Static Pruning for Multi-Representation Dense Retrieval

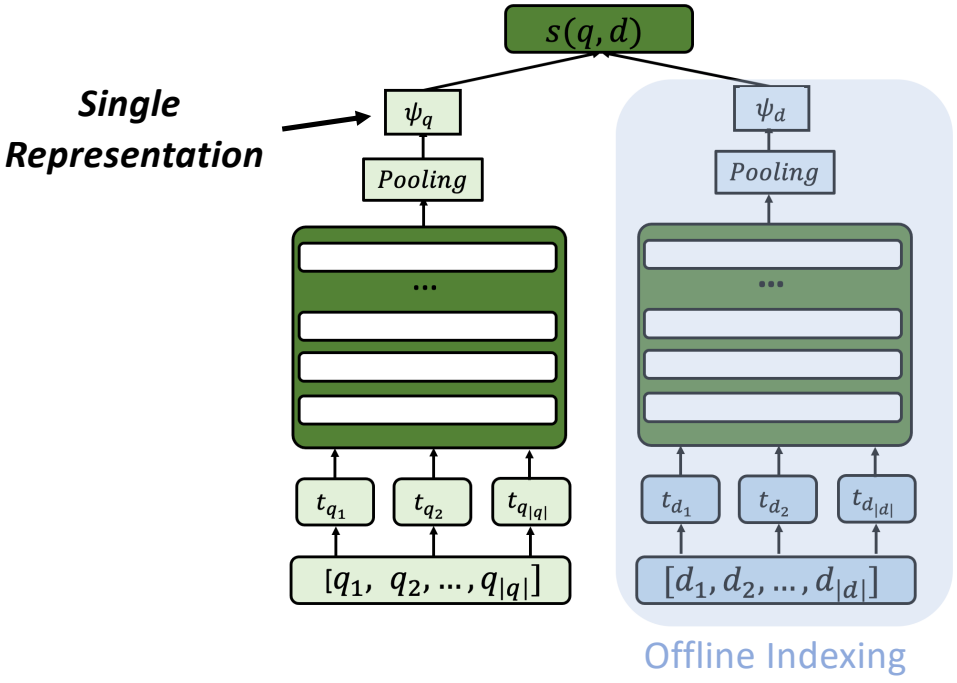
Antonio Acquavia[12], **Craig Macdonald[1]**, Nicola Tonellotto[2]

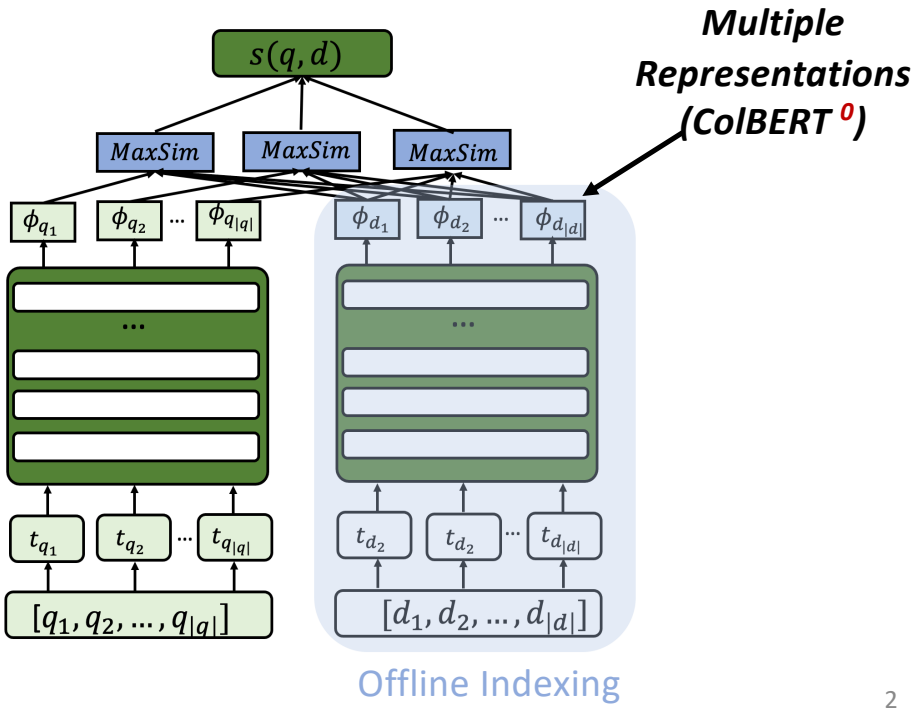
[1]University of Glasgow; [2]University of Pisa

