SEINE: SEgment-based Indexing for NEural information retrieval

Sibo Dong, Justin Goldstein, Grace Hui Yang
InfoSense, Georgetown University, USA

@SIGIR 2022 ReNeuIR Workshop

Two Pipelines for Neural IR

Representation-Based

$$\begin{array}{lccr} \text{Learn } E(d_i) \rightarrow \text{Store } E(d_i) \text{ in index}, \forall i & \searrow & & \\ & & \forall i, E(q) \cdot E(d_i) \rightarrow & \text{Top-k results} \\ \text{Learn } E(q) \rightarrow E(q) & \nearrow & & \text{by MIPS, ANN.} \end{array} \quad (1)$$

- Dense, representation-based retrievers:
    - DPR, SBERT, Condenser, ICT, RocketQA, ANCE, RepBERT, ColBERT, …
- Sparse, representation-based retrievers:
    - SparTerm, EPIC, SPLADE, FLOPS, …

Two Pipelines for Neural IR

Interaction-Based

$$\begin{array}{lccr} \text{Process } d_i \longrightarrow \text{Store}\langle v, d_i \rangle \text{ in index, } \forall i & \searrow & & \\ & & \text{Lookup } q \text{ in } v \longrightarrow \quad \begin{array}{c}\text{Top-k results} \\ \text{by } s(q, d_i),\end{array} & (2) \\ \text{Process } q \longrightarrow q & \nearrow & & \end{array}$$

- Dense, interaction-based retrievers:
  - MonoBERT, MonoT5
- Sparse, interaction-based retrievers:
  - DUET, KRNM, HiNT, DeepImpact, MatchPyramid DeepTileBars, …

Two Pipelines for Neural IR

Representation-Based

$$\begin{array}{lccr} \text{Learn } E(d_i) \rightarrow \text{Store } E(d_i) \text{ in index}, \forall i & \searrow & & \\ & & \forall i, E(q) \cdot E(d_i) \rightarrow & \text{Top-k results} \\ \text{Learn } E(q) \rightarrow E(q) & \nearrow & & \text{by MIPS, ANN.} \end{array} \tag{1}$$

Interaction-Based

$$\begin{array}{lccr} \text{Process } d_i \rightarrow \text{Store} \langle v, d_i \rangle \text{ in index}, \forall i & \searrow & & \\ & & \text{Lookup } q \text{ in } v \rightarrow & \text{Top-k results} \\ \text{Process } q \rightarrow q & \nearrow & & \text{by } s(q, d_i), \end{array} \tag{2}$$

Representation- vs. Interaction-Based

- Representation-based:
  - Popular at the moment
  - But,
    - Lower effectiveness
    - It “index” cannot be reused
    - Most research is in the “indexing” phase, few about “retrieval”
    - Q-D interaction is kept at a minimum

Representation-Based

$$\begin{array}{lcc} \text{Learn } E(d_i) \rightarrow \text{Store } E(d_i) \text{ in index}, \forall i & \searrow & \\ & & \forall i, E(q) \cdot E(d_i) \rightarrow \quad \text{Top-k results} \\ \text{Learn } E(q) \rightarrow E(q) & \nearrow & \text{by MIPS, ANN.} \end{array} \tag{1}$$

Interaction-Based

$$\begin{array}{lcc} \text{Process } d_i \rightarrow \text{Store} \langle v, d_i \rangle \text{ in index}, \forall i & \searrow & \\ & & \text{Lookup } q \text{ in } v \rightarrow \quad \text{Top-k results} \\ \text{Process } q \rightarrow q & \nearrow & \text{by } s(q, d_i), \end{array} \tag{2}$$

Representation- vs. Interaction-Based

- Interaction-Based
  - Higher effectiveness
  - Its pre-neural example is BM25— a long time winner
  - But,
    - Comp. cost is prohibitive
    - Do not have an index

Representation-Based

$$
\begin{array}{lccc}
\text{Learn } E(d_i) \rightarrow \text{Store } E(d_i) \text{ in index}, \forall i & \searrow & & \\
 & & \forall i, E(q) \cdot E(d_i) \rightarrow & \text{Top-k results} \\
\text{Learn } E(q) \rightarrow E(q) & \nearrow & & \text{by MIPS, ANN.}
\end{array}
\tag{1}
$$

Interaction-Based

$$
\begin{array}{lccc}
\text{Process } d_i \rightarrow \text{Store} \langle v, d_i \rangle \text{ in index}, \forall i & \searrow & & \\
 & & \text{Lookup } q \text{ in } v \rightarrow & \text{Top-k results} \\
\text{Process } q \rightarrow q & \nearrow & & \text{by } s(q, d_i),
\end{array}
\tag{2}
$$

Inverted Index for Interaction-Based NeuIR

- Fast lookup
- DeepCT, HDCT, TILDE
  - Indices are tailored to specific retrievers
- SNRM
  - Put latent terms in index (like semantic indexing)

Our Proposal: SEINE

- SEINE: Segment-based neural indexing
- Focus on interaction-based neural IR
- General, flexible, reusable index design to support Q-D interaction at different granularities
  - Document-level: BM25, TILDE
  - Term-level: KRNM,
  - Topic-level: HiNT, DeepTileBars
- Decompose retrieval methods and identify the atomic Q-D interaction units

System Overview

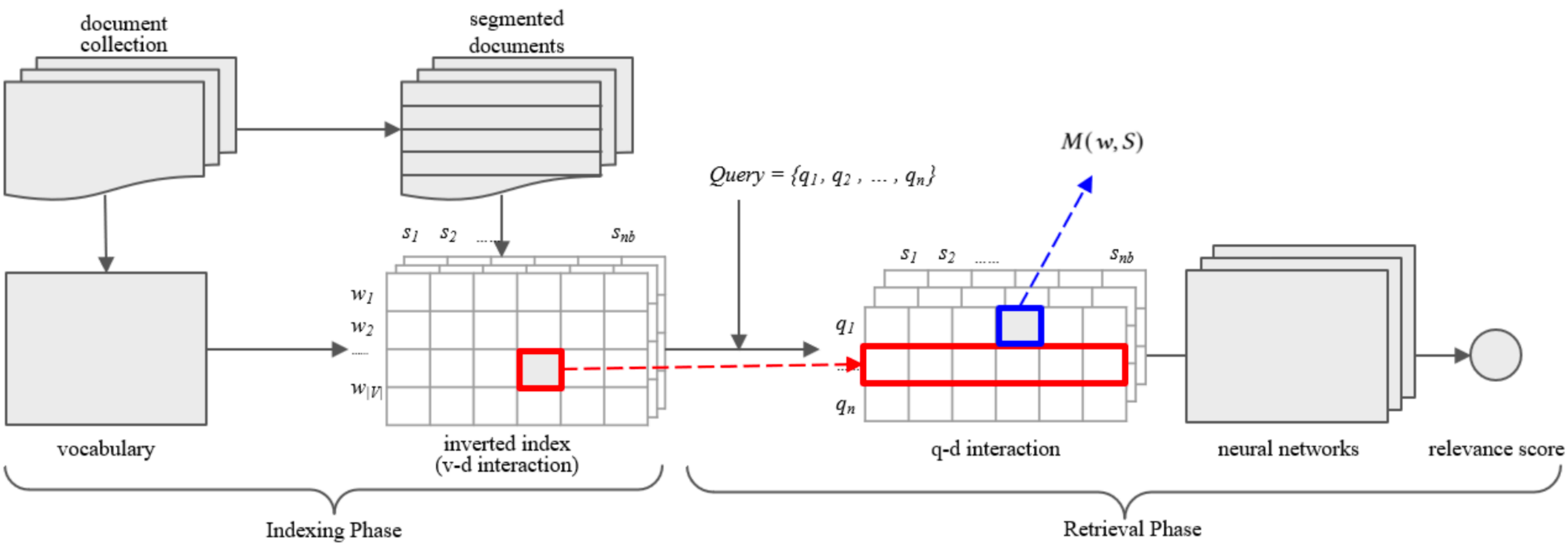


Figure 1: SEINE: SEgment-based Indexing for NEural information retrieval.

Steps

- Pre-process the document collection
- Segment documents (to support various interaction granularities)
  - Done by TextTiling
- Store *atomic* Q-D interactions in inverted index
- Accelerate with Spark

Atomic Interaction Functions

- Term frequency (e.g. both non-neural and neural retrievers such as DeepTileBars)
- Inverted Document Frequency (e.g. both non-neural and neural retrievers such as HiNT)
- Operations over BERT embeddings
  - Linear aggregation (e.g. DeepCT, HDCT)
  - Max operation (e.g. ColBERT, EPIC)
  - Multi-layer Perceptron (e.g. DeepImpact)
- Kernel functions
  - Dot product (e.g. MatchPyramid, dense retrievers such as COIL)
  - Gaussian kernel (e.g. KRNM, DeepTileBars)
  - Cosine Similarity (e.g. KNRM, HiNT)
- Conditional probabilities (e.g. TILDE)

Parallel Programming by Spark

Algorithm 1 Spark pseudo-code for indexing.

Initialize Spark environment and configuration
Import functions segmentation, interaction
Vocab $\leftarrow RDD\{w_1, w_2, ..., w_{|V|}\}$ ▷ create RDD
Corpus $\leftarrow RDD\{d_1, d_2, ..., d_{|C|}\}$ ▷ create RDD
Segmts $\leftarrow$ Corpus.*map* (segmentation) ▷ document segmentation
Cart $\leftarrow$ Vocab.$cartesian$(Segmts)
Index $\leftarrow$ Cart.*map* (interaction) ▷ calculate M as in § 2.3
Index $\leftarrow$ Index.*filter* $(tf > \sigma_{index})$
Index $\leftarrow$ Index.*reshape* ▷ v-S to v-d
Index.*saveAsPickleFile* ()

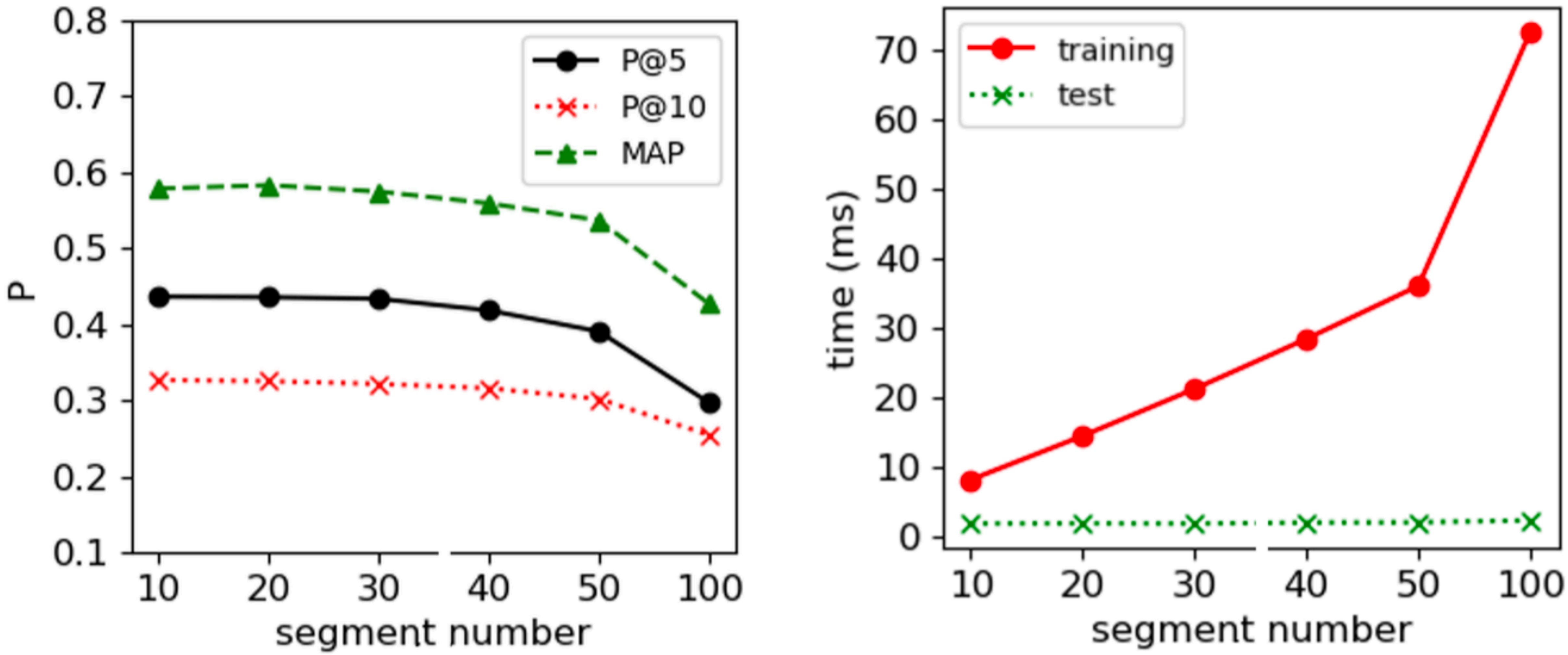
Main Results - LETOR MQ2007

		Effectiveness					Efficiency			
Indexing	**Retrieval**	P@5	P@10	MAP	nDCG@5	nDCG@10	Training (ms)		Test (ms)	
No Index	Dot Product	0.328	0.344	0.418	0.282	0.332				
	KNRM	0.355	0.382	0.461	0.379	0.412	42.63		16.70	
	HiNT⋄	0.461	0.418	0.505	0.463	0.490	1139.34		1009.11	
	DeepTileBars	0.429	0.408	0.474	0.398	0.434	163.91		73.68	
InvIdx‡	BM25⋄	0.388	0.366	0.456	0.384	0.414				
SNRM	Dot Product	0.288*	0.307*	0.368*	0.254*	0.302*				
	KNRM	0.322*	0.347*	0.417*	0.337*	0.404	35.56	1.2×	13.17	1.3×
	HiNT	0.401*	0.358*	0.423*	0.379*	0.402*	958.03	1.2×	860.76	1.1×
	DeepTileBars	0.281*	0.304*	0.359*	0.238*	0.290*	131.43	1.2×	56.97	1.3×
SEINE	Dot Product	0.328	0.344	0.418	0.282	0.332				
	BM25 w/ DeepCT weight	0.315	0.327	0.397	0.266	0.314				
	KNRM	0.342	0.372	0.447	0.374	0.401	11.67†	3.7×	1.22†	13.7×
	HiNT	0.453	0.409	0.492	0.452	0.483	834.64	1.4×	706.42	1.4×
	DeepTileBars	0.412	0.404	0.468	0.391	0.427	22.62†	7.4×	2.67†	28.1×

Main Results - LETOR MQ2008

		Effectiveness					Efficiency			
Indexing	**Retrieval**	P@5	P@10	MAP	nDCG@5	nDCG@10	Training (ms)		Test (ms)	
No Index	Dot Product	0.333	0.281	0.462	0.411	0.183				
	KNRM	0.355	0.355	0.472	0.499	0.225	41.27		16.86	
	HiNT⋄	0.367	0.255	0.505	0.501	0.244	1139.34		1009.11	
	DeepTileBars	0.425	0.321	0.567	0.548	0.259	167.20		74.96	
InvIdx‡	BM25⋄	0.337	0.245	0.465	0.461	0.220				
SNRM	Dot Product	0.380†	0.300†	0.513†	0.483†	0.223†				
	KNRM	0.303*	0.229*	0.417*	0.417*	0.199*	36.78	1.1×	12.68	1.3×
	HiNT	0.291*	0.209*	0.401*	0.445*	0.221*	941.09	1.2×	904.16	1.1×
	DeepTileBars	0.356*	0.291*	0.481*	0.434*	0.200*	134.10	1.2×	58.14	1.3×
SEINE	Dot Product	0.333	0.281	0.462	0.411	0.183				
	BM25 w/ DeepCT weight	0.307	0.239	0.448	0.400	0.197				
	KNRM	0.346	0.252	0.462	0.485	0.218	11.58†	3.6×	1.24†	13.6×
	HiNT	0.362	0.250	0.485	0.489	0.236	828.84	1.4×	687.85	1.5×
	DeepTileBars	0.422	0.322	0.569	0.544	0.255	22.62†	7.4×	2.67†	28.1×

Impact of Segment Size



- Effectiveness seems to peak at 20-ish segments per document (roughly 200~270 words; close to natural paragraph breaks)
- Training efficiency changes dramatically
- No big impact on test efficiency

Conclusion

- SEINE is a general, reusable indexing framework
    - Store atomic interactions between query and segments
    - Supports various interaction-based neural retrievers
- Currently, SEINE does not support indexing for MonoBERT
    - Future work