Lexically-Accelerated Dense Retrieval

Hrishikesh Kulkarni[1], Sean MacAvaney[2], Nazli Goharian[1], Ophir Frieder[1]

[1]Georgetown University
[2]University of Glasgow

ACM SIGIR 2023

Problem

This is where we want to be

1. Lexical retrieval:

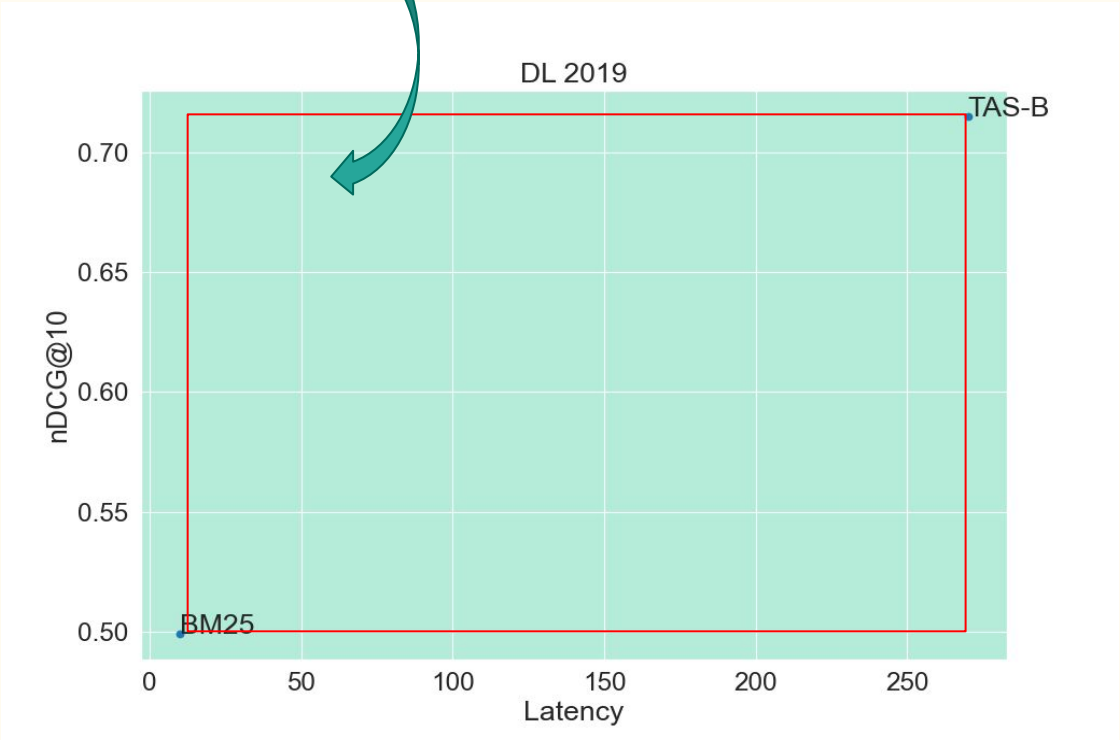
   Inverted index - high efficiency

   Vocabulary based - low effectiveness

2. Dense retrieval:

   Exhaustive - low efficiency