ACM DocEng 2023, Limerick, Ireland

Dense Retrieval

Search using **vector-based representations** (obtained from pre-trained language models) are becoming more popular due to their ability to conduct semantic search – more than just **exact term matches**

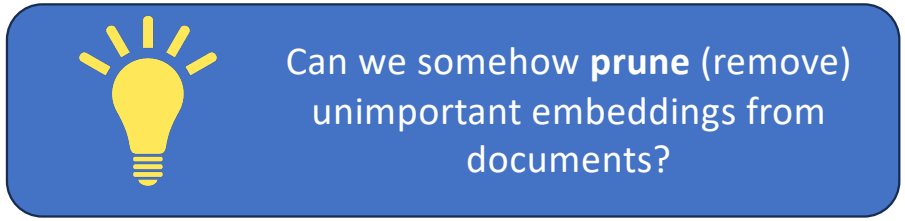


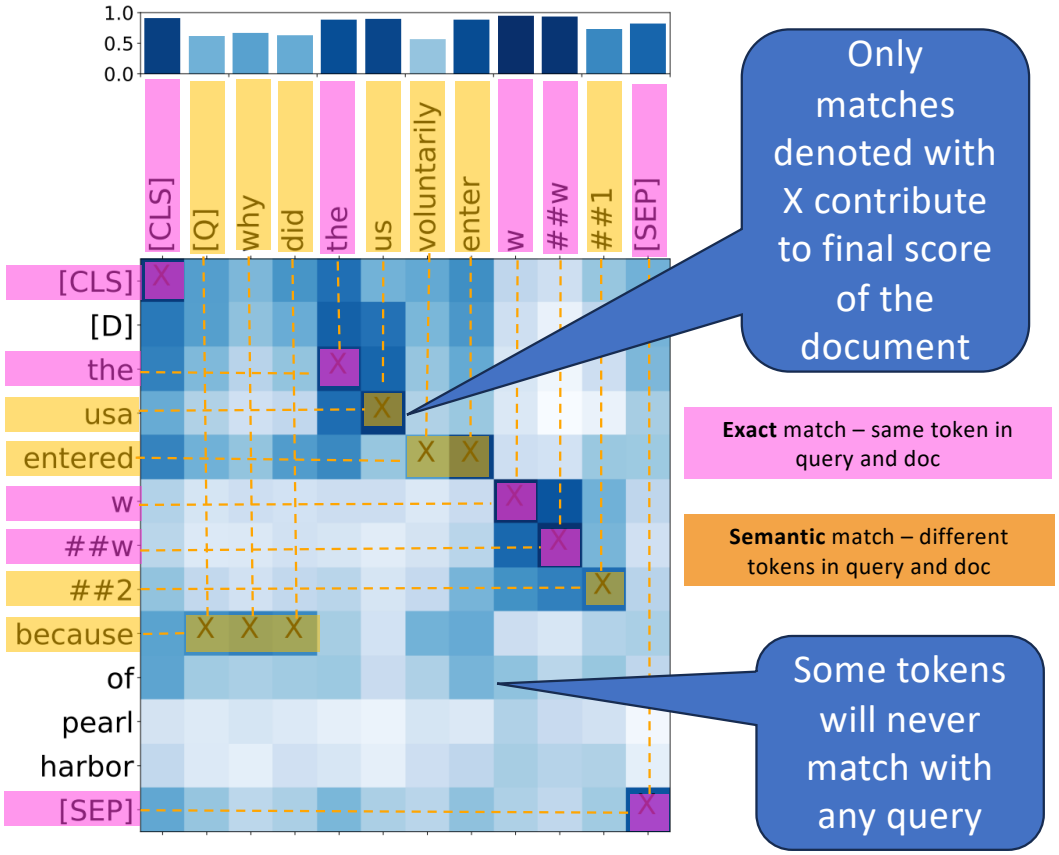


ColBERT Multiple Representation Dense Retrieval

ColBERT is attractive due to:

