $\text{candidates} = \text{probe} * 2^{13}$ and for MS MARCO we set $\text{candidates} = \text{probe} * 2^{14}$. We leave extensive tuning of hyperparameters to future work.

We train on MS MARCO using 64-way tuples for distillation, sampling them from the top-500 retrieved passages per query. The training set of MS MARCO contains approximately 800k queries, though only about 500k have associated labels. We apply distillation using all 800k queries, where each training example contains exactly one “positive”, defined as a passage labeled as positive or the top-ranked passage by the cross-encoder teacher, irrespective of its label.

We train for 400k steps, initializing from a pre-

finetuned checkpoint using 32-way training examples and 150k steps. To generate the top- k passages per training query, we apply two rounds, following Khattab et al. (2021b). We start from a model trained with hard triples (akin to Khattab et al. (2021b)), train with distillation, and then use the distilled model to retrieve for the second round of training. Preliminary experiments indicate that quality has low sensitivity to this initialization and two-round training, suggesting that both of them could be avoided to reduce the cost of training.

Unless otherwise stated, the results shown represent a single run. The latency results in §3 are averages of three runs. To evaluate for Open-QA retrieval, we use evaluation scripts from Khattab et al. (2021b), which checks if the short answer string appears in the (titled) Wikipedia passage. This adapts the DPR (Karpukhin et al., 2020) evaluation code.¹⁰ We use the preprocessed Wikipedia Dec 2018 dump released by Karpukhin et al. (2020).

For out-of-domain evaluation, we elected to follow Thakur et al. (2021) and set the maximum document length of ColBERT, RocketQAv2, and ColBERTv2 to 300 tokens on BEIR and LoTTE. Formal et al. (2021a) selected maximum sequence length 256 for SPLADEv2 both on MS MARCO and on BEIR for both queries and documents, and we retained this default when testing their system on LoTTE. Unless otherwise stated, we keep the default query maximum sequence length for ColBERTv2 and RocketQAv2, which is 32 tokens. For the ArguAna test in BEIR, as the queries are themselves long documents, we set the maximum query length used by ColBERTv2 and RocketQAv2 to 300. For Climate-FEVER, as the queries are relatively long sentence claims, we set the maximum query length used by ColBERTv2 to 64.

We use the open source BEIR implementation¹¹ and SPLADEv2 evaluation¹² code as the basis for our evaluations of SPLADEv2 and ANCE as well as for BM25 on LoTTE. We use the Anserini (Yang et al., 2018a) toolkit for BM25 on the Wikipedia Open-QA retrieval tests as in Khattab et al. (2021b). We use the implementation developed by the RocketQAv2 authors for evaluating RocketQAv2.¹³

¹⁰https://github.com/facebookresearch/DPR/blob/main/dpr/data/qa_validation.py

¹¹<https://github.com/UKPLab/beir>

¹²<https://github.com/naver/splade>

¹³<https://github.com/PaddlePaddle/RocketQA>

⁹<https://github.com/facebookresearch/faiss>

Topic	Communities	# Passages	# Search queries	# Forum queries
Writing	ell.stackexchange.com	108143	433	1196
	literature.stackexchange.com	4778	7	58
	writing.stackexchange.com	29330	23	163
	linguistics.stackexchange.com	12302	22	116
	worldbuilding.stackexchange.com	122519	12	470
Recreation	rpg.stackexchange.com	89066	91	621
	boardgames.stackexchange.com	20340	67	179
	scifi.stackexchange.com	102561	343	852
	photo.stackexchange.com	51058	62	350
Science	chemistry.stackexchange.com	39435	245	267
	stats.stackexchange.com	144084	137	949
	academia.stackexchange.com	76450	66	302
	astronomy.stackexchange.com	14580	15	88
	earthscience.stackexchange.com	6734	10	50
	engineering.stackexchange.com	12064	16	77
	datascience.stackexchange.com	23234	15	156
	philosophy.stackexchange.com	27061	34	124
Technology	superuser.com	418266	441	648
	electronics.stackexchange.com	205891	118	314
	askubuntu.com	296291	132	480
	serverfault.com	323943	148	506
	webapps.stackexchange.com	31831	77	55
Lifestyle	pets.stackexchange.com	10070	20	87
	lifehacks.stackexchange.com	7893	2	50
	gardening.stackexchange.com	20601	16	182
	parenting.stackexchange.com	18357	10	87
	crafts.stackexchange.com	3094	4	50
	outdoors.stackexchange.com	13324	16	76
	coffee.stackexchange.com	2249	11	50
	music.stackexchange.com	47399	65	287
	diy.stackexchange.com	82659	135	732
	bicycles.stackexchange.com	35567	40	229
	mechanics.stackexchange.com	27680	98	246

Table 9: Per-community distribution of LoTTE dev dataset passages and questions.