   Semantic - high effectiveness

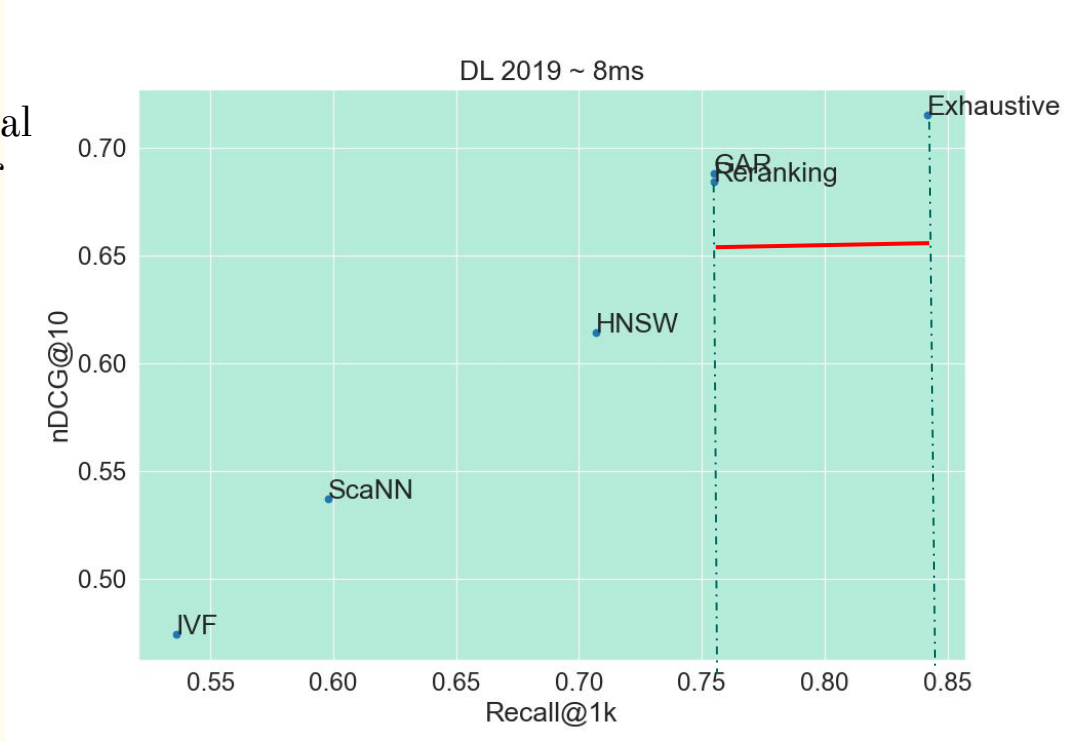


Existing Solutions

1. Approximation methods: Methods which aim to reduce the computational overhead by approximating results of a full dense retriever e.g. IVF, ScaNN, HNSW, GAR etc

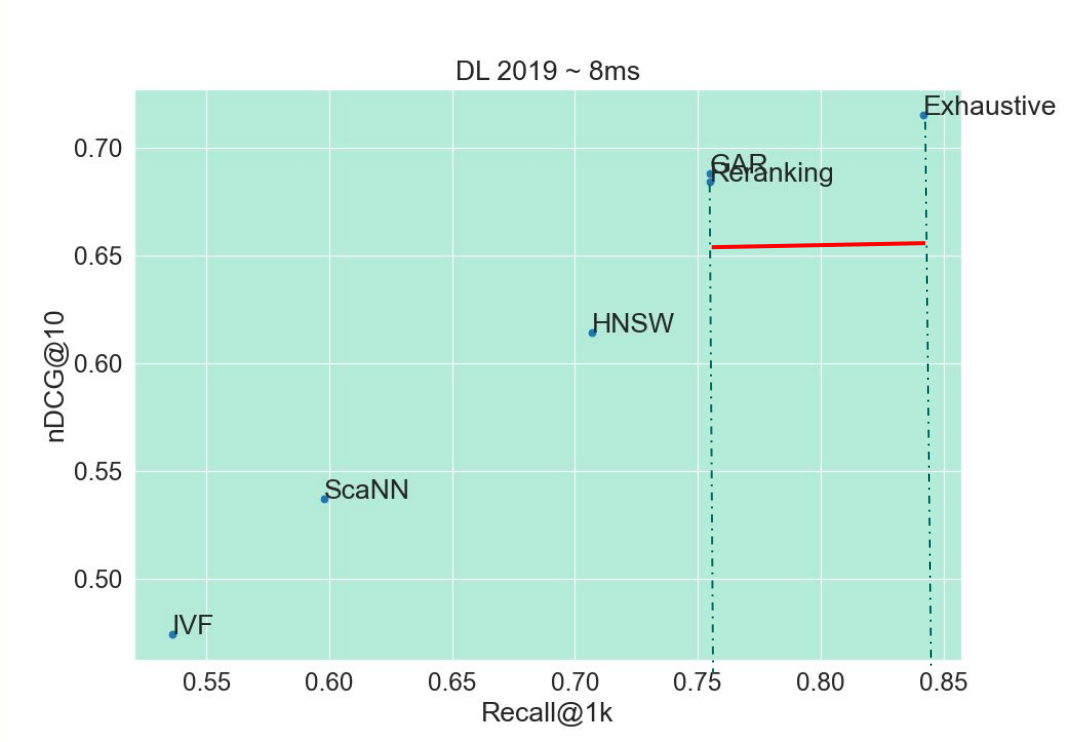
   Good at approximating top results - nDCG@10

   Suffer in terms of Recall - Recall@1000



Where do the Approximation Methods Lack?

