Information Retrieval

Sept 30, 2025 @ Introduction to Human Language Technology

Eugene Yang eugene.yang@jhu.edu

Slides borrowed from SIGIR24 Tutorial
“Neural Methods for Cross-Language Information Retrieval”

What is Information Retrieval?

(relevant)

Retrieve information from a storage based on user's information need

Don’t we have Google?

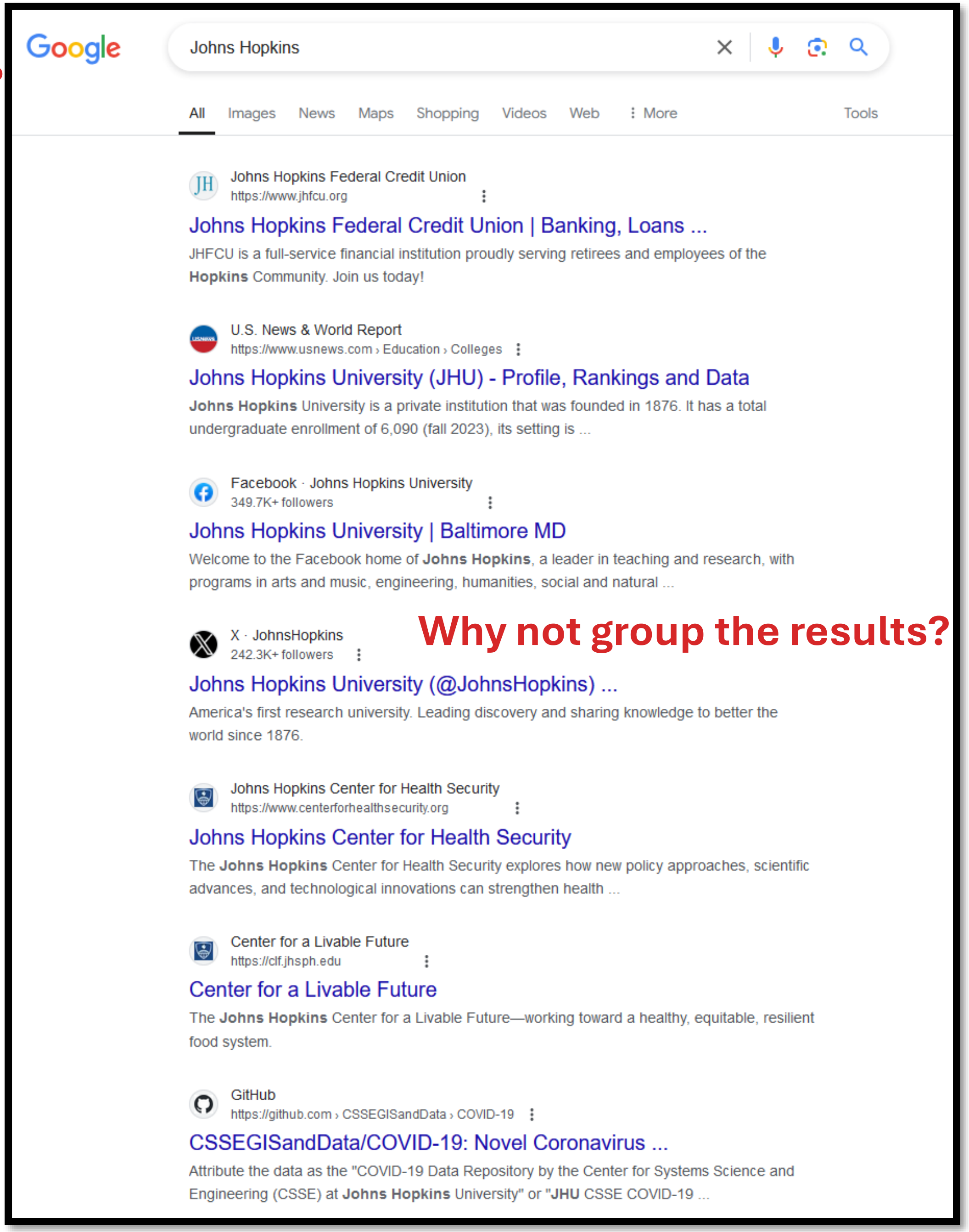
Yes, **but Google is not all.**

What if I’m looking for the person?

