Phrase Retrieval Learns Passage Retrieval, Too

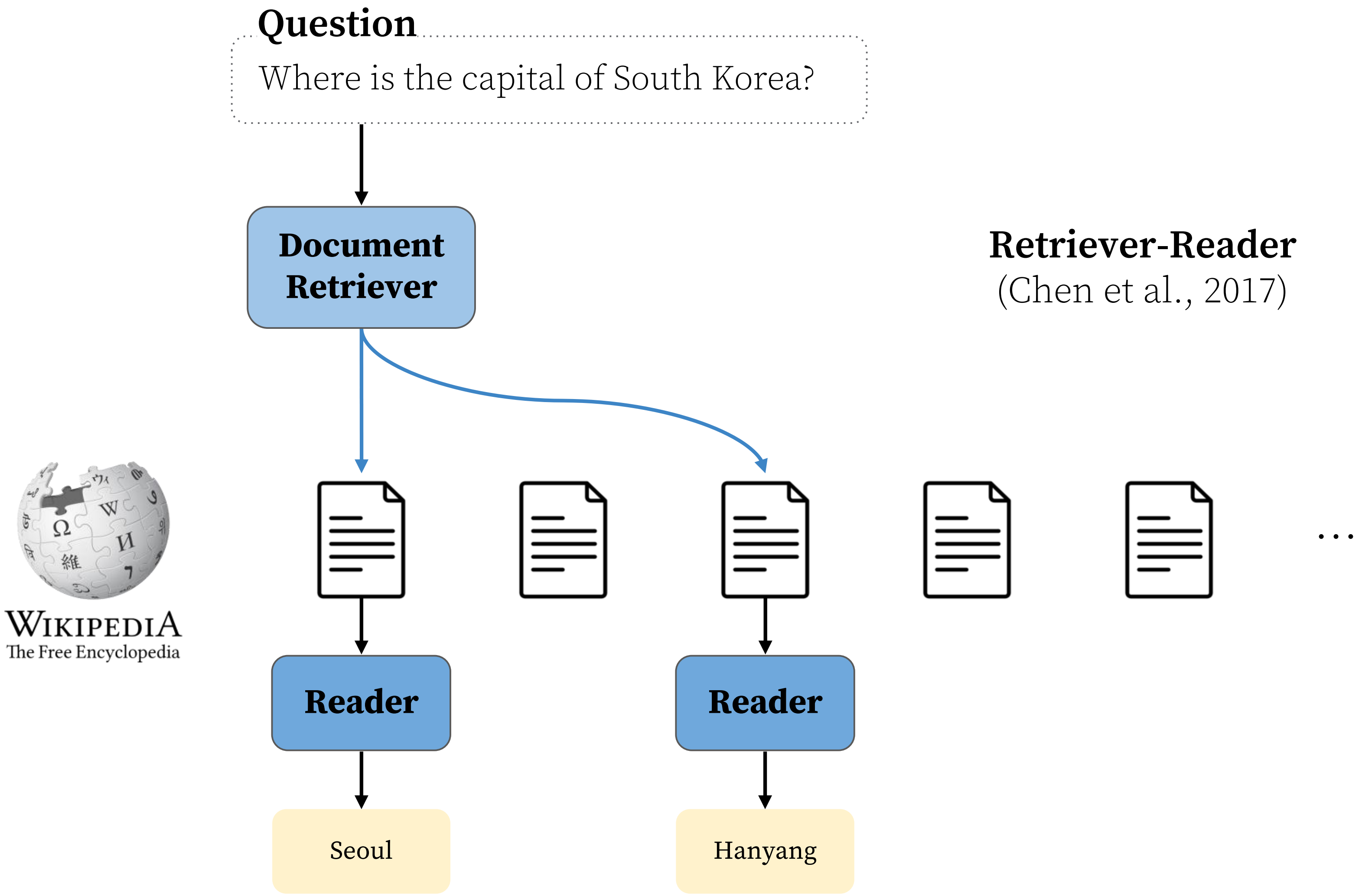
Princeton University 이진혁 연구원

CONTENTS

01

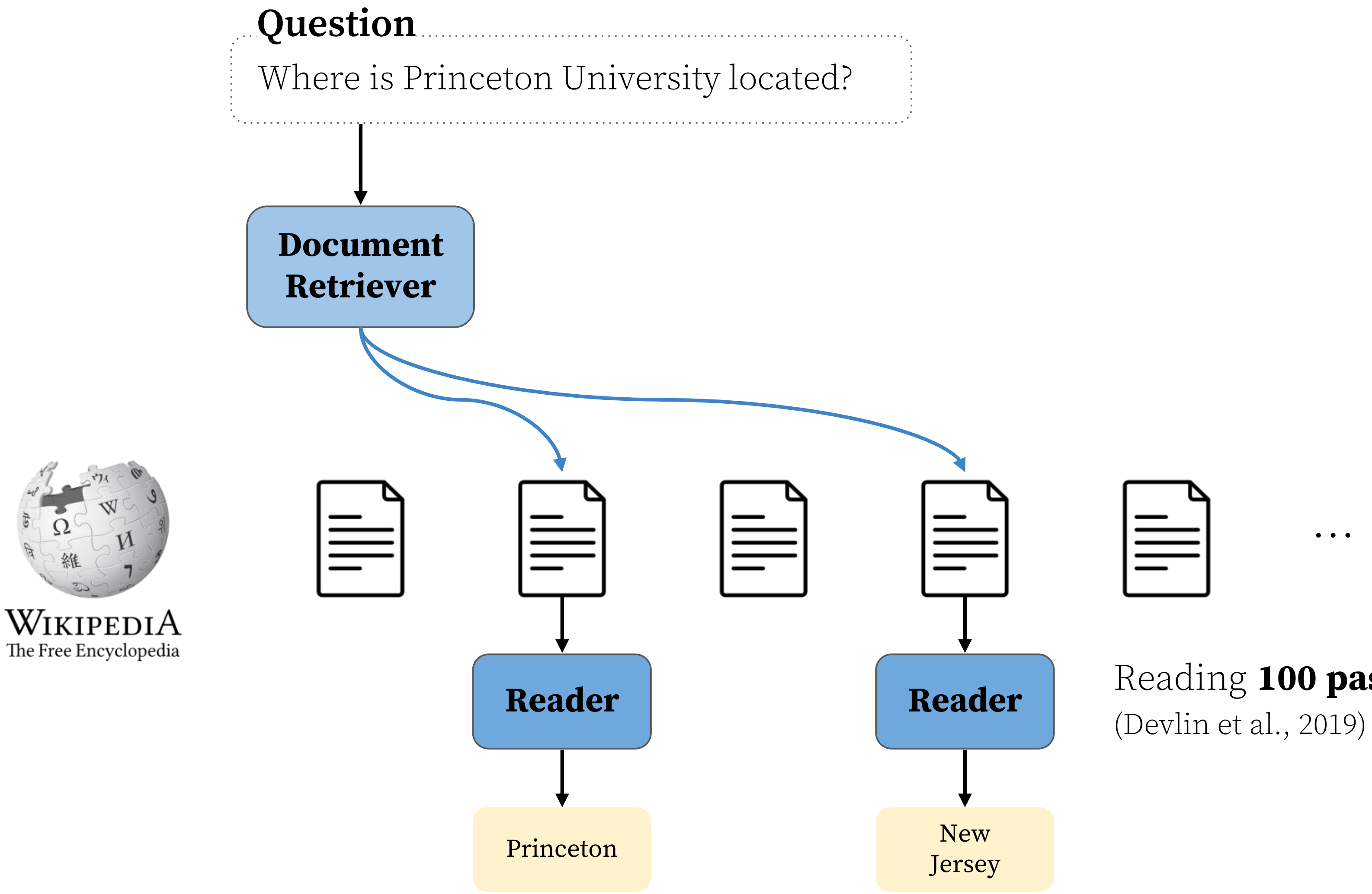
Background

01 Open-Domain Question Answering



Retriever-Reader
(Chen et al., 2017)

01 Open-Domain Question Answering

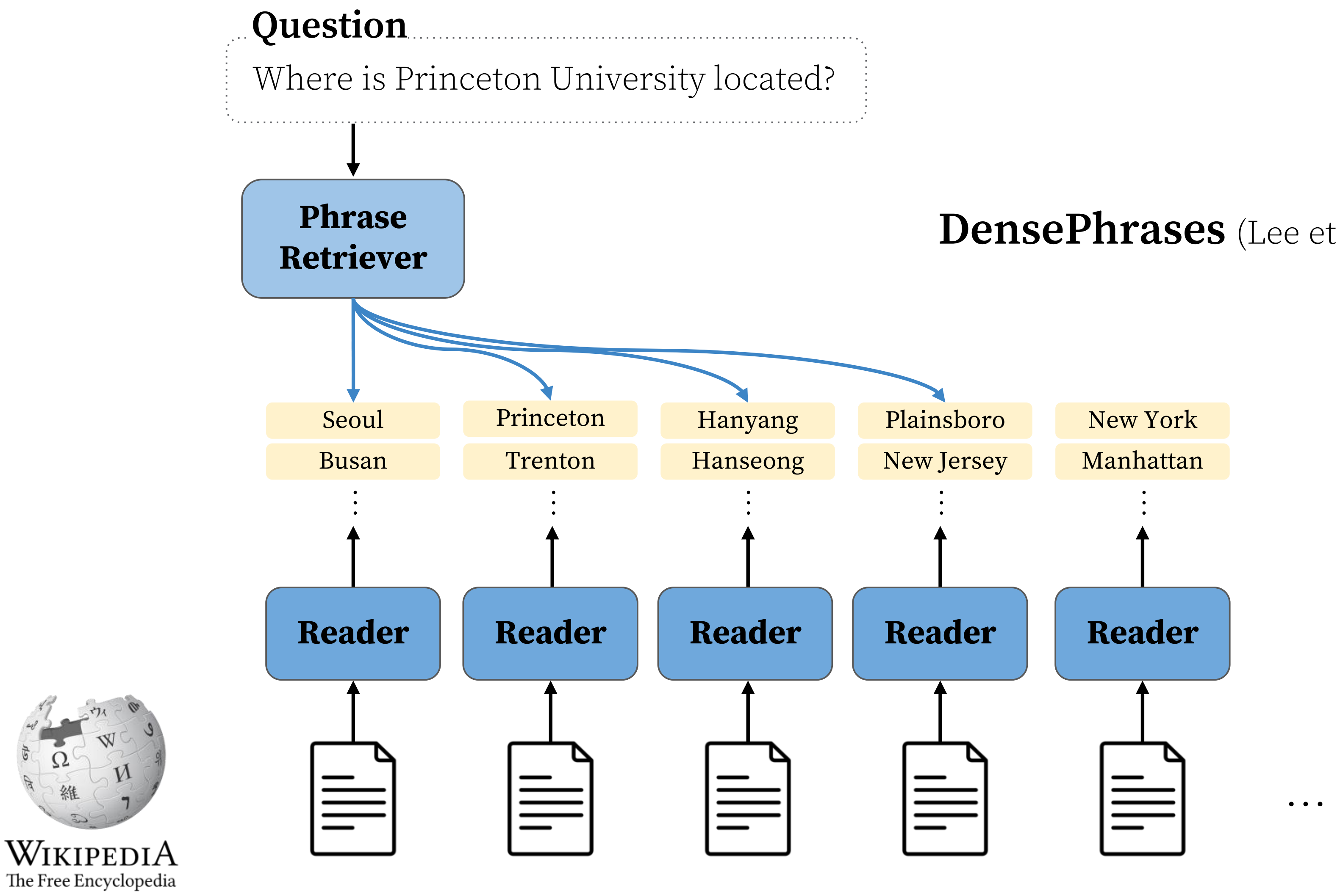


Reading **100 passages** with **BERT** (Devlin et al., 2019) is inefficient!

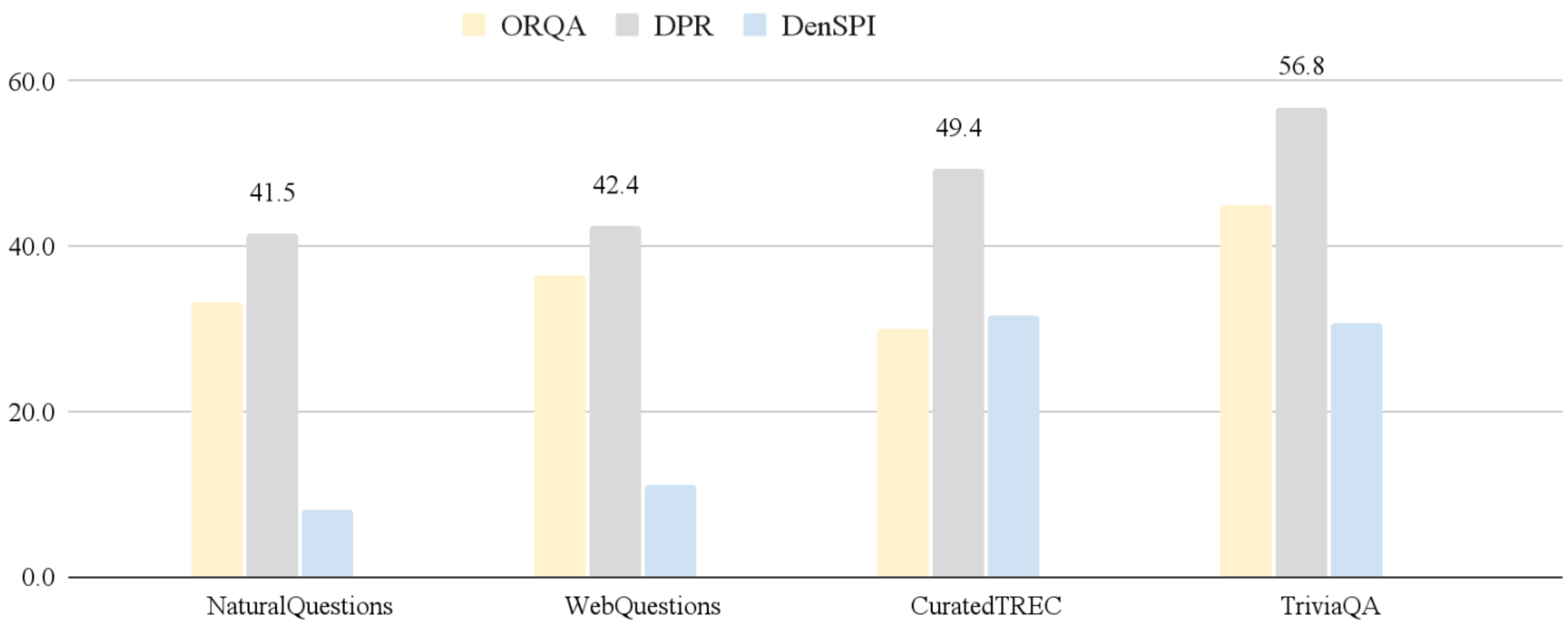
01 **Phrase Retrieval** for Open-Domain QA

Phrase = any contiguous segment of text up to L words (Seo et al., 2019)

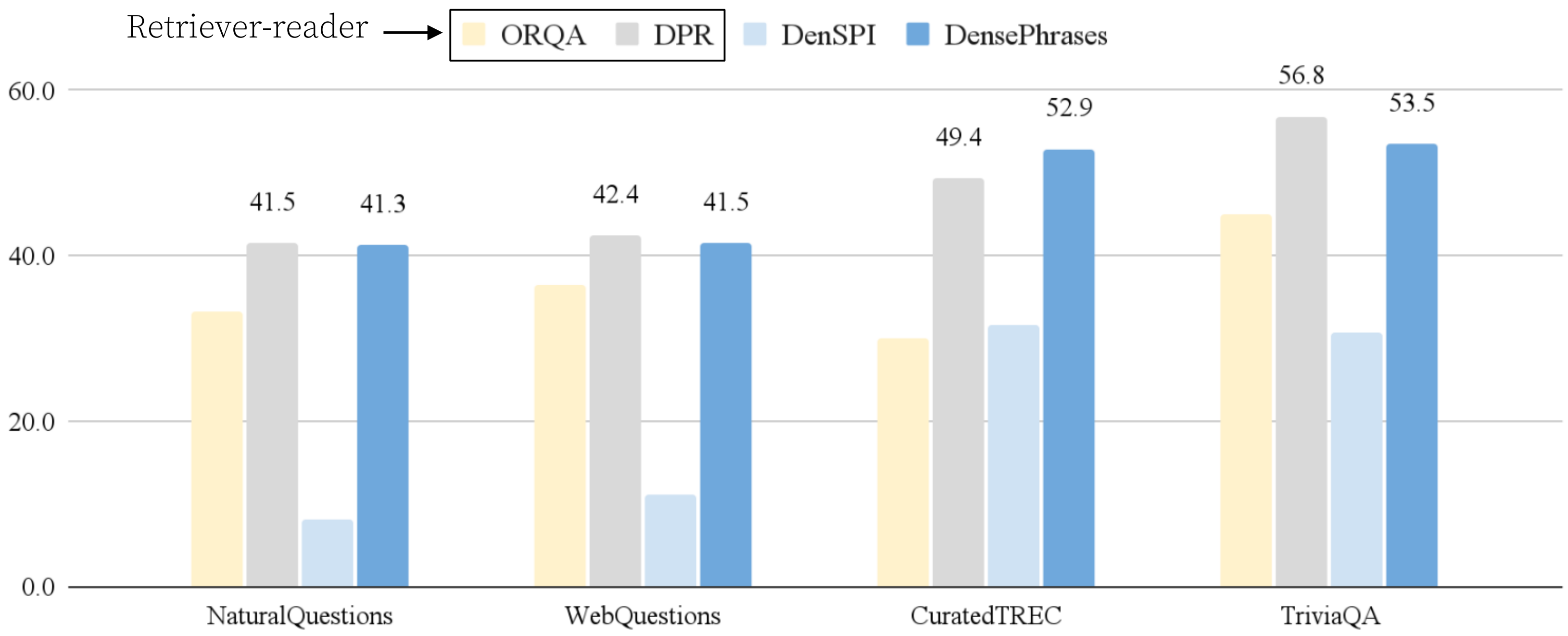
DensePhrases (Lee et al., 2021)



01 Phrase Retrieval is **Accurate** and **Fast**

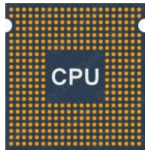


01 Phrase Retrieval is **Accurate** and **Fast**



Without any reader model, phrase retrieval is **competitive** with retriever-reader approaches.

0.04 Q/sec (DPR) < **13.6 Q/sec** (DensePhrases)



Dense phrase retrieval makes open-domain QA **fast** and **simple**!

02

Research Motivation

02 Fixed Granularity for Text Retrieval

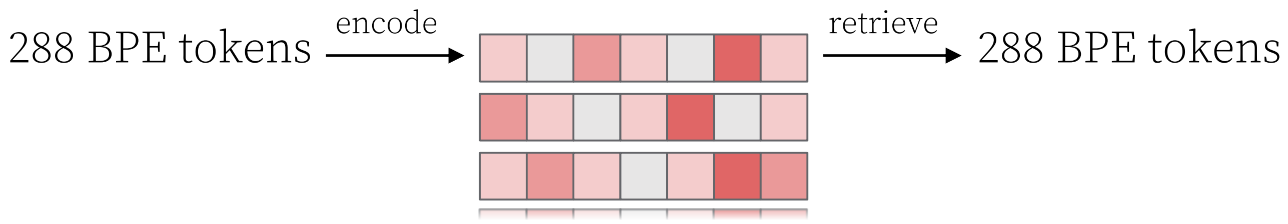
Sentence Retrieval

SBERT (Reimers et a., 2019), SimCSE (Gao et al., 2021): **1 sentence**

Passage Retrieval

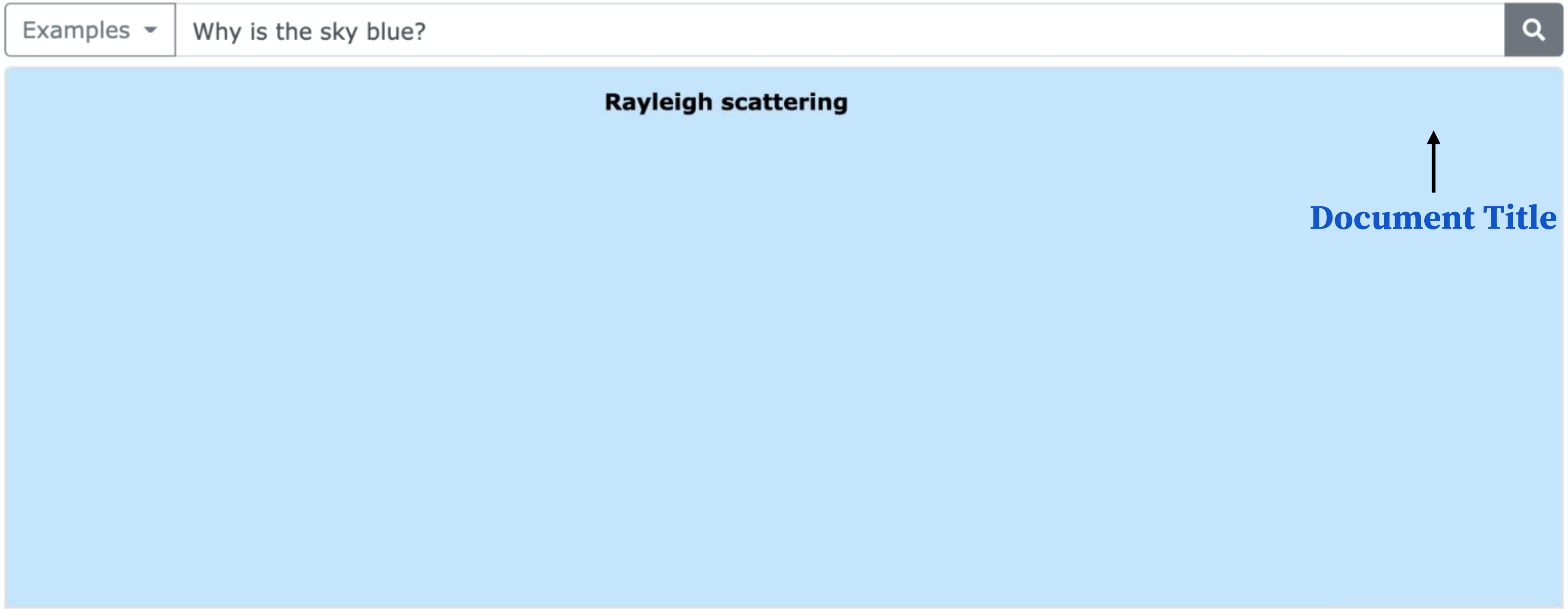
ORQA (Lee et al., 2019): **288 BPE tokens** for a passage

DPR (Karpukhin et al., 2020): **100 words** for a passage



Different index for **different** granularity?

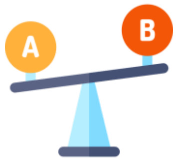
02 **Phrases** as a Basic Retrieval Unit



Retrieving **Phrases** $\underset{\nLeftarrow}{\Rightarrow}$ Sentences $\underset{\nLeftarrow}{\Rightarrow}$ Passages $\underset{\nLeftarrow}{\Rightarrow}$ Documents $\underset{\nLeftarrow}{\Rightarrow}$ ...

Single index for **multi** granularity!

02 In This Talk ...



Q1: Is this **better** than passage retrievers?

Experiment #1: Passage Retrieval / Experiment #2: Open-domain QA

Q2: **Why** does this work?

Analysis / Experiment #3: Entity Linking & Dialogue

Q3: How **efficient** is this?

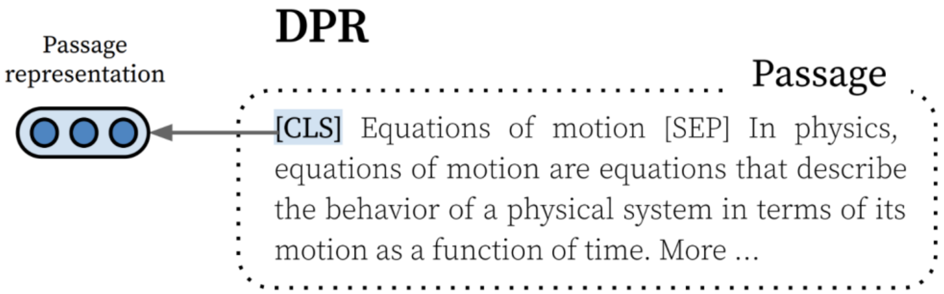
Phrase Filtering & Quantization-aware Fine-tuning

03

Formulation / Experiments #1, #2

03 Formulation

Passage Retrieval

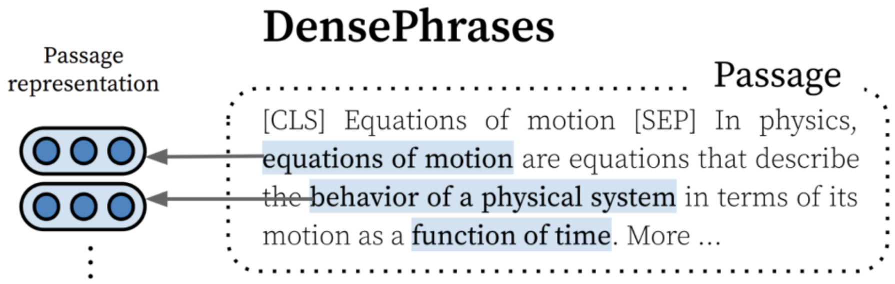


Question vector

$$f(p,q) = E_p(p)^\top E_q(q)$$

Single vector for each passage

Phrase-based Passage Retrieval

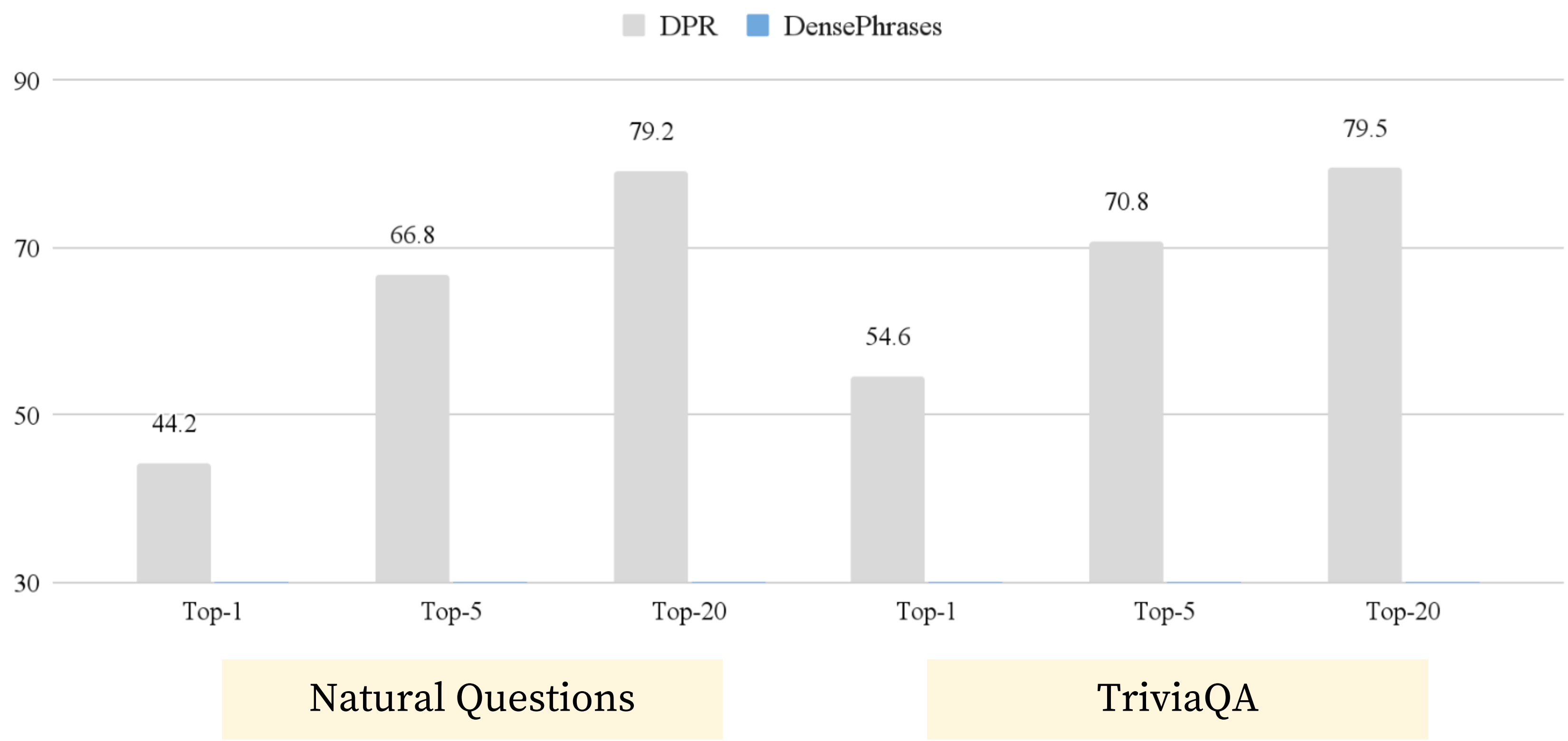


Phrase vector

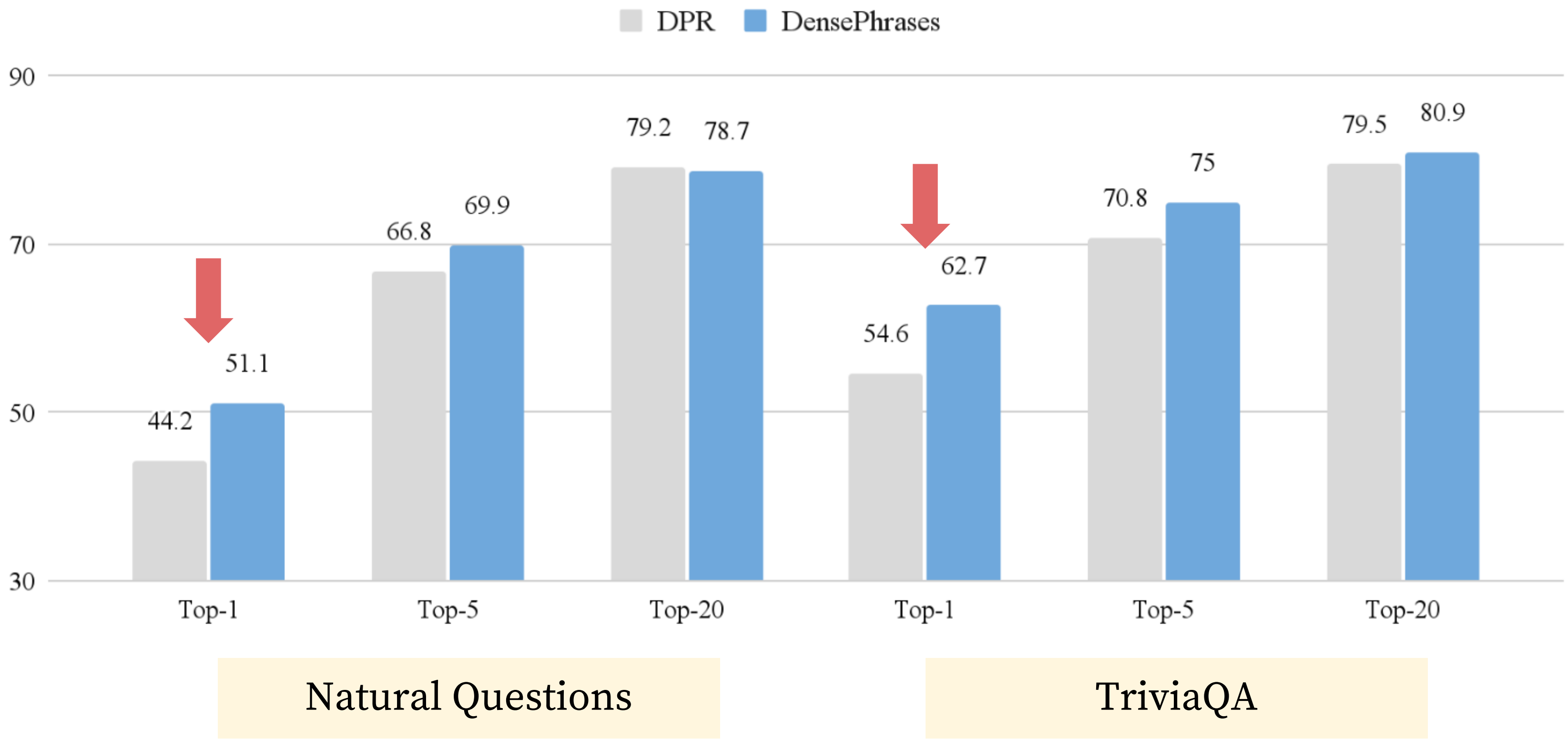
$$\tilde{f}(p,q) := \max_{s^{(p)} \in \mathcal{S}(p)} E_s(s^{(p)})^\top E_q(q)$$

Multiple (phrase) vectors for each passage

03 **Passage Retrieval:** DPR vs DensePhrases



03 **Passage Retrieval**: DPR vs DensePhrases

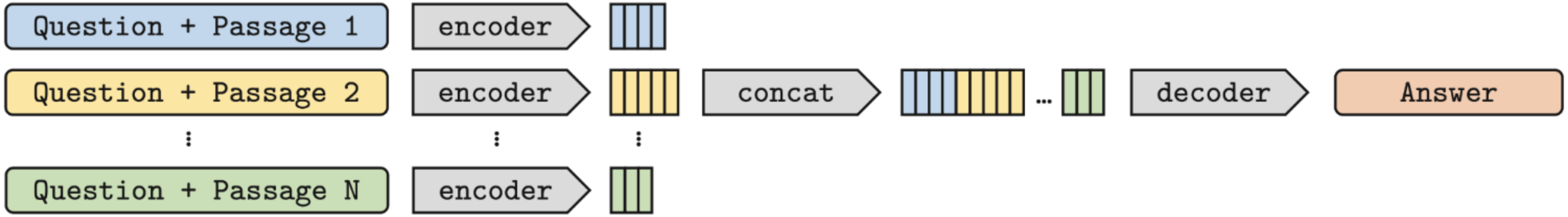


Without any re-training, **DensePhrases outperforms DPR** on passage retrieval!

Larger gains when **k is small**.

03 **Fusion-in-Decoder** for Open-domain QA

Izacard and Grave, 2021



Feeds top-k passages from **DPR** to **T5** (Raffel et al., 2020) to generate answers.

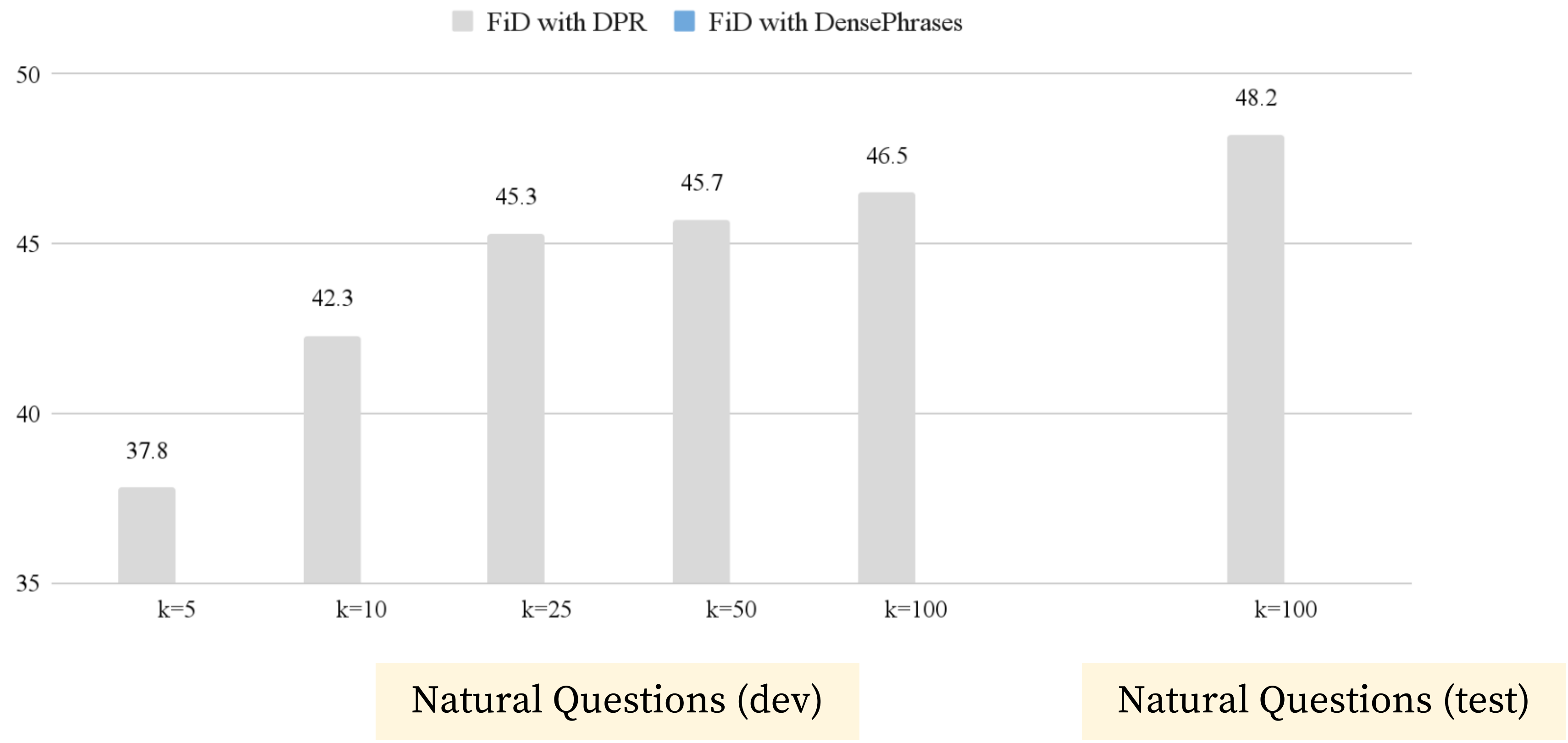
FiD achieves state-of-the-art performance **when k is large** (e.g., k=100).

Requires 64 **32GB** V100 GPUs for training!

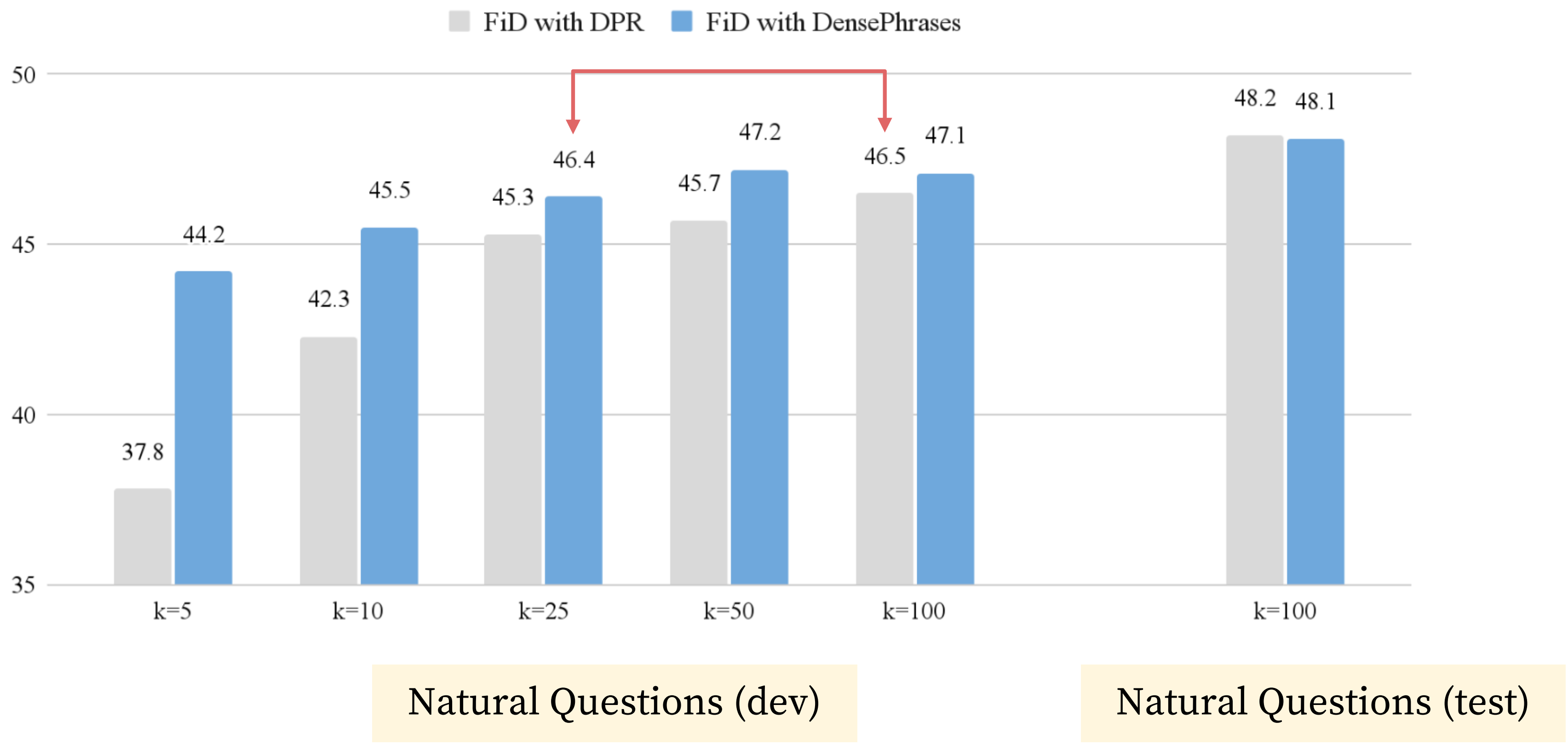
…

Feed top-k passages from **DensePhrases** to T5 to generate answers?

03 **Open-domain QA**: DPR vs DensePhrases



03 **Open-domain QA**: DPR vs DensePhrases



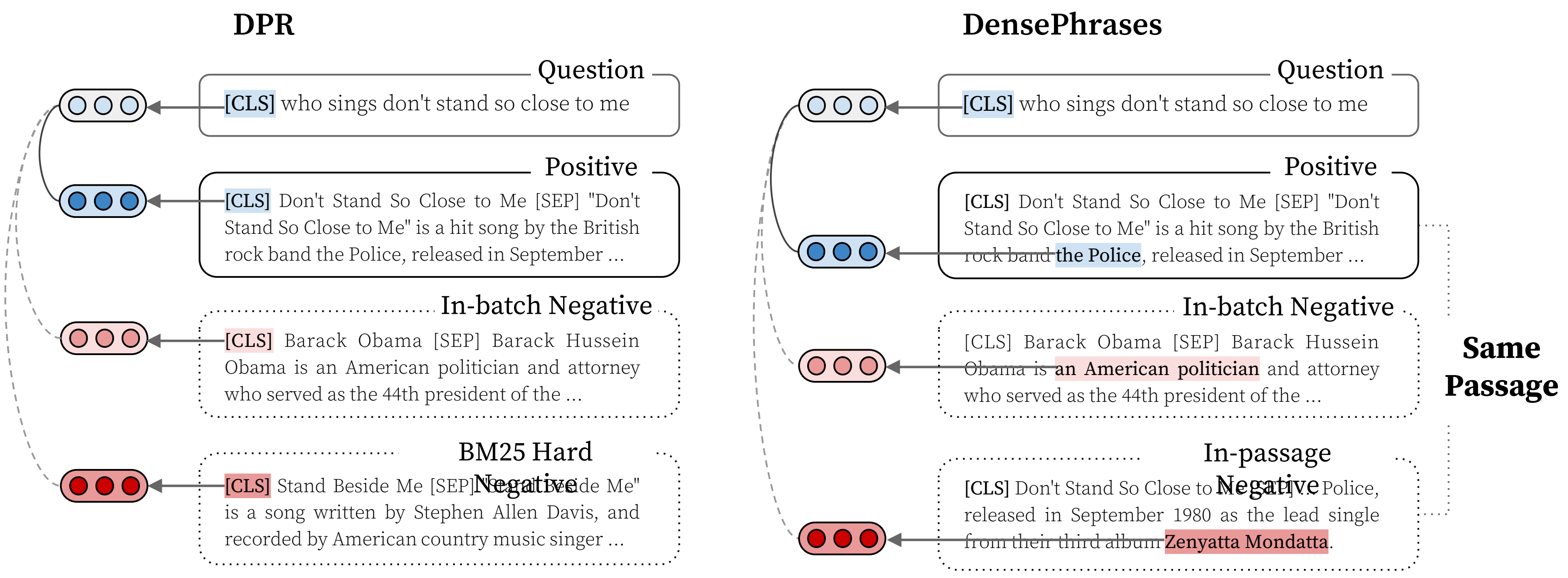
DensePhrases outperforms DPR on open-domain QA (+6.4 EM when k=5).

k=25~50 is enough for good performance (k ≤ 50 fits in **24GB**)

04

Analysis / Experiments #3

04 Why DensePhrases > DPR on Passage Retrieval?

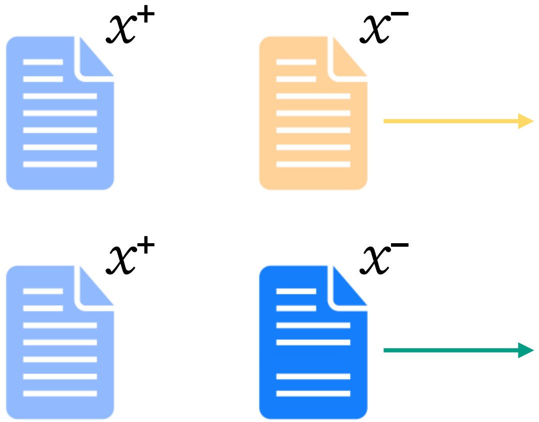


In-passage negatives in DensePhrases work similar to **BM25 hard negatives** in DPR!

04 Analysis with L_{topic} and L_{hard}

L_{topic}

L_{hard}



Topically different
(A random passage from Wikipedia)

Topically same,
Only x^+ has the answer
(the same passage without an answer)

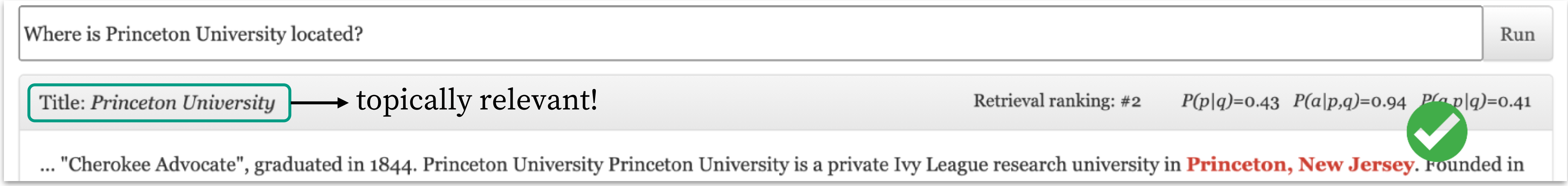
For both metrics, lower numbers are better.

DPR has good L_{topic} while **DensePhrases** has good L_{hard}.

04

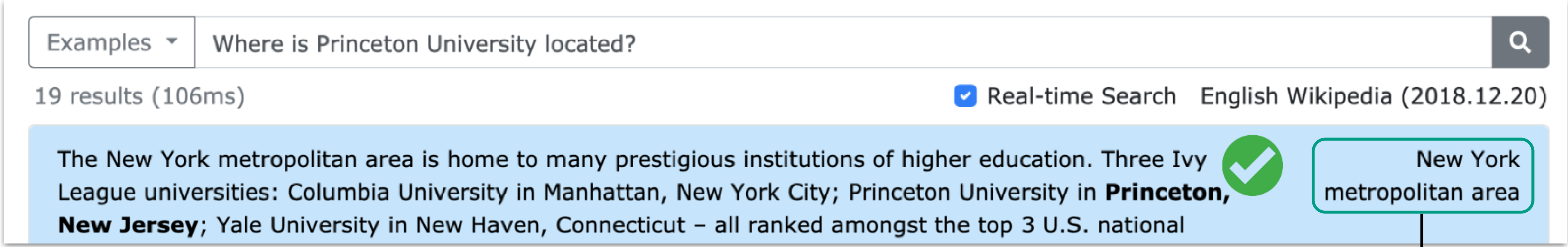
$\mathcal{L}_{\text{topic}}$ and $\mathcal{L}_{\text{hard}}$: What Do They Really Mean?

DPR (Karpukhin et al., 2020)



http://qa.cs.washington.edu:2020/

DensePhrases (Lee et al., 2021)



http://densephrases.korea.ac.kr

topically less relevant,
but still correct answer!

Good $\mathcal{L}_{\text{hard}}$ can give correct answer
even when **the passage is less relevant**.

04 But WHAT IF...? L_{topic} Matters?

For many coarse-granularity retrieval,
we need good L_{topic} !

Entity Linking

United Nations Security Council

[START_ENT] **Security Council** [END_ENT] members expressed concern on Thursday.

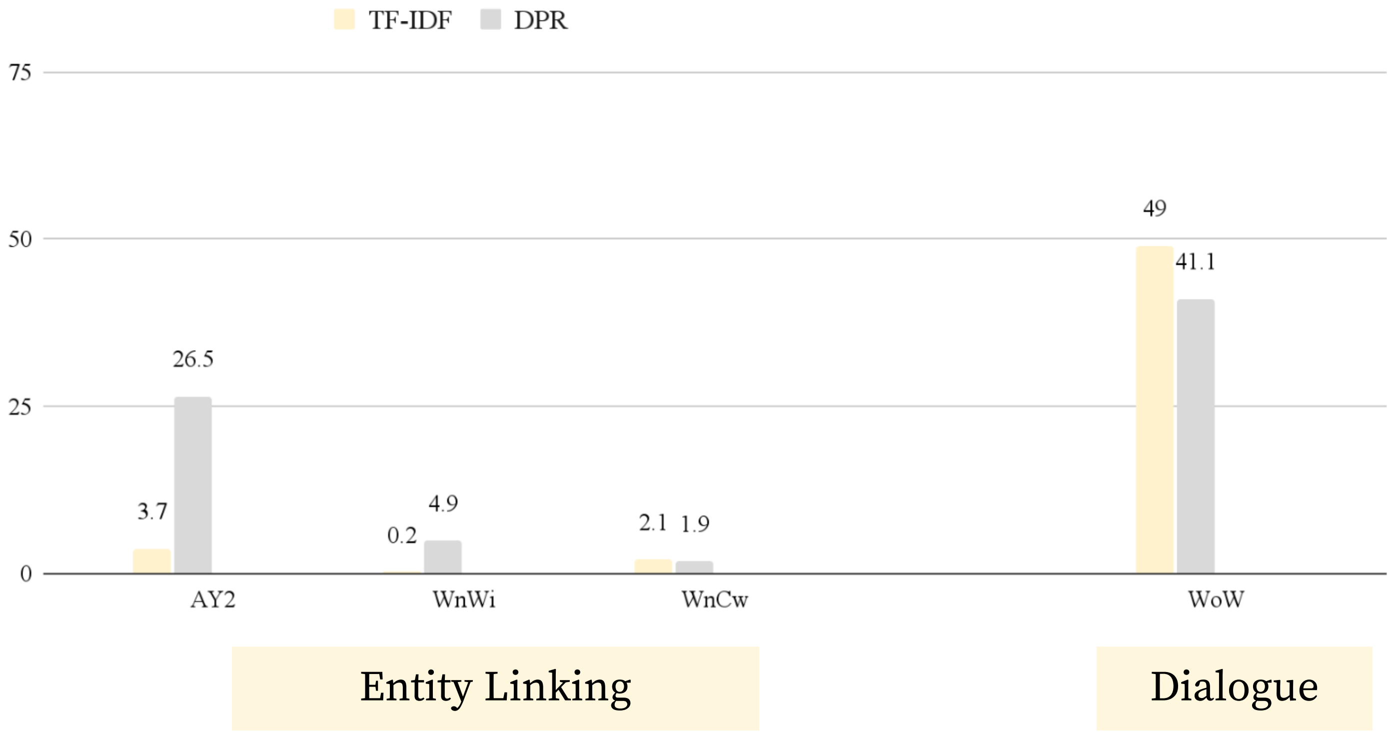
Knowledge-grounded Dialogue

Yamaha Corporation

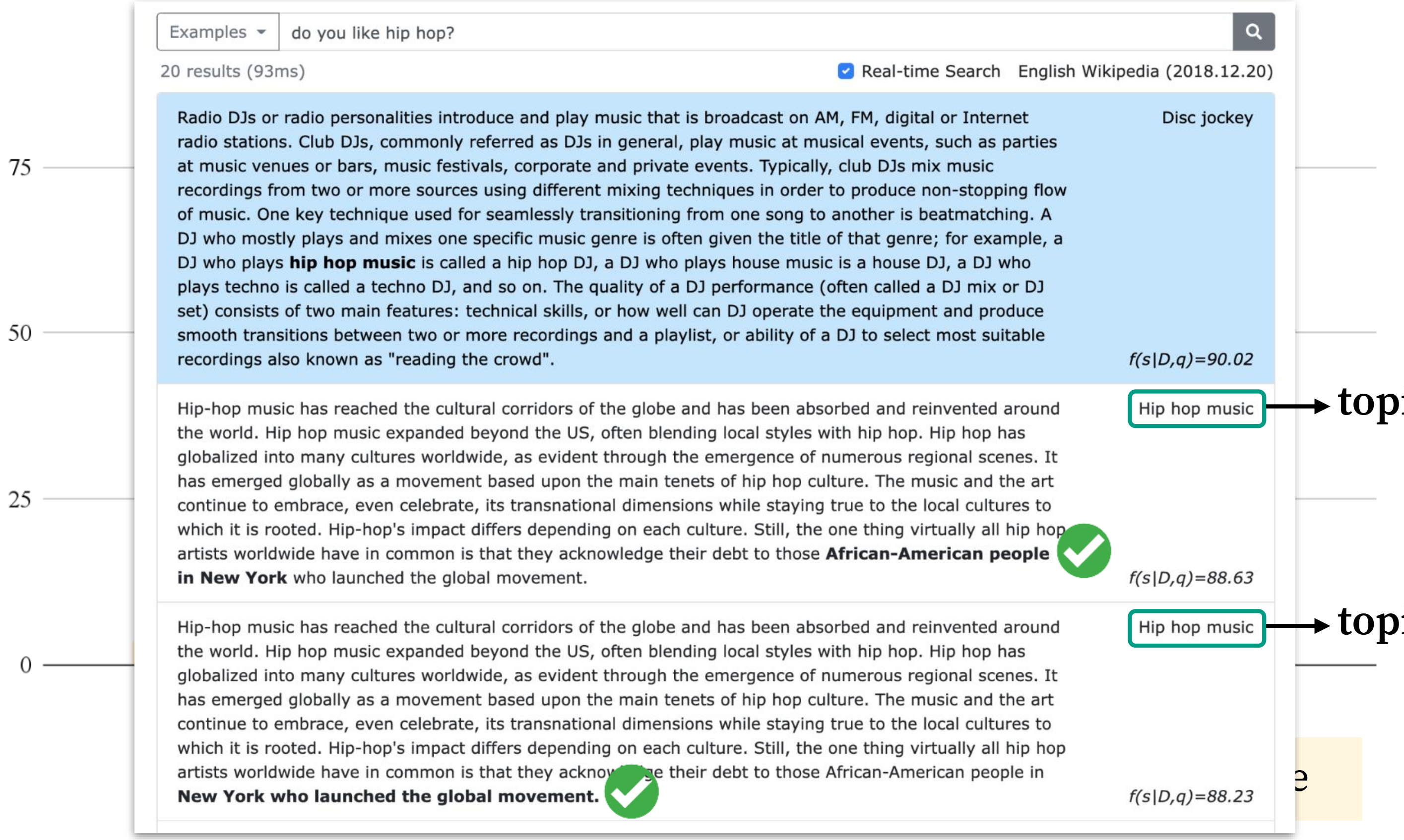
Have you heard of Yamaha? They started as a piano manufacturer in 1887!

Only **one document** is relevant (annotated) for each query!
(KILT; Petroni et al., 2021)

04 Retrieval for **Entity Linking** & **Dialogue**



04 Retrieval for **Entity Linking** & **Dialogue**

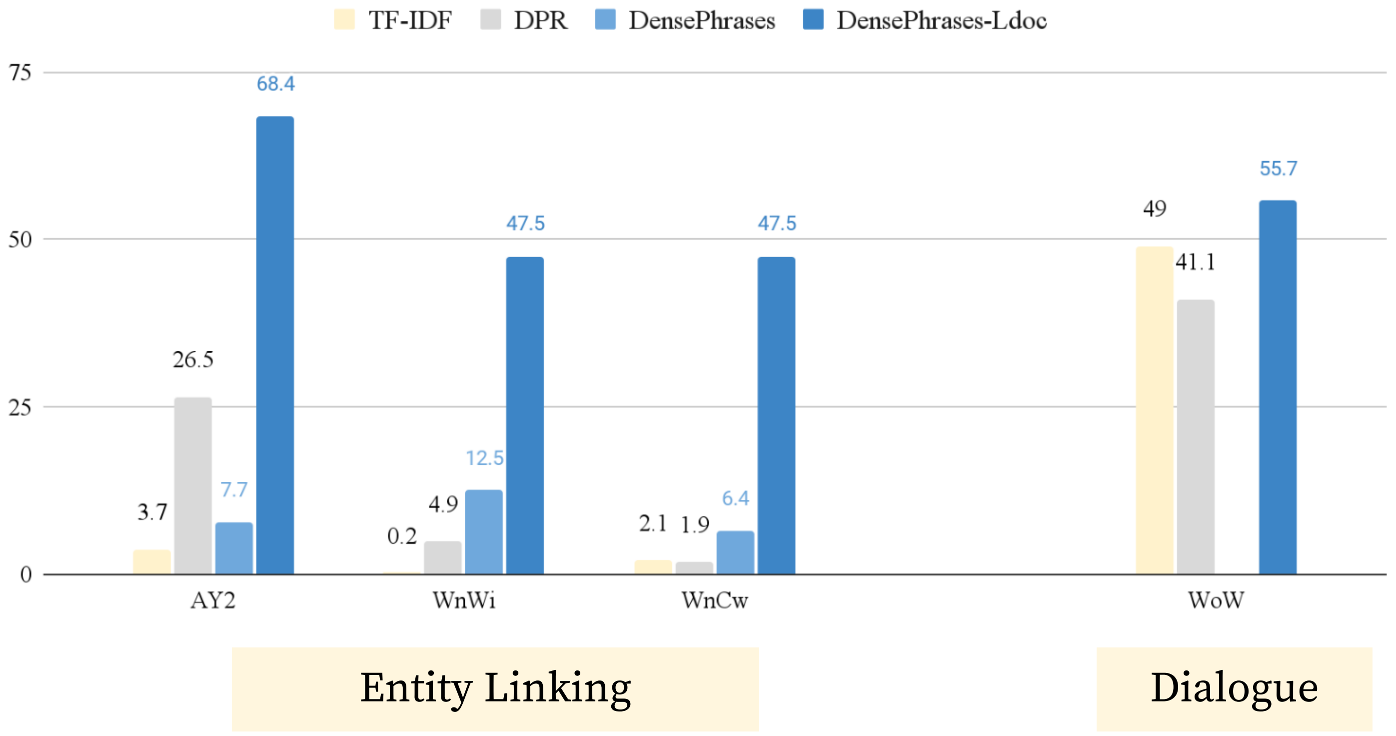


topically relevant!
(annotated)

topically relevant!
(annotated)

Maximize the marginal probability of
any phrases in the relevant document

04 Retrieval for **Entity Linking & Dialogue**



DensePhrases can be adapted to **retrieve topically relevant documents**!

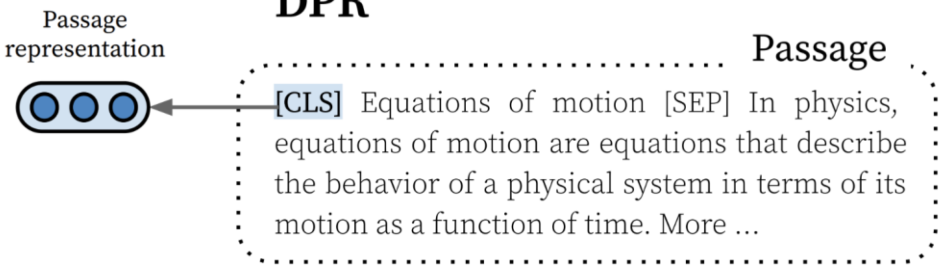
05

Complexity Analysis

05 Problem of **Multi-vector** Encoding

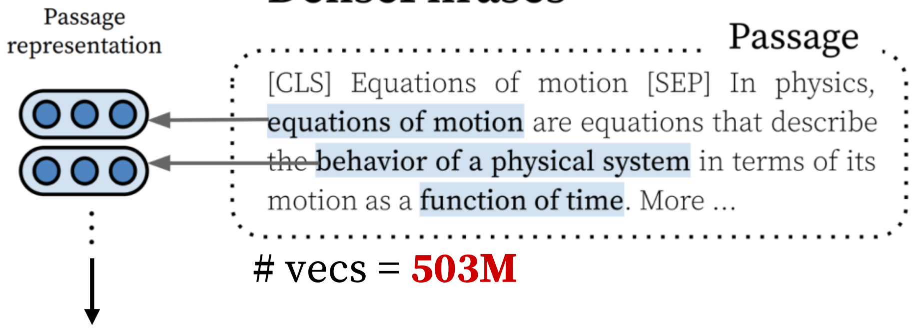
Luan et al., 2021; Khattab and Zaharia, 2020

DPR



vecs = **21M**

DensePhrases



vecs = **503M**

More vectors, **more space**!

Phrase indexes are **heavy**!

1.2TB (Seo et al., 2019) **1.5TB** (Lee et al., 2020) **320GB** (Lee et al., 2021)

05 **Reducing** the Size of Phrase Index

“The New York metropolitan area is
home to many prestigious
institutions of higher education.”

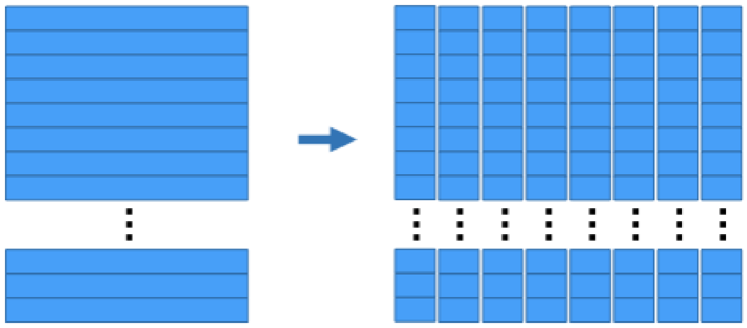
Phrase Filter
(Lee et al., 2021)

“The New York metropolitan area”

“prestigious institutions”

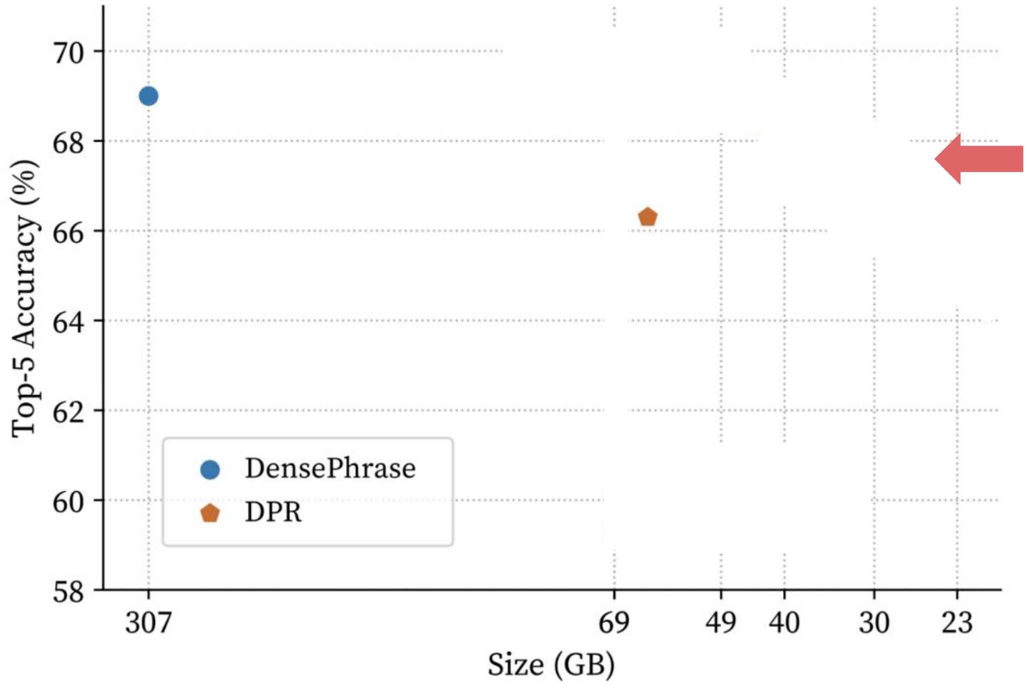
“higher education”

…



Optimized Product Quantization
(Ge et al., 2013)
+
Query-side Fine-tuning
(Lee et al., 2021)
=
Quantization-aware Fine-tuning

05 **Reducing** the Size of Phrase Index



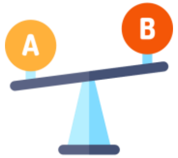
We can safely reduce the size down to **23GB**! (DPR = 69GB)

DensePhrases with **# vector/passage = 8.8** is similar to DPR.

06

Conclusion

06 Conclusion



Q1: Is this **better** than passage retrievers?

Yes! **DensePhrases > DPR** on passage retrieval and open-domain QA!

Q2: **Why** does this work?

Better at **fine-grained entailment**, can be used for coarse retrieval.

Q3: How **efficient** is this?

Can safely reduce the index size from **307GB to 23GB**!

Paper:	https://arxiv.org/abs/2109.08133
Code & Models:	https://github.com/princeton-nlp/DensePhrases
Demo:	http://densephrases.korea.ac.kr/
E-mail:	jinhyuklee@cs.princeton.edu

감사합니다

Phrase Retrieval Learns Passage Retrieval, Too