1. Seed points influenced by graph building algorithms and don't take query characteristics into account
2. Random exploration of search space is expensive
3. Approaches like reranking only explore documents having lexical overlap with the query
4. Termination at local minimas



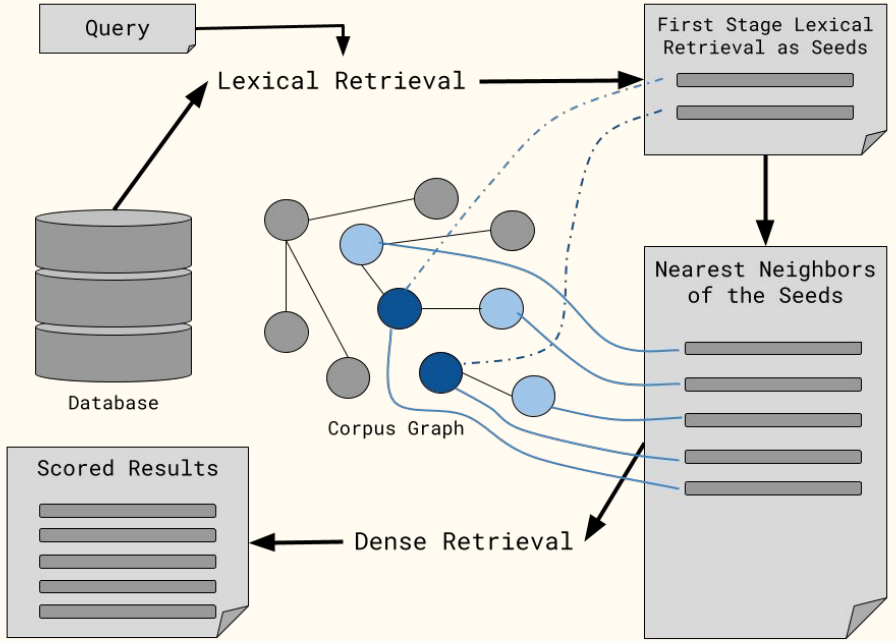
How we address this gap?

1. LADR: improves efficiency without compromising on retrieval effectiveness

2. Informed exploration by identifying right seed points

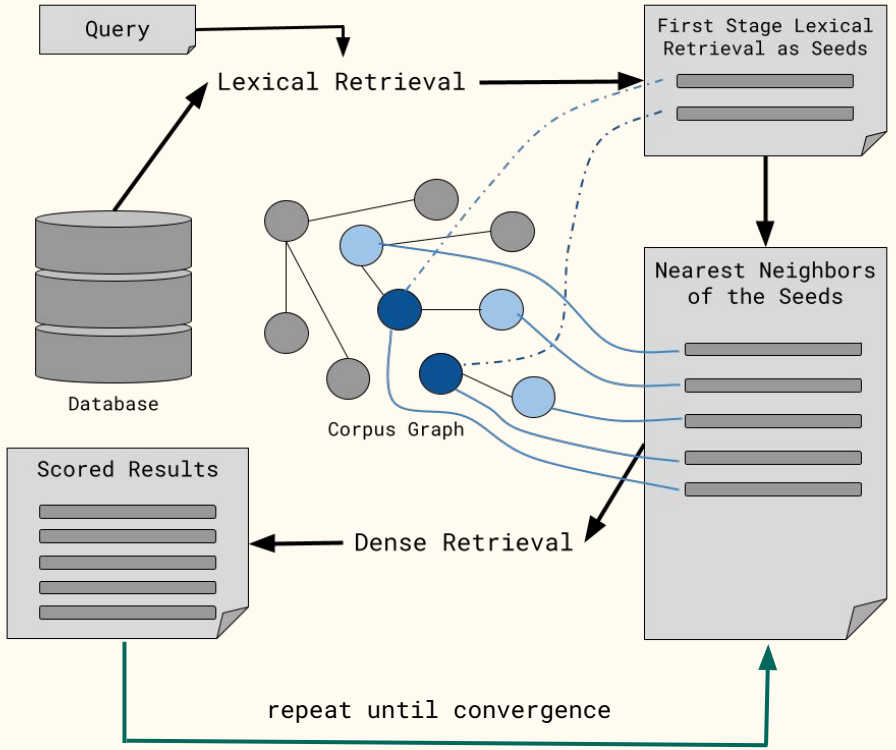
3. Exploration: Pre-computed document proximity graph

   Seed Points: Results from Efficient lexical model

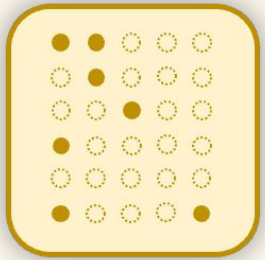


Alternative: Adaptive LADR

Iteratively explores neighbors of top c results until they converge

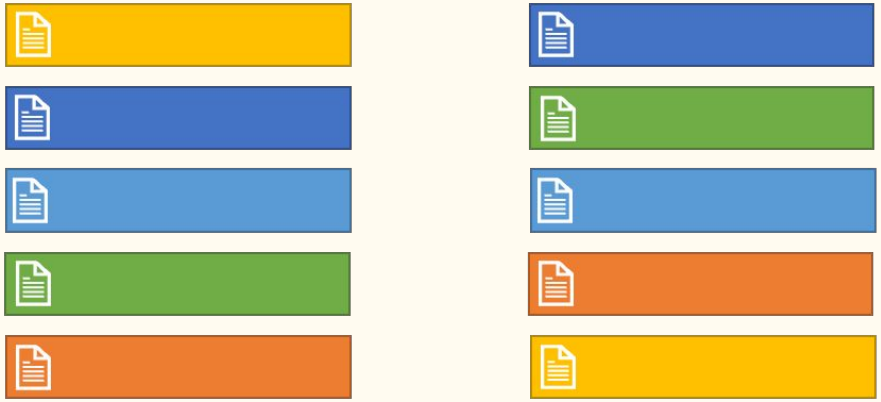


Alternative: Adaptive LADR



BM25

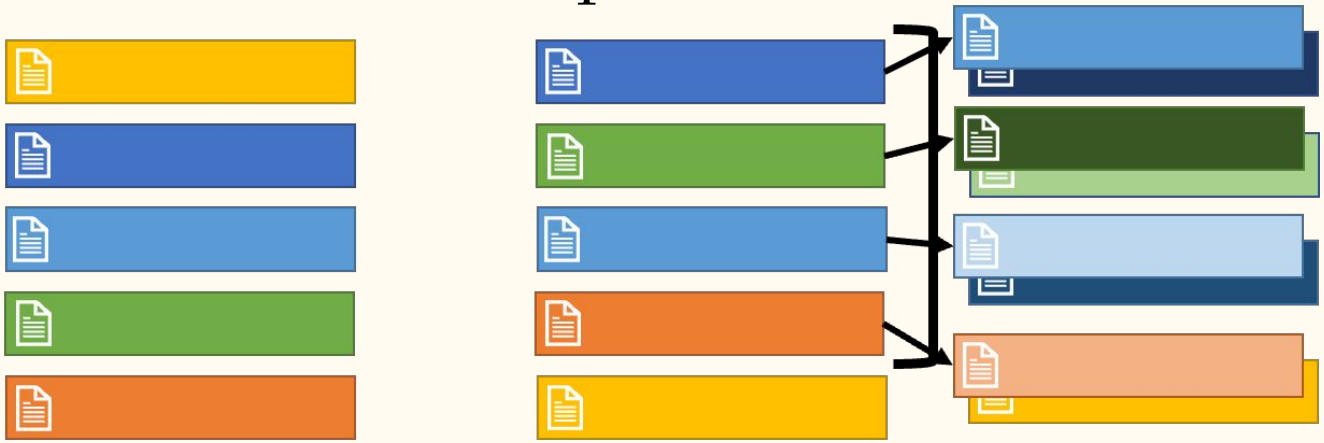
Alternative: Adaptive LADR



BM25

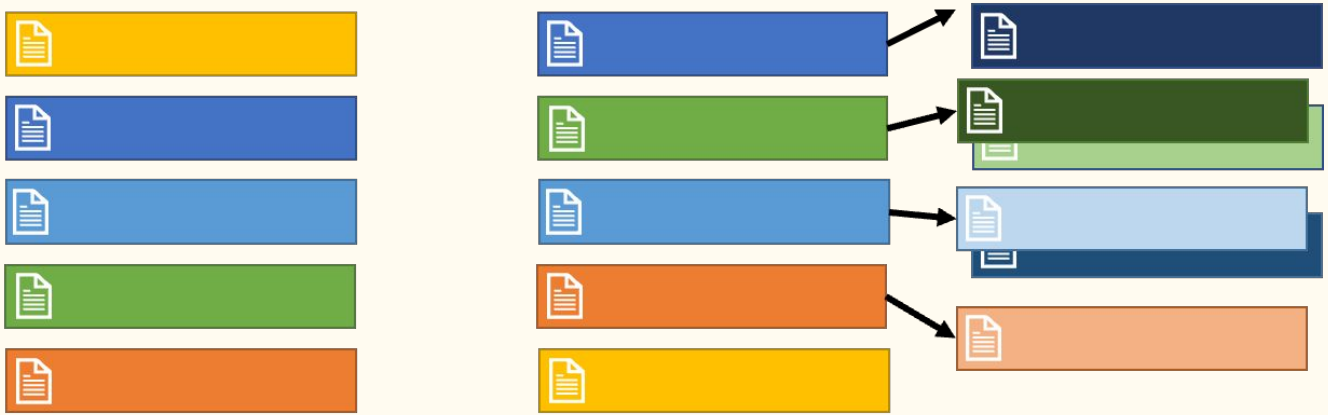
TAS-B

Alternative: Adaptive LADR



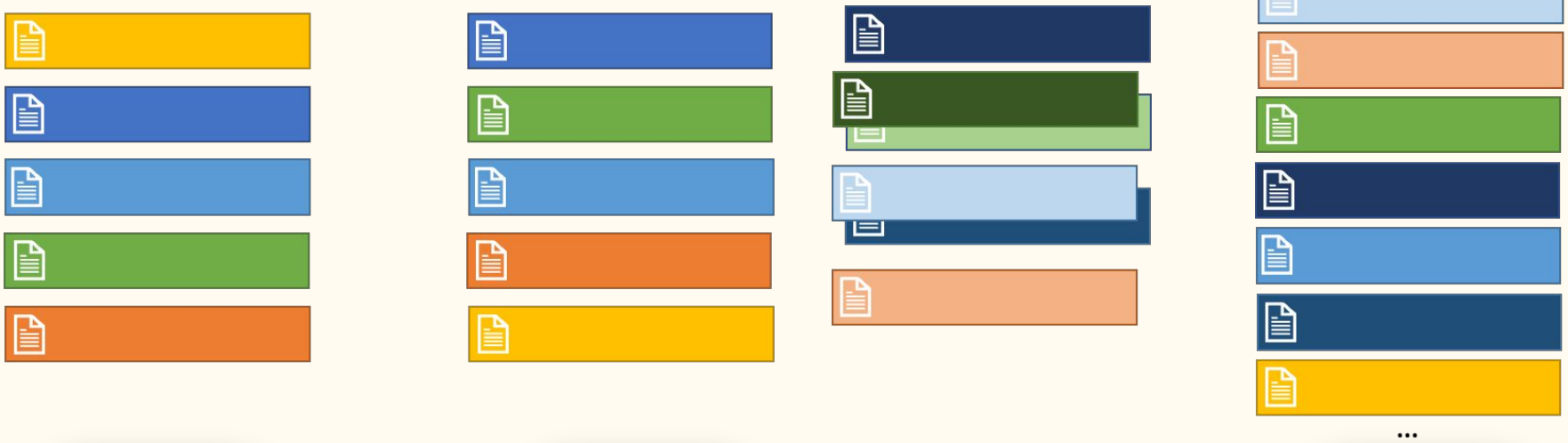
BM25 TAS-B Corpus Graph

Alternative: Adaptive LADR

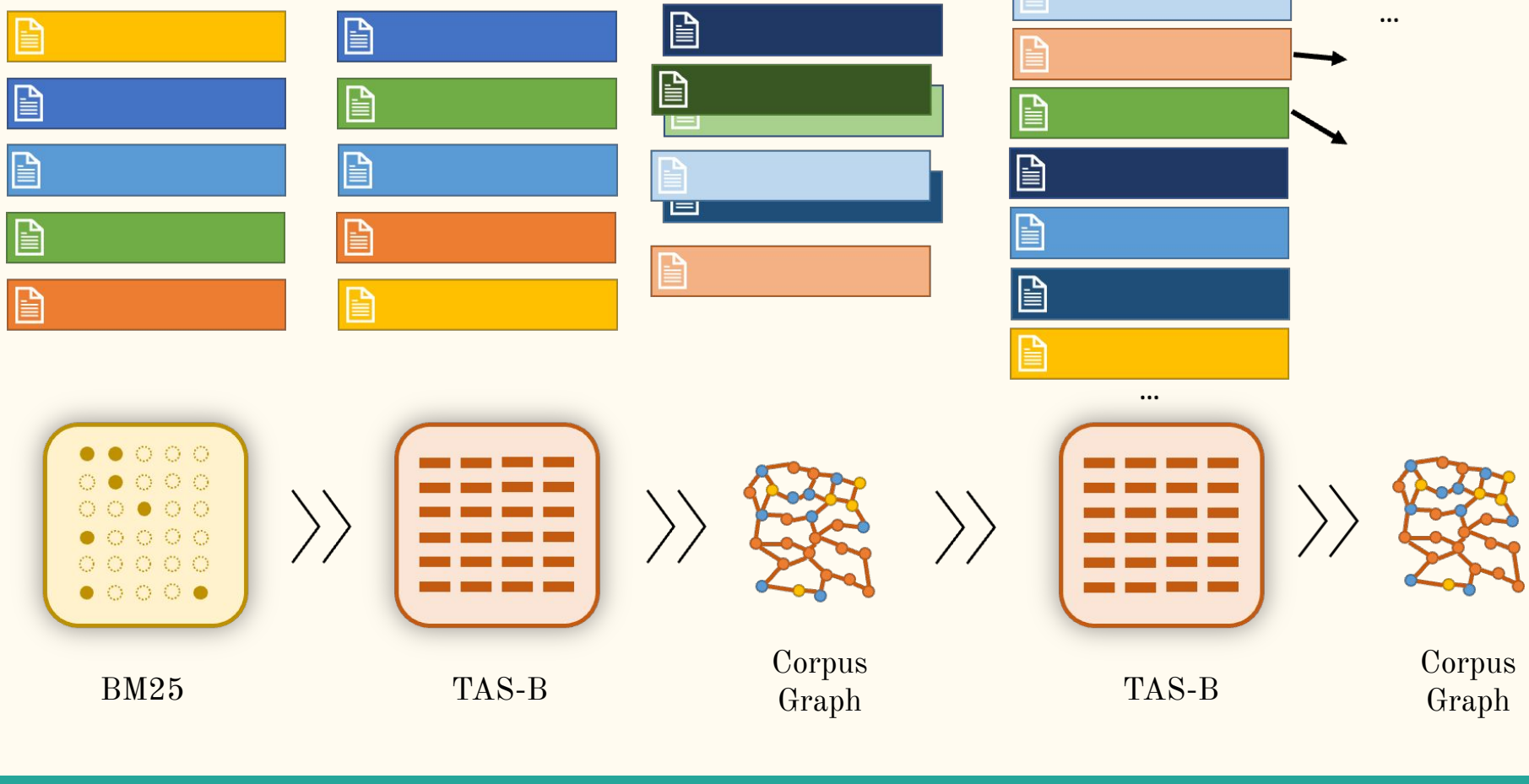


BM25 TAS-B Corpus Graph

Alternative: Adaptive LADR



Alternative: Adaptive LADR



Proactive LADR

1. Seed with lexical results
2. Add neighbors
3. Score seeds and their neighbors

Vector comparison: O(kn)
Storage: O(kD)

D is the corpus size, k is no. of neighbors

Adaptive LADR

1. Seed with lexical results
2. Add neighbors
3. Score seeds and their neighbors
4. Repeat 2 and 3 until no new neighbors can be identified

Vector comparison: O(D)*
Storage: O(kD)

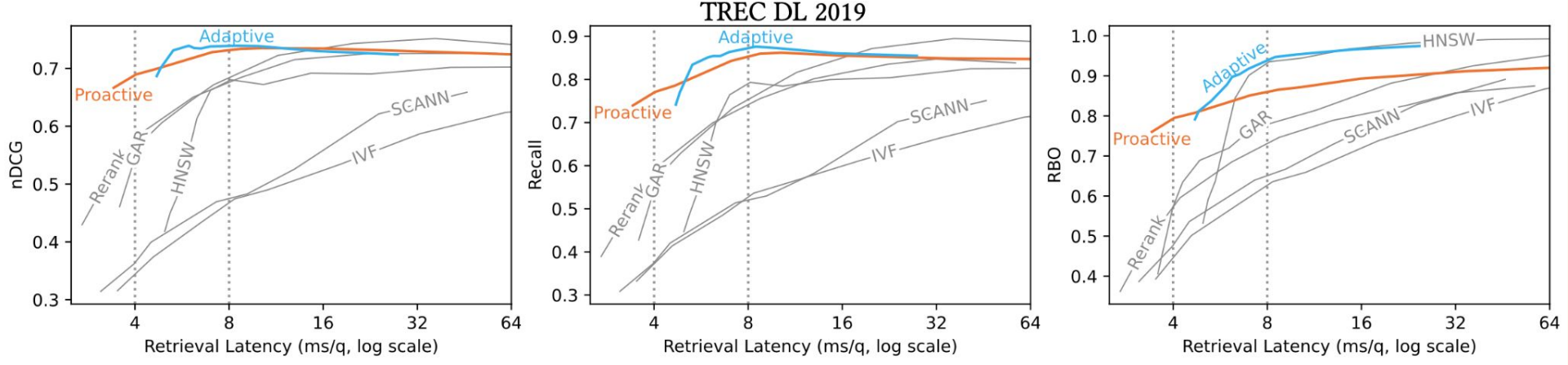
*O(D) is worst case - does not happen in practice due to clustering hypothesis

RQ1:
How does LADR compare to other approximation techniques in terms of effectiveness and efficiency?

1. Operational points: 4ms/query, 8ms/query
2. Dense retrieval models: TAS-B, RetroMAE, TCT-ColBERT-HNP, ANCE
3. Baselines: IVF, ScaNN, HNSW, GAR, Re-ranking
4. Datasets: DL 2019, DL 2020, Dev (small)

RQ1: nDCG and Recall@1k

Statistically significant improvements observed across exhaustive evaluations!
Effectiveness-efficiency Pareto-frontier created by LADR among approx k nearest neighbor techniques!



RQ2:
Is LADR applicable to a variety of single-representation dense retrieval models?

1. Dense retrieval models: TAS-B, RetroMAE, TCT-ColBERT-HNP, ANCE
2. Statistically significant results observed on all four dense retrieval models.
3. Hence, LADR can be used on top of wide range of dense retrieval models.

Proactive LADR
DL 2019 ~ 4ms

Dense Retrieval Model	nDCG@10	Recall@1k
TAS-B	0.690	0.771
RetroMAE	0.691	0.765
TCT-Colbert-HNP	0.680	0.747
ANCE	0.645	0.751

Adaptive LADR
DL 2019 ~ 8ms

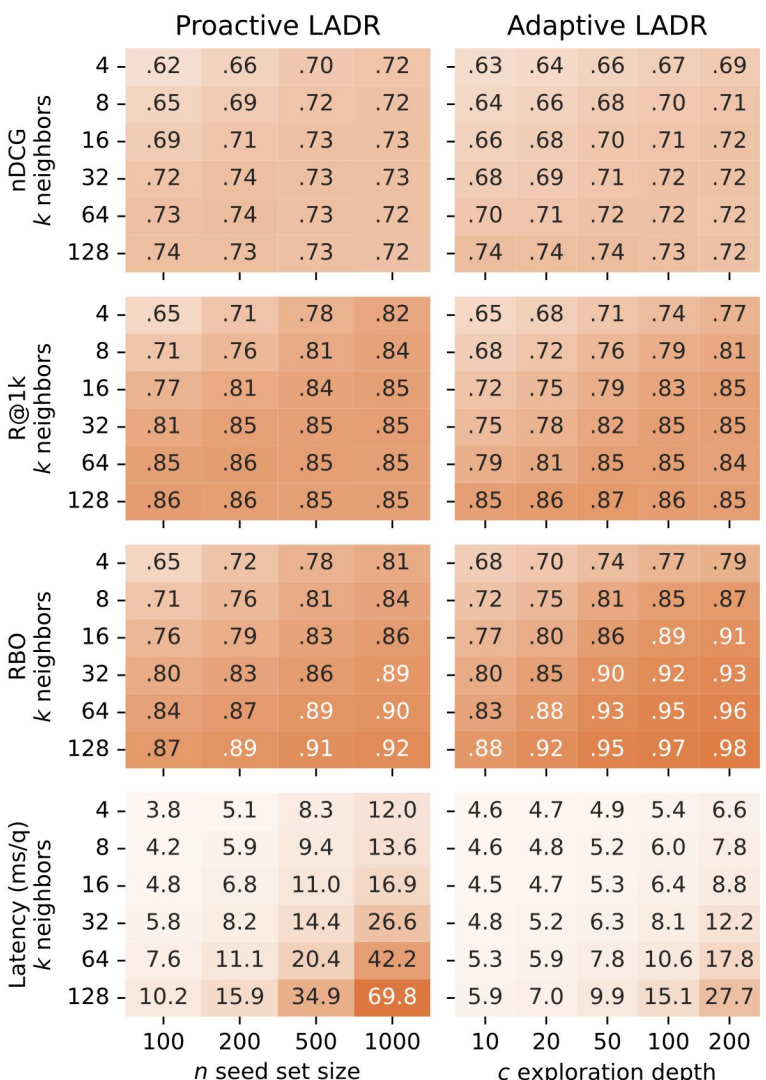
Dense Retrieval Model	nDCG@10	Recall@1k
TAS-B	0.738	0.872
RetroMAE	0.740	0.866
TCT-Colbert-HNP	0.729	0.848
ANCE	0.665	0.820

RQ3:
What are the computational overheads of LADR?

RQ4:
How do the parameters introduced by LADR affect the effectiveness and efficiency of LADR?

RQ3 and RQ4: Overhead and Parameters

1. Latency increases with increase in no. of neighbors, seed set size and exploration depth but so does RBO.
2. Best nDCG and Recall is obtained with higher no. of neighbors but lower seed set size.
3. Adaptive LADR should be preferred over Proactive for larger time budgets.
4. k, c, n parameters introduced in LADR can be tuned deliver best performance in given time budget.
5. Adaptive LADR: 0.98 RBO in 27.7ms/q
6. Proactive LADR: 0.92 RBO in 69.8ms/q



RQ5: Is an exact nearest neighbor graph needed for LADR to be effective?

Document Proximity Sources - Statistically equivalent performance observed across:

a. Exact
b. HNSW
c. BM25

Method	nDCG@10	Recall@1k
Exact	0.730	0.850
Approx	0.731	0.845
BM25	0.732	0.835

RQ6: What are the trade-offs between proactive and adaptive LADR in terms of precision, recall and latency?

1. Adaptive LADR performs better than Proactive LADR under same time budget.
2. Proactive LADR can deliver good results in very low time budgets.
3. Adaptive LADR has a O(D) worst case time complexity (even though unlikely). While Proactive LADR has O(kn).

Lexically-Accelerated Dense Retrieval

- Reduces computational overhead of dense retrieval maintaining high efficacy
- Delivers a proactive and an adaptive strategy for optimal use of time budget
- Establishes a new Pareto frontier for low latency approximate dense retrieval
- Robust to parameters and alternate sources of document proximity

Hrishikesh Kulkarni[1], Sean MacAvaney[2], Nazli Goharian[1], Ophir Frieder[1]

[1]Georgetown University
[2]University of Glasgow