Why not read my mind?

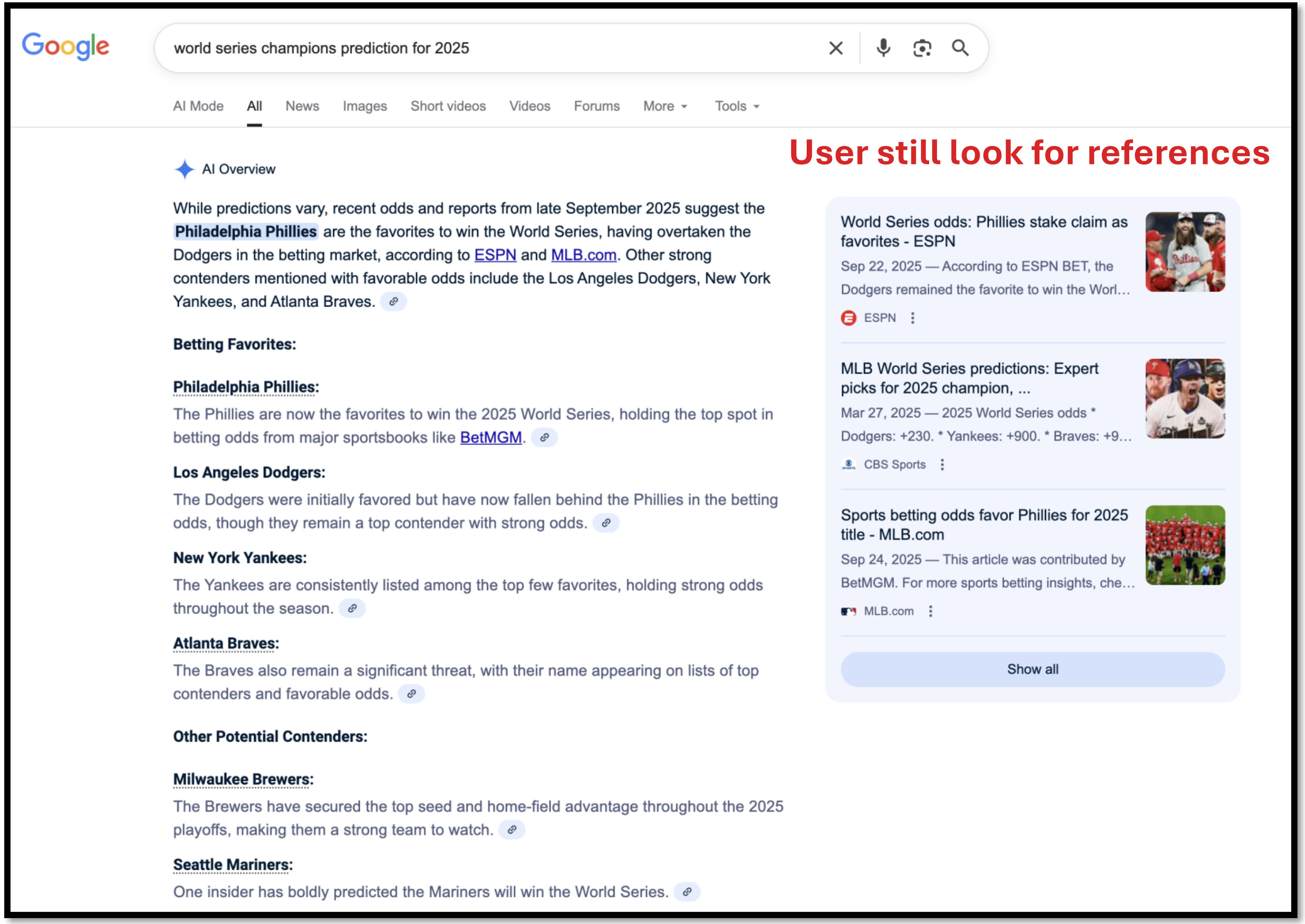
Why in a list?

Why asking me to read?



Why not group the results?

Still need to understand the intent (it’s about MLB)



User still look for references

Still need retrieval
for source information

Google Search is just one implementation

Google trained us well!

- Even faster?
- Smarter?
- Cross language?