- Use of token-level embedded representations. These allows more extensions (e.g. query expansion – ColBERT-PRF [1,2])
- "Zero shot" effectiveness
- Extensible to other PLMs such as Roberta[3]
- Explainable nature of its scoring[1]





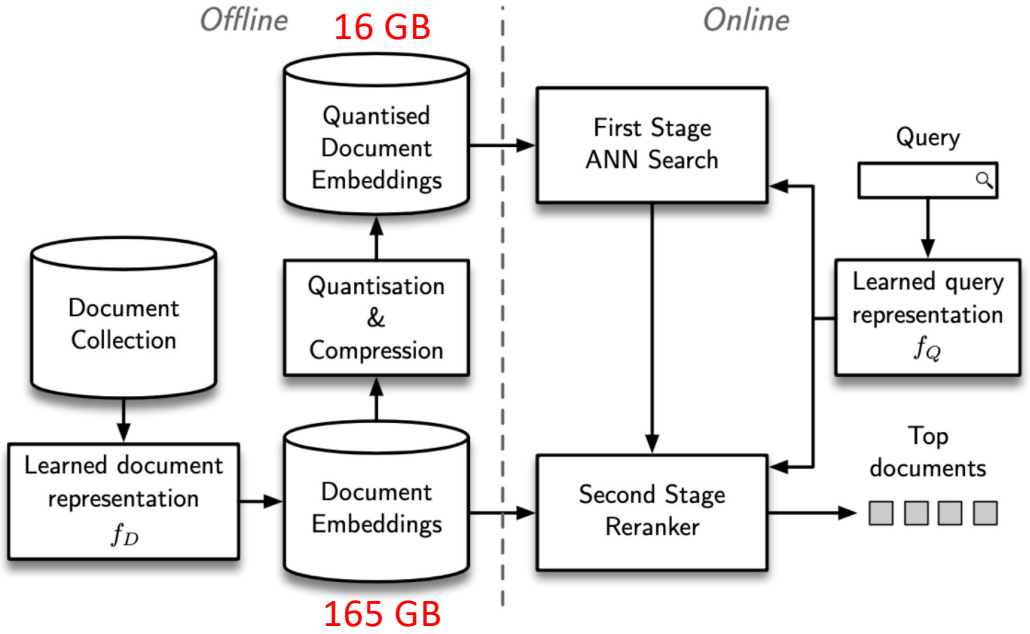
Top: The **histogram** depicts the magnitude of contribution for each query embedding to the final score of the document.
Bottom: **Contextualised interaction matrix** between a query and a document text.

Scale of the Problem…

ColBERT uses a two-stage ranking architecture

(i) an approximate nearest neighbour search, to identify a candidate set

(ii) Exact re-scoring using document embeddings



Index Sizes – MSMARCO passage corpus (8.8M passages)	Total
Sparse: Inverted Index (e.g. BM25)	1.4GB
Dense: ANCE (single representation	25.4GB
Dense: ColBERT (multiple representation)	185GB

185GB must be kept in memory for fast retrieval!

Background: Static Pruning of Sparse Indices

Static pruning approaches aim to identify (occurrences of) words that **do not contribute** to the ordering of the top ranked documents.

Uniform

Completely removing terms or documents from the index
e.g. manually curated stopwords, or posting lists for low IDF terms

Term-centric

Pruning decision is taken on a **posting's rank wrt. the other postings** in the corresponding term's posting list
i.e. *"is this document a good document for this term?"*

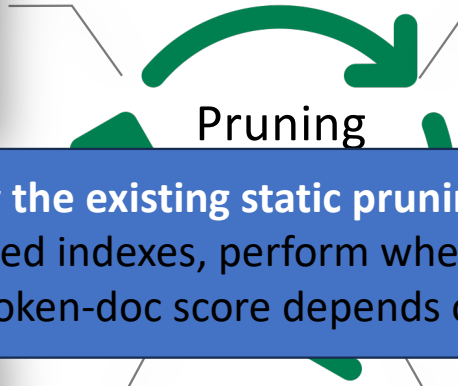
Document-centric

Pruning decision is taken on **a posting's rank within the document** it refers to
i.e. *"is this term important in this document?"*

Related Work: Pruning for Dense Retrieval

A straightforward approach is to **reduce the dimension** of the embeddings, as done by ColBERT and miniLM[4].

Khattab et al.[5] proposed to to **quantise** the embeddings index, resulting in reduced space usage.



Unclear how the existing static pruning strategies, proposed for sparse inverted indexes, perform when **adapted to dense embedding** indexes (as token-doc score depends on embeddings)

Tonellotto and Macdonald[6] **pruned query embeddings** for effective retrieval while reducing the number of documents requiring to be exactly scored.

Lassance et al.[7] studied the impact of token pruning in ColBERT **during model training**, obtaining a space reduction of 30% at the cost of a new neural model training phase. Instead, we aim to alter a normal ColBERT index.

Proposal: Static Pruning Methods for ColBERT

uniform pruning: remove all embeddings in any document corresponding to the tokens ranked by their **global importance**, up to a threshold number of globally removed tokens

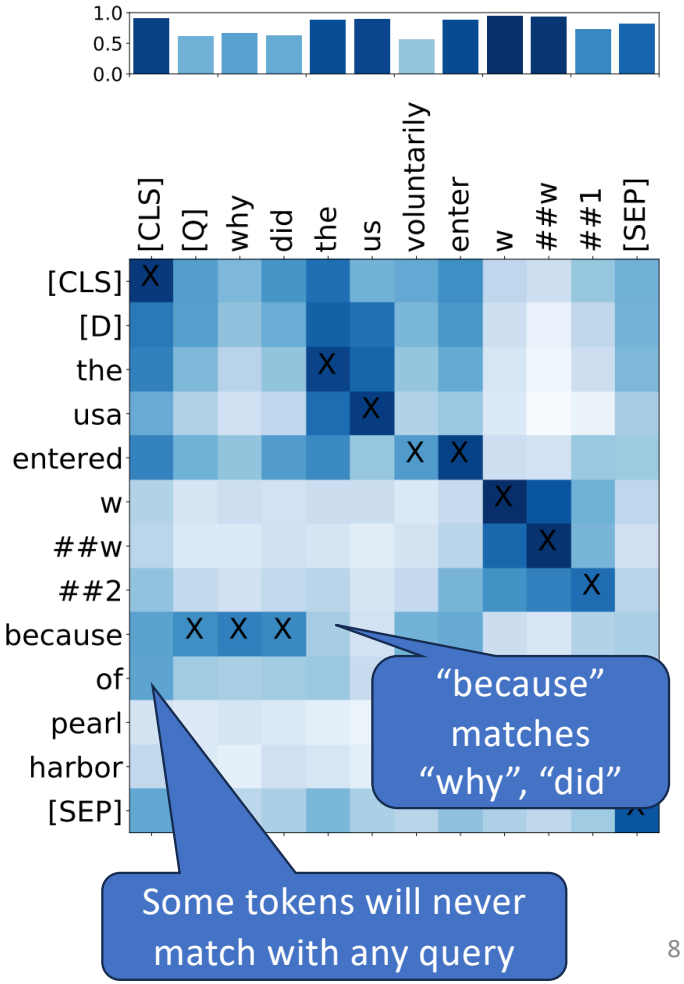
document-centric pruning: remove a few embeddi… each document corresponding to the tokens ranked by their **global importance**, up to a threshold number of tokens removed per document

How to define the important and unimportant embeddings/tokens?

- Embeddings are pruned (i.e. after encoding), rather than tokens (before encoding)
- This preserves the contextualised nature of the embeddings

Computing Embedding Importance

- In previous pruning work, query logs were often used to measure token importance – tokens not used in queries did not need to appear in documents
- In contrast, in ColBERT, due to *semantic matching*, we argue that we cannot prune tokens/embeddings from documents just based on their occurrence in historical queries
- We hypothesise that tokens that are frequent in the corpus (i.e. low IDF) have embeddings that aren't useful to match (e.g. *of/in/the*) etc.
  - *Uniform*: We remove embeddings for the τ_u tokens with lowest IDF weights from all documents
  - *Document-Centric*: We remove embeddings for the τ_d tokens with lowest IDF weights in each document



Research Questions

- **RQ1:** What is the impact, in terms of effectiveness and space reduction, of **document-centric pruning** in dense retrieval, where we remove in each document a given number of embeddings, corresponding to low IDF terms?

- **RQ2:** What is the impact, in terms of effectiveness and space reduction, of **uniform pruning in dense retrieval,** where we remove globally a given number of embeddings, corresponding to low IDF terms?

- **RQ3:** What is the impact, in term of effectiveness, of learning a new **quantised document embeddings index** for first stage ANN search after static pruning?

Experimental Setup

Dataset: MSMARCO passage ranking corpus (8.8M passages)

Code Implementation: PyTerrier_ColBERT

Baselines: CoLBERT, and ColBERT implementation using miniLM[4] – this has lower dimensional embeddings

Evaluation: TREC 2019 and TREC 2020 query sets
Measures: nDCG@10, MAP etc.

Pruning Implementations:

- Original
- Stopwords (Uniform)
- Random doc-centric
- IDF doc-centric

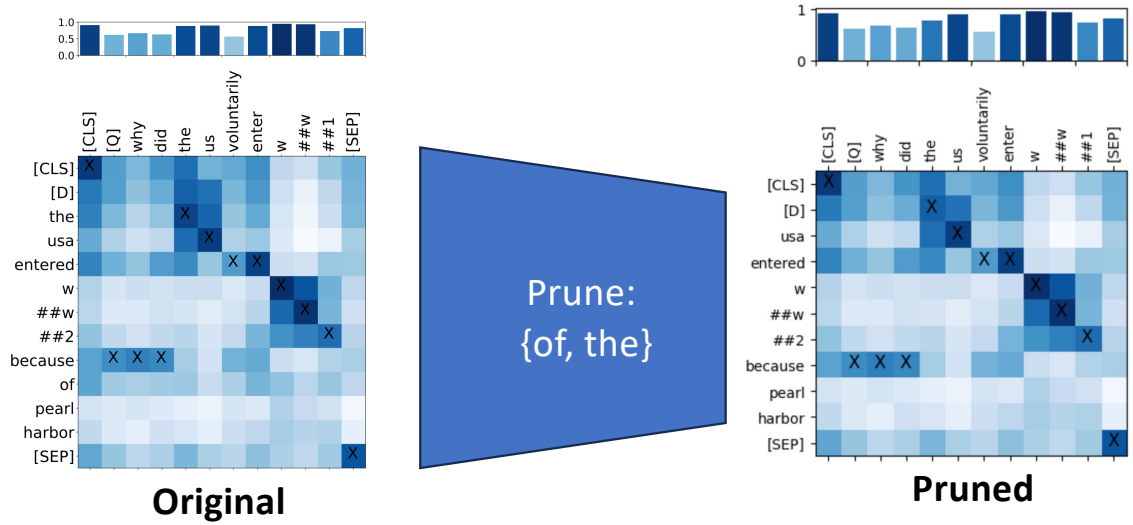
$$\tau_d = \{1, 3, 5, 8, 10, 15, 20, 25, 30\}$$

- IDF uniform
- MSMARCO uniform
- MSN uniform

$$\tau_u = \{1, 10, 25, 50, 100, 250, 500, 750, 1000, 2000, 3000\}$$

Pruning Evaluation (1)

- Success is a small index that is still effective at search..
- Its costly to reindex every time to test pruning evaluations
- Instead, we propose to evaluate using the Document Length averaged over the top-retrieved documents



84% embeddings of Original

- We report the Average Document Length@100 relative to the Original (unpruned) index

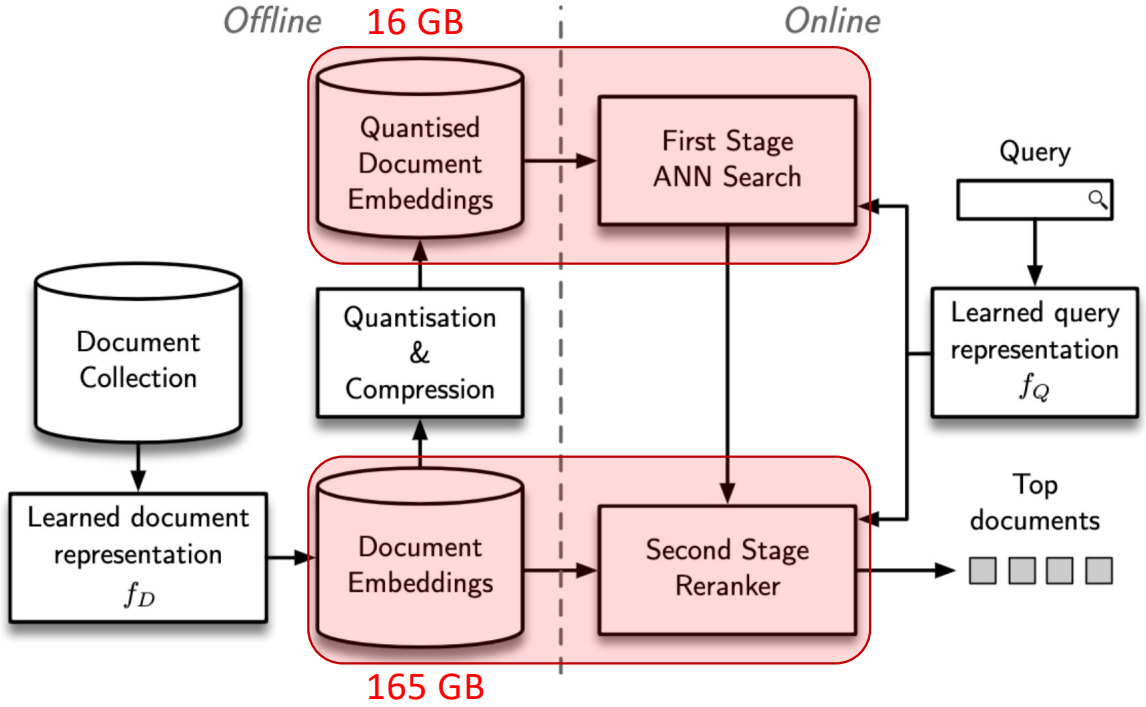
Pruning Evaluation (2)

In RQ1 & RQ2, we use Average Document Length to measure pruning

- This is pruning of the (large) document embeddings index structure only
- Recall of Candidate Set is unaffected

For RQ3, we test pruning **both** ANN search and document embeddings index structures, measuring actual index size

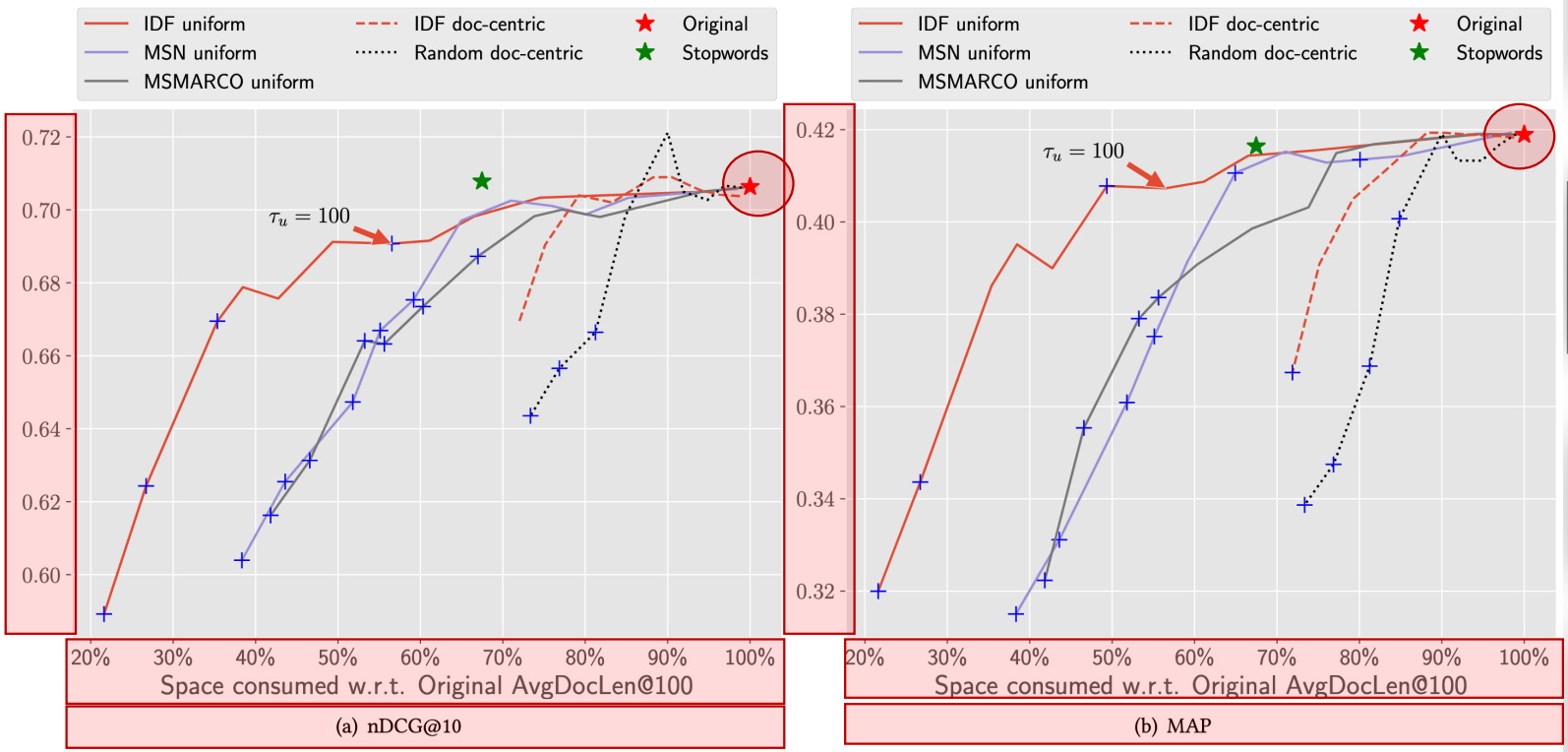
- Possible impact on Recall!



Reranking Evaluation

RQ1 & RQ2

Results for TREC 2019 (RQs1&2)



Random pruning quickly causes significant effectiveness drop as pruning level increases

Similarly, IDF doc-centric pruning is also aggressive. Significance observed at 28% reduction in AvgDocLen

Stopwords removal is competitive – above all Uniform approaches

Among Uniform methods, IDF is best; query log start pruning "good" tokens beyond 35% pruning

In general, for Uniform IDF pruning, τ_u = 100 is a reasonable setting – usually insignificant effectiveness degradations. We'll study this point further for RQ3, comparing with Stopwords…

Full Index Evaluation

RQ3

Full Index Pruning (RQ3)

Setting	Index Size			TREC 2019			TREC 2020		
	Total	ANN	Embeddings	nDCG@10	MAP	R@1000	nDCG@10	MAP	R@1000
Original	185 GB	16 GB	165 GB	0.706	0.419	0.671	0.687	0.459	0.733
miniLM	30%	100%	25%	0.632*	0.368*	0.625*	0.664	0.423*	0.695
Stopwords (orig. ANN)	70%	100%	67%	0.708	**0.416**	0.671	**0.678**	0.452	**0.733**
Stopwords (new ANN)	67%	69%	67%	**0.709**	**0.416**	**0.681**	0.674	0.449	0.713*
IDF uniform $\tau = 100$ (orig. ANN)	60%	100%	55%	0.691*	0.407	0.680	0.674	**0.455**	**0.733**
IDF uniform $\tau = 100$ (new ANN)	55%	54%	55%	0.690*	0.398*	0.678	0.667	0.442	0.709

Index sizes are markedly reduced - both for ANN and for Embeddings

Keeping the original ANN index tends to exhibit slightly higher effectiveness (exception: Stopwords on TREC 2019)

Despite reducing index sizes by 45%, τ_u=100 exhibits mostly comparable effectiveness to the original unpruned system

Using the IDF Uniform pruned index is always more effective than miniLM

Verification on TREC-Covid

- 171k paper abstracts, 50 topics
- IDF computations using MSMARCO

Setting	Index Size			Effectiveness	
	Total	ANN	Embeddings	nDCG@10	MAP
Original	6.5GB	5.5GB	526MB	0.680	0.154
Stopwords (orig. ANN)	72%	100%	72%	0.685	0.151
Stopwords (new ANN)	69%	73%	72%	**0.689**	0.143
IDF uniform $\tau = 100$ (orig. ANN)	65%	100%	65%	0.671	**0.154**
IDF uniform $\tau = 100$ (new ANN)	62%	64%	65%	0.672	0.147

IDF uniform pruning reduces index size more than Stopwords, but at the cost of a little more effectiveness (not significant)

Replacing the unpruned ANN index with a pruned index results in slight drops in MAP, but nDCG@10 remains stable

- Overall, IDF uniform pruning (with $\tau = 100$) is again appropriate for reducing index sizes (achieving a 38% reduction of the index), statistically indistinguishable effectiveness to an unpruned index (e.g. 1.3% reduction in nDCG@10: 0.680→0.672), and better than Stopwords

Conclusions

- We demonstrated how classical static pruning approaches could be generalised to consider embedding-based dense retrieval indexes
- We proposed both document-centric and uniform pruning methods, based on the IDF of the corresponding BERT tokens
- By removing the embeddings associated with the terms with the lowest IDF tokens, we can markedly reduce the index size, while minimising degradation in effectiveness
- Our experiments on the MSMARCO v1 passage showed that by removing embeddings corresponding to the 100 most frequent BERT tokens, the total **index size is reduced by 45%,** and effectiveness is only **marginally reduced** (up to 4% reduction)
- For our experiments on TREC Covid, we observed a **statistically indistinguishable** 1.3% reduction in nDCG@10 for a **38% reduction in total index size**

- Future work: We have observed supervised models have shown that learned token importance is more important than IDF, e.g. DeepImpact[8], CWPRF[2]. For instance, 'us' can refer to USA or a stopword pronoun. We may be able to do better with **supervised** token importance models

References

0. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. Omar Khattab and Matei Zahara. In Proc. SIGIR 2020.
1. ColBERT-PRF: Semantic Pseudo-Relevance Feedback for Dense Passage and Document Retrieval. Xiao Wang, Craig Macdonald, Nicola Tonellotto, Iadh Ounis. In ACM Transactions on the Web 2023.
2. Effective Contrastive Weighting for Dense Query Expansion. Xiao Wang, Sean MacAvaney, Craig Macdonald, Iadh Ounis. In Proc. ACL 2023.
3. Reproducibility, Replicability, and Insights into Dense Multi-Representation Retrieval Models: from ColBERT to Col*. Xiao Wang, Craig Macdonald, Nicola Tonellotto, Iadh Ounis. In Proc. SIGIR 2023.
4. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. In Proc. NeurIPS 2020.
5. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. In Proc. NAACL 2022.
6. Query Embedding Pruning for Dense Retrieval. Nicola Tonellotto and Craig Macdonald. In Proc. CIKM 2021.
7. Learned Token Pruning in Contextualized Late Interaction over BERT. Carlos Lassance, Maroua Maachou, Joohee Park, and Stéphane Clinchant. In Proc. SIGIR 2022.
8. Learning Passage Impacts for Inverted Indexes. Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. In Proc. SIGIR 2021.

University of Glasgow

Università di Pisa

Questions?

Code:

https://github.com/cmacdonald/colbert_static_pruning