Introduction to Information Retrieval Systems

Computer Science Dept, Yonsei University

Seungone Kim

Referenced Papers

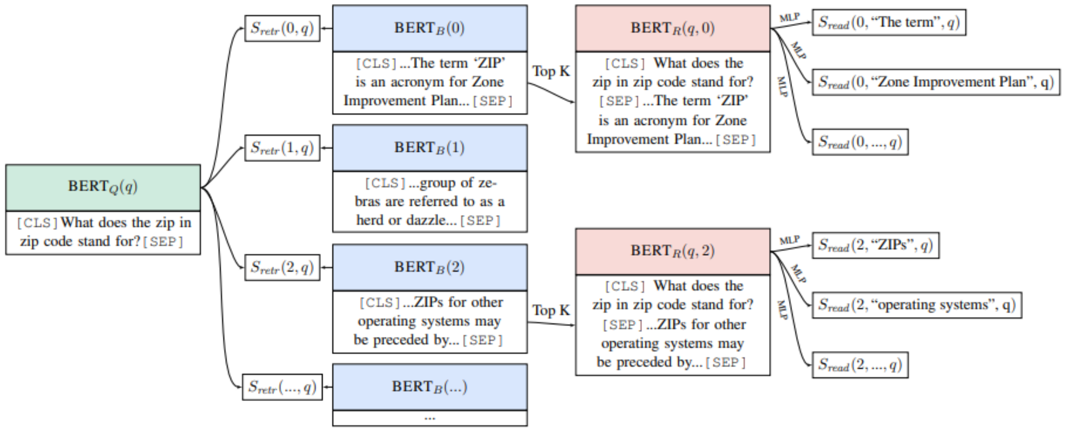
- Prerequisites
  - Reading Wikipedia to Answer Open-Domain Questions (DrQA, ACL 2017)
  - Latent Retrieval for Weakly Supervised Open Domain Question Answering (OrQA, ACL 2019)
  - REALM : Retrieval-Augmented Language Model Pretraining (PMLR 2020 Poster)

- Key Papers
  - ColBERT : Efficient and Effective Passage Search via Contextualized Late Interaction over BERT (SIGIR 2020)
  - RepBERT : Contextualized Text Embedding for First-Stage Retrieval (arXiv, Tsinghua Univ)
  - Dense Passage Retrieval for Open-Domain Question Answering(DPR, EMNLP 2020)
  - COIL : Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List (NAACL 2021)

- Next Week
  - Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks(arXiv, Facebook, London Univ, New York Univ) -> NeurIPS 2020

What is Information Retrieval?

- Short review on prerequisites
  - Information Retrieval is a task of narrowing the search space of documents to only focus on reading documents that are likely relevant to the given query.
  - DrQA uses a non-machine learning retriever and a neural reader(GRU) where only the reader is trained.
  - OrQA uses an end-to-end model where retriever(dense-retriever) and reader component(transformer) are jointly learned.
  - REALM is similar to OrQA that retriever and reader are jointly trained.
  - While OrQA uses Inverse Cloze Task to train the system, REALM uses Masked Language Modeling pre-training.



Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

What is Information Retrieval?

- Classic Approaches
  - BM25 is a ranking algorithm that measures the relevance between a document and a given query using term-based approach.

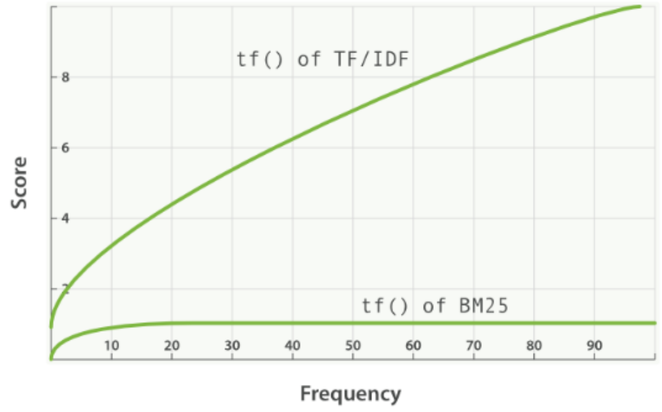
$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{\overbrace{f(q_i,D)}^{\text{문서 }D\text{에서 }q_i\text{의 term frequency}} * (k_1+1)}{f(q_i,D) + \underbrace{k_1}_{\text{파라메터}} * (1 - \underbrace{b}_{\text{파라메터}} + b * \frac{\overbrace{|D|}^{\text{문서 }D\text{ 의 길이}}}{\underbrace{avgdl}_{\text{문서 집합의 평균 문서 길이}}})}$$

$$IDF(q_i) = \ln(1 + \frac{(\overbrace{docCount}^{\text{총 문서의 개수}} - f(q_i) + 0.5)}{\underbrace{f(q_i)}_{\text{해당 단어를 포함하는 문서의 개수}} + 0.5})$$

- q_i refers to the i-th token within the query (Could use Tokens or words)
- IDF is the inverse document frequency of q_i => e.g. 10개의 문서에서 1번만 등장하면 1.99, 10번 모두 등장하면 0.05
- f(q_i, D) refers to how often the i-th token appears within the document D

- k1 is a hyperparameter (usually 1.2~2.0) that restricts the impact a single token could affect the score
- k1 decides how much the score should get increased if a term appears more than once

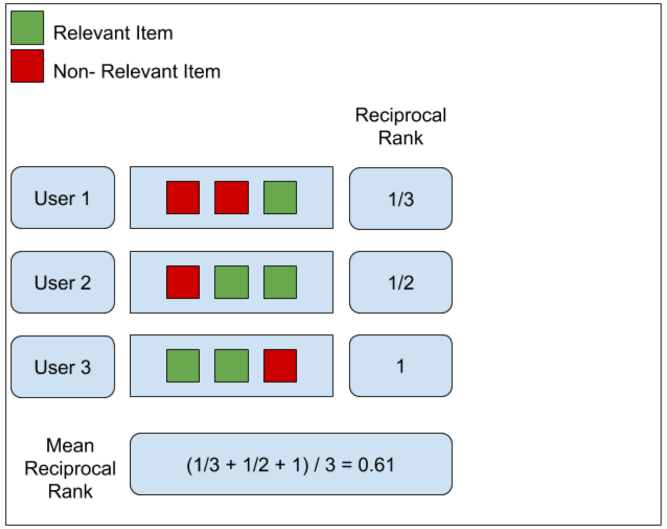
- b is a hyperparameter (usually 0.75) that determines how important the length of the document is
- |D|/avg이 refers to how long the document is compared to the average length of all the documents

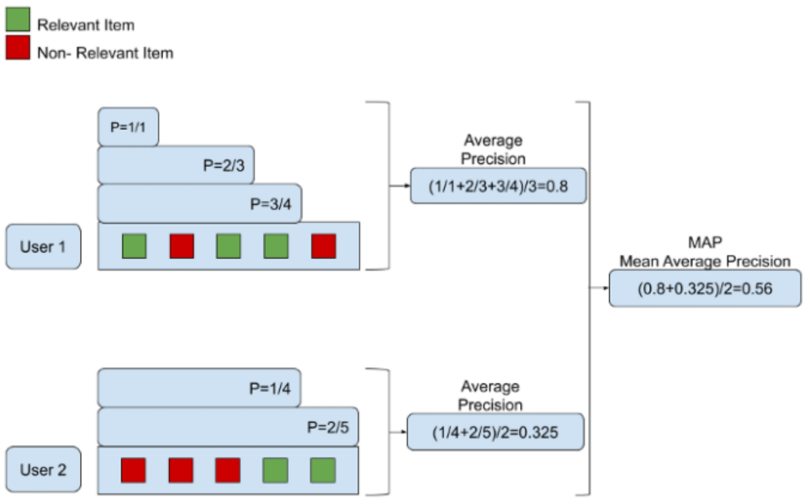


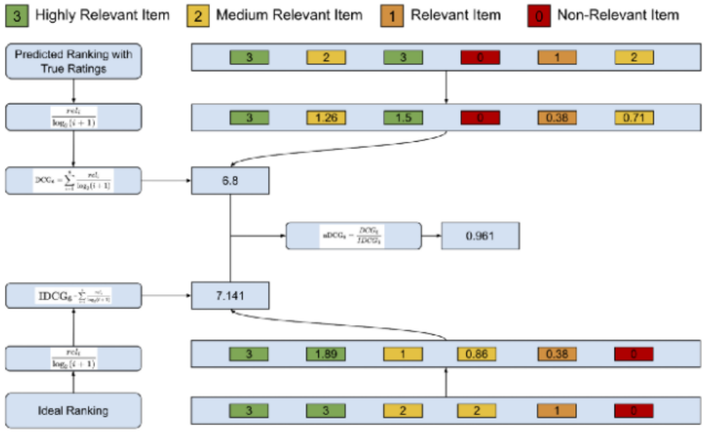
What is Information Retrieval?

- Datasets
  - Given a query, find the most relevant document
  - MS MARCO : A Human Generated Machine Reading Comprehension Dataset
  - Uses nDGC(Normalized Discounted Cumulative Gain), MRR(Mean Reciprocal Rank), MAP(Mean Average Precision) as metric

Approach	NDCG@10 (TREC DL 19 Reranking)	MRR@10 (MS Marco Dev)
BM25 (ElasticSearch)	45.46	17.29
msmarco-distilroberta-base-v2	65.65	28.55
msmarco-roberta-base-v2	67.18	29.17
msmarco-distilbert-base-v2	68.35	30.77





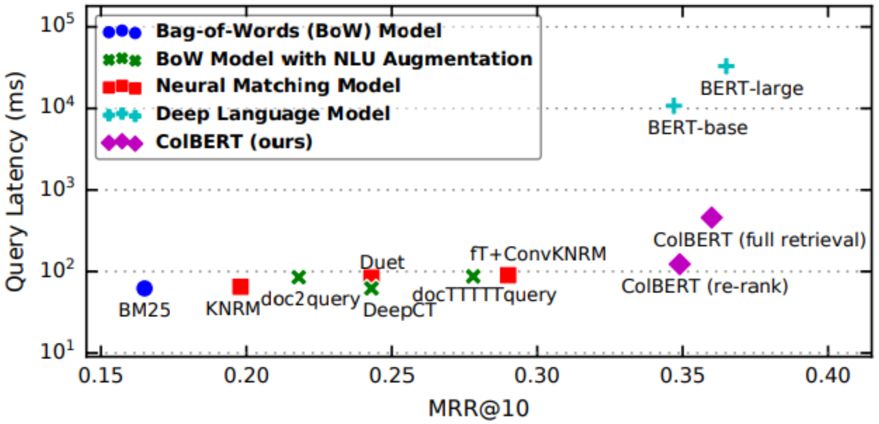


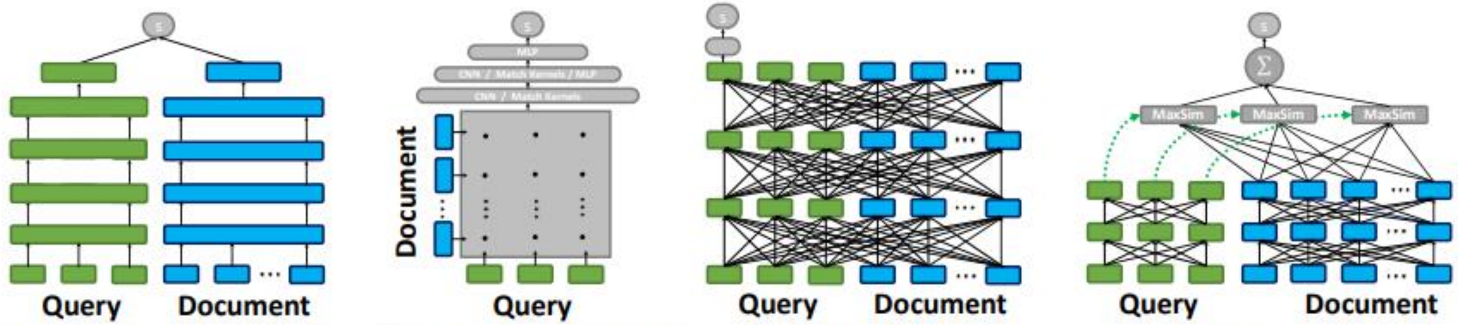
Multi-vector Representations

- ColBERT : Efficient and Effective Passage Search via Contextualized Late Interaction over BERT (SIGIR 2020)
    - Recent progress in NLU is driving fast-paced advances in IR, largely owned to fine-tuning LMs for document ranking.
    - While effective, the LM based ranking models increase computational cost, as they must feed each Query, Document pair to compute a single relevance score.
    - To tackle this problem, recent work started to explore using NLU techniques to augment traditional retrieval models like BM25.

    - ColBERT introduces a late interaction architecture that encodes Query and Document using BERT.
    - After encoding each component, ColBERT employs a cheap yet powerful interaction step that models their fine-grained similarity.
    - ColBERT can leverage the expressiveness of deep LMs by pre-computing document representations offline.

    - ColBERT's pruning-friendly interaction mechanism enables leveraging vector-similarity indexes for end-to-end retrieval directly from a large document collection.





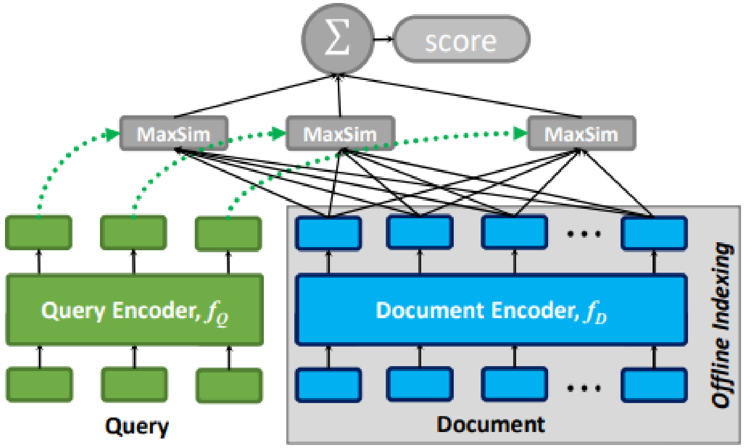
(a) Representation-based Similarity (e.g., DSSM, SNRM)
(b) Query-Document Interaction (e.g., DRMM, KNRM, Conv-KNRM)
(c) All-to-all Interaction (e.g., BERT)
(d) Late Interaction (i.e., the proposed ColBERT)

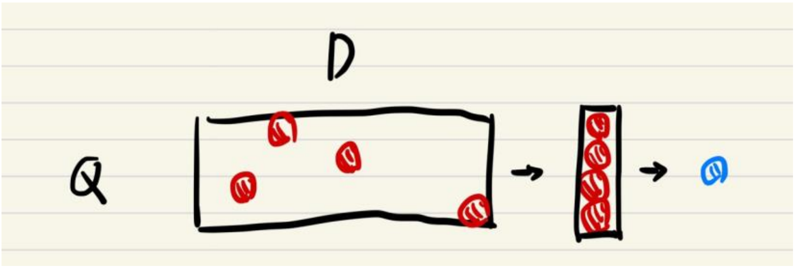
Multi-vector Representations

- ColBERT : Efficient and Effective Passage Search via Contextualized Late Interaction over BERT (SIGIR 2020)
  - Every query embedding interacts with all document embeddings via a MaxSim operator, which computes maximum similarity. (cossim)
  - Intuitively, interaction mechanism softly searches for each query in a manner that reflects its context in the query against the document's embeddings, quantifying the strength of the match via the largest similarity score between tq and td.
  - Given these term scores, it then estimates the document relevance by summing the matching evidence across all query terms.

  - Besides its cheap characteristic, it is also amenable to highly-efficient pruning for top-k retrieval.
  - Instead of applying MaxSim between every Q,D pairs, use fast vector-similarity data structures to conduct search (faiss from Facebook)

  - Other cheap choices(e.g. summation of average similarity scores instead of maximum) are possible, however many are less amenable to pruning.
  - Compared to existing BERT-based models, ColBERT delivers over 170x speedup.





$$E_q := \mathtt{Normalize}(\ \mathtt{CNN}(\ \mathtt{BERT}(``[Q]q_0q_1...q_l\#\#...\#")\ )\ )$$

$$E_d := \mathtt{Filter}(\ \mathtt{Normalize}(\ \mathtt{CNN}(\ \mathtt{BERT}(``[D]d_0d_1...d_n")\ )\ )\ )$$

$$S_{q,d} := \sum_{i\in[|E_q|]} \max_{j\in[|E_d|]} E_{q_i} \cdot E_{d_j}^T$$

Dense Retrieval Methods

- RepBERT : Contextualized Text Embeddings for First-Stage Retrieval (arXiv, Tsinghua Univ)
  - Similar to ColBERT that it encoded documents into fixed-length embeddings offline and save to disk to improve efficiency.
  - Selection of most relevant document can be formulated as Maximum Inner Product Search(MIPS).

  - Goal of training is to make the embedding inner product of relevant pairs of queries and documents larger than those of irrelevant pairs.
  - Adopt MultiLabelMarginLoss as loss function.
  - During training, it is expensive to sample negative documents; trick is to use other query-document pairs in same mini-batch.

Table 1: Performances of first-stage retrieval and two-stage retrieval models on MS MARCO Passage Ranking dataset

	MRR@10 Dev	MRR@10 Test	R@1000 Dev	Latency (ms/query)
BM25(Anserini) [2]	0.184	0.186	0.853	50
doc2query [2]	0.215	0.218	0.893	90
DeepCT [1]	0.243	0.239	0.913	55
docTTTTTquery [3]	0.277	0.272	**0.947**	64
Ours (RepBERT)	**0.304**	**0.294**	0.943	80
Best non-ensemble, non-BERT [19]	0.298	0.291	0.814	-
BM25 + BERT Large [20]	0.365	0.358	0.814	3,400

Table 2: Reranking accuracy (MRR@10) of BERT Large [20] using different first-stage retrieval techniques at different reranking depths. Dataset: MS MARCO dev. The improvement is relative to the reranking performance using BM25(Anserini) index.

Depths	BM25(Anserini)	doc2query	DeepCT	docTTTTTquery	RepBERT
5	0.232	0.265 (+14%)	0.279 (+20%)	0.314 (+36%)	**0.319 (+38%)**
10	0.276	0.307 (+11%)	0.320 (+16%)	**0.351 (+27%)**	0.344 (+25%)
50	0.336	0.354 (+5%)	0.361 (+8%)	**0.375 (+12%)**	0.370 (+10%)
500	0.366	0.373 (+2%)	0.374 (+2%)	**0.380 (+4%)**	0.377 (+3%)
1000	0.371	0.376 (+1%)	0.376 (+1%)	**0.380 (+2%)**	0.376 (+1%)

$$\mathrm{Input}(text) = [\mathrm{CLS}] \quad \mathrm{Tokenize}(text) \quad [\mathrm{SEP}]$$

$$\mathrm{Embed}(text) = \mathrm{Encoder}(text) = \mathrm{Average}(\mathrm{BERT}(\mathrm{Input}(text)))$$

$$\mathrm{Rel}(query, doc) = \mathrm{Embed}(query)^{\top} \cdot \mathrm{Embed}(doc)$$

$$\mathcal{L}(q, d_1^+, ..., d_m^+, d_{m+1}^-, ..., d_n^-) = \frac{1}{n} \cdot \sum_{1 \le i \le m, m < j \le n} \max(0, 1 - (\mathrm{Rel}(q, d_i^+) - \mathrm{Rel}(q, d_j^-)))$$

Dense Retrieval Methods

- Dense Passage Retrieval for Open-Domain Question Answering (EMNLP 2020)
  - Retrieval can be practically implemented using dense representation alone.
  - A term-based system would have difficulty retrieving a content where tokens with similar meaning but different words appears.
  - Bad guy == villain

  - Dense encodings are learnable by adjusting embedding functions.
  - Training a better dense embedding model only using pairs of questions and passages, without additional pretraining.

  - Simply fine-tuning the question and passage encoders on existing question-passage pairs is sufficient to greatly outperform BM25.
  - A higher retrieval precision indeed translates to a higher end-to-end QA accuracy.

  - More complex model frameworks do not necessarily provide additional values.

Training	Retriever	Top-20 NQ	Top-20 TriviaQA	Top-20 WQ	Top-20 TREC	Top-20 SQuAD	Top-100 NQ	Top-100 TriviaQA	Top-100 WQ	Top-100 TREC	Top-100 SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8	73.7	76.7	71.1	84.1	80.0
Single	DPR	78.4	79.4	73.2	79.8	63.2	85.4	**85.0**	81.4	89.1	77.2
	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**	83.8	84.5	80.5	92.7	**81.3**
Multi	DPR	**79.4**	78.8	**75.0**	**89.1**	51.6	**86.0**	84.7	**82.9**	93.9	67.6
	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2	83.9	84.4	82.3	**94.1**	78.6

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	$\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
Single	BM25	32.6	52.4	29.9	24.9	38.1
	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
	BM25+DPR	38.8	**57.9**	41.1	**50.6**	35.8

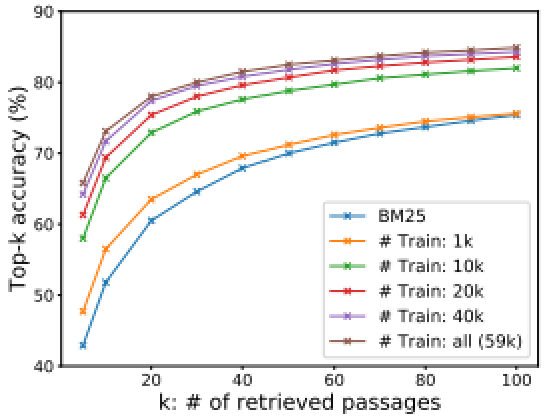
Dense Retrieval Methods

- Dense Passage Retrieval for Open-Domain Question Answering (EMNLP 2020)
  - Goal of DPR is to index all passages in a low-dimensional and continuous space such that it can retrieve efficiently the top k passages relevant to the input question for the reader at run time.

  - Dense encoder E_p maps any text passage to a d-dimensional real-valued vector.
  - Dense encoder E_q maps any input question to a d-dimensional real-valued vector at runtime.
  - $sim(q,p) = E_q(q)^T E_p(p)$ where we use CLS token.
  - Although more expressive model forms for measuring the similarity do exist, the similarity function needs to be decomposable so that the representations of the collection of passages can be precomputed.

  - Relevant pairs of questions and passages will have smaller distance(higher similarity)
  - While positive examples are available explicitly, select negative examples using 1)Random, 2)BM25, 3)Gold
  - Best model uses gold passages from the same mini-batch and one BM25 negative example.

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) \quad (2)$$
$$= -\log \frac{e^{\mathrm{sim}(q_i, p_i^+)}}{e^{\mathrm{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\mathrm{sim}(q_i, p_{i,j}^-)}}.$$



Dense Retrieval Methods

- Dense Passage Retrieval for Open-Domain Question Answering (EMNLP 2020)

Question	**Passage received by BM25**	**Passage retrieved by DPR**
What is the body of water between England and Ireland?	Title:British Cycling . . . **England** is not recognised as a region by the UCI, and there is no English cycling team outside the Commonwealth Games. For those occasions, British Cycling selects and supports the **England** team. Cycling is represented on the Isle of Man by the Isle of Man Cycling Association. Cycling in Northern **Ireland** is organised under Cycling Ulster, part of the all-Ireland governing **body** Cycling **Ireland**. Until 2006, a rival governing **body** existed, . . .	Title: Irish Sea . . . Annual traffic between Great Britain and **Ireland** amounts to over 12 million passengers and of traded goods. **The Irish Sea** is connected to the North Atlantic at both its northern and southern ends. To the north, the connection is through the North Channel between Scotland and Northern **Ireland** and the Malin Sea. The southern end is linked to the Atlantic through the St George's Channel between Ireland and Pembrokeshire, and the Celtic Sea. . . .
Who plays Thoros of Myr in Game of Thrones?	Title: No One (Game of Thrones) . . . He may be "no one," but there's still enough of a person left in him to respect, and admire who this girl is and what she's become. Arya finally tells us something that we've kind of known all along, that she's not no one, she's Arya Stark of Winterfell." "No One" saw the reintroduction of Richard Dormer and **Paul Kaye**, who portrayed Beric Dondarrion and **Thoros** of **Myr**, respectively, in the third season, . . .	Title: Pål Sverre Hagen Pål Sverre Valheim Hagen (born 6 November 1980) is a Norwegian stage and screen actor. He appeared in the Norwegian film "Max Manus" and played Thor Heyerdahl in the Oscar-nominated 2012 film "Kon-Tiki". Pl Hagen was born in Stavanger, Norway, the son of Roar Hagen, a Norwegian cartoonist who has long been associated with Norwayś largest daily, "VG". He lived in Jtten, a neighborhood in the city of Stavanger in south-western Norway. . . .

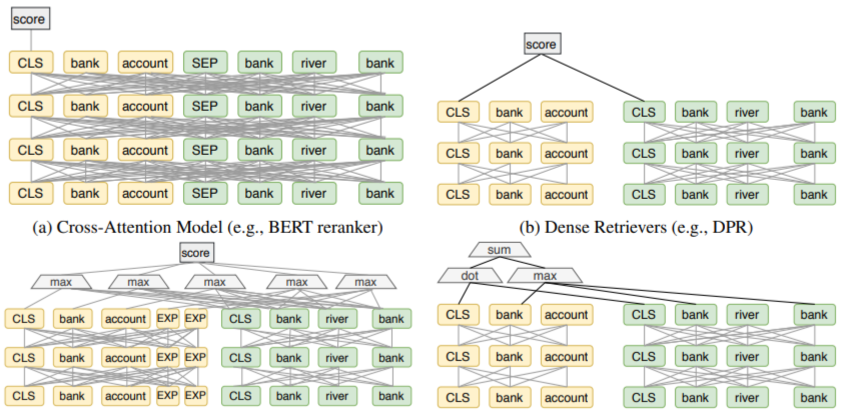
Dense Retrieval Methods

- COIL : Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List (NAACL 2021)
  - How much gain can we get if we introduce contextualized representations back to lexical exact match systems?
  - In other words, can we build a system that still performs exact query-document token matching but compute matching signals with contextualized token representations instead of heuristics?
  - This may seem a constraint on the model, but exact lexical match produce more explainable and controlled patterns than soft matching.

  - COIL scoring is based on overlapping query document tokens' contextualized representations.
  - The new architecture stores contextualized token representations in inverted lists, bringing together the efficiency of EM and representation power of deep language models.

  - COIL brings semantic matching into lexical IR systems.
  - COIL shows matching signals induced from exact lexical match can capture complicated matching patterns.

	MS MARCO Document Ranking				
	Dev Rerank	Dev Retrieval		DL2019 Retrieval	
Model	MRR@10	MRR@10	Recall@1K	NDCG@10	MRR@1K
Lexical Retriever					
BM25	–	0.230	0.886	0.519	0.805
DeepCT	–	0.320	0.942	0.544	0.891
DocT5Query	–	0.288	0.926	0.597	0.837
BM25+BERT reranker	0.383	–	–	–	–
Dense Retriever					
Dense (BM25 neg)	n.a.	0.299	0.928	0.600	n.a.
Dense (rand + BM25 neg)	n.a.	0.311	0.952	0.576	n.a.
Dense (our train)	0.358	0.340	0.883	0.546	0.785
COIL-tok	0.381	0.385	0.952	0.626	0.921
COIL-full	0.388	0.397	0.962	0.636	0.913



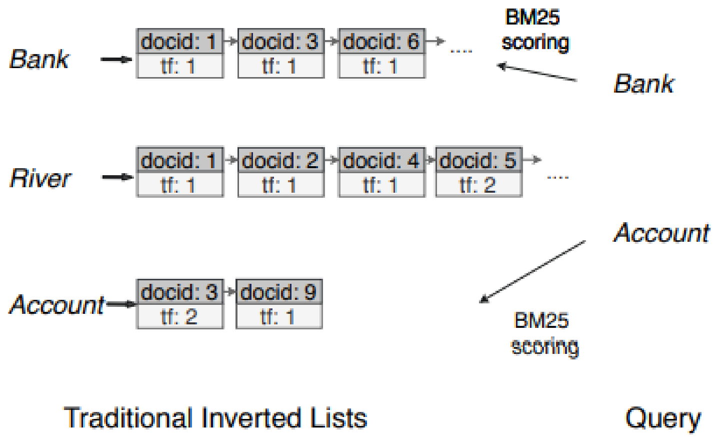
(a) Cross-Attention Model (e.g., BERT reranker)
(b) Dense Retrievers (e.g., DPR)
(c) ColBERT: All-to-All Match
(d) COIL: Contextualized Exact Match

Dense Retrieval Methods

- COIL : Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List (NAACL 2021)
  - Inverted List maps back from a term to a list of documents where the term occurs.
  - The inverted list maps back from a term to a list of documents where the term occurs instead of entire document collection.

  - Define contextualized exact lexical match scoring function between query document based on vector similarities between exact matched query document token pairs.
  - In order to overcome vocabulary mismatch problem, use extra CLS token from query and document.

  - Like DPR, use batch negatives and hard negatives generated by BM25 to use a negative log likelihood loss function.

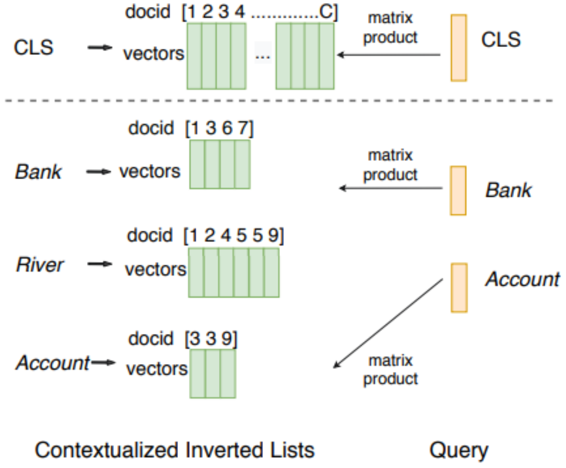


$$s = \sum_{t \in q \cap d} \sigma_t(h_q(q,t), h_d(d,t))$$

$$s_{\text{BM25}} = \sum_{t \in q \cap d} idf(t) h_q^{\text{BM25}}(q,t) h_d^{\text{BM25}}(d,t)$$

$$h_q^{\text{BM25}}(q,t) = \frac{tf_{t,q}(1+k_2)}{tf_{t,q}+k_2}$$

$$h_d^{\text{BM25}}(d,t) = \frac{tf_{t,d}(1+k_1)}{tf_{t,d}+k_1(1-b+b\frac{|d|}{\text{avgdl}})}$$



$$\boldsymbol{v}_i^q = \boldsymbol{W}_{tok}\text{LM}(q,i) + \boldsymbol{b}_{tok}$$

$$\boldsymbol{v}_j^d = \boldsymbol{W}_{tok}\text{LM}(d,j) + \boldsymbol{b}_{tok}$$

$$s_{\text{tok}}(q,d) = \sum_{q_i \in q \cap d} \max_{d_j = q_i}(\boldsymbol{v}_i^{q\intercal} \boldsymbol{v}_j^d)$$

$$\boldsymbol{v}_{cls}^q = \boldsymbol{W}_{cls}\text{LM}(q,\text{CLS}) + \boldsymbol{b}_{cls}$$

$$\boldsymbol{v}_{cls}^d = \boldsymbol{W}_{cls}\text{LM}(d,\text{CLS}) + \boldsymbol{b}_{cls}$$

$$s_{\text{full}}(q,d) = s_{\text{tok}}(q,d) + \boldsymbol{v}_{cls}^{q\ \intercal} \boldsymbol{v}_{cls}^d$$

$$\mathcal{L} = -\log \frac{\exp(s(q,d^+))}{\exp(s(q,d^+)) + \sum_l \exp(s(q,d_l^-))}$$

Dense Retrieval Methods

- COIL : Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List (NAACL 2021)

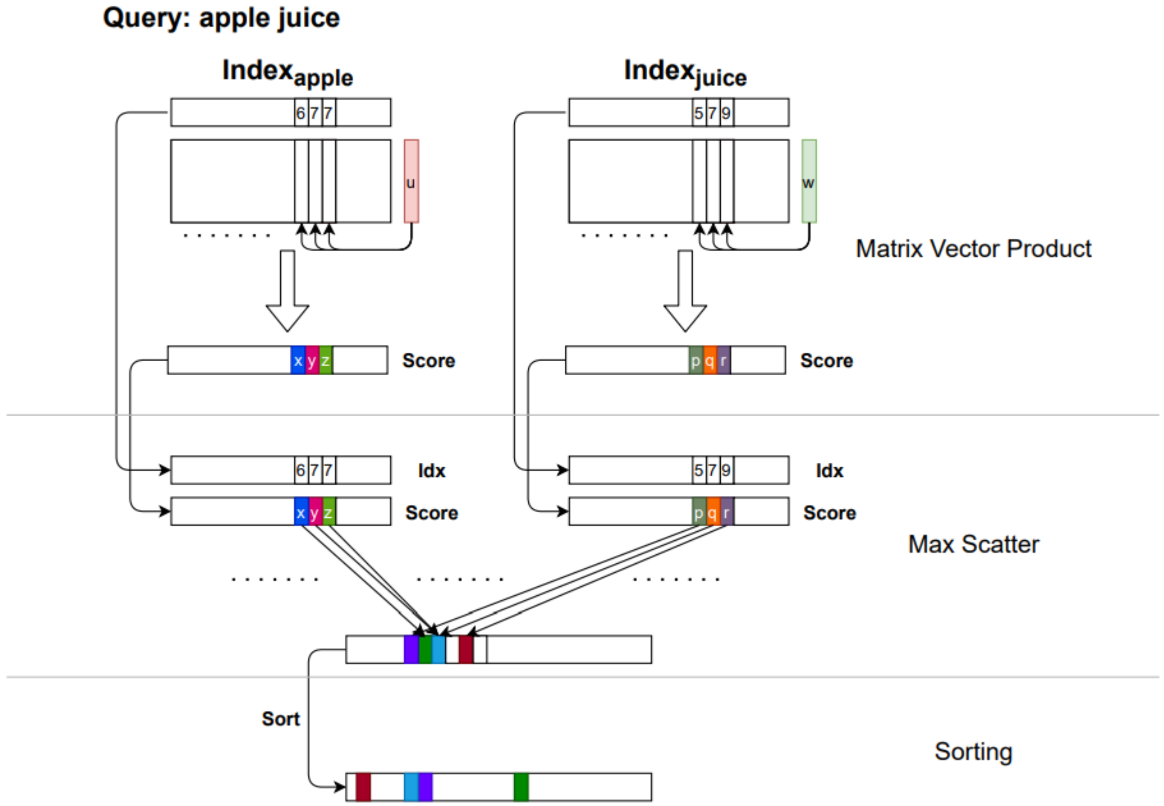


Figure 5: COIL Search of query "apple juice".

ANY QUESTIONS?