Hard Matching Problem

- Text to text
  - Search in notes
  - Cross language search
  - Cross domain search
- Text to other modalities
  - Image search
  - Video search

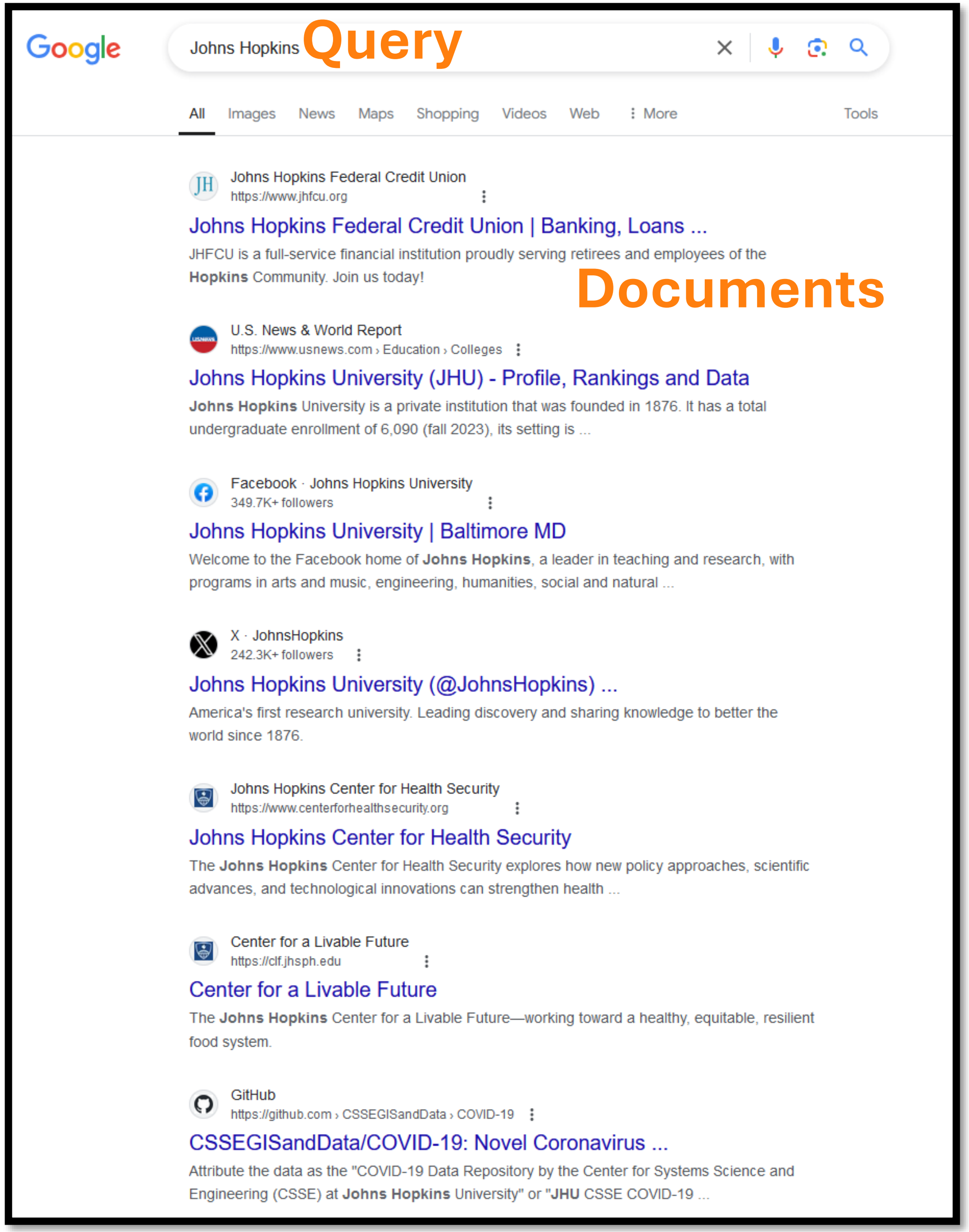
Different Search Process

- Iterative search
  - e.g., electronic discovery and systematic review
- Conversational search
  - Alexa search
- Recommendation systems
  - Implicit queries

Core Problem

- Rank relevant document at top
- Do it fast

Ranked List



Design Space

Effectiveness

- Definition of relevancy
- How to model relevancy

Downstream Customer

- Human or LLM?
- What kind of presentation?

Efficiency

- How fast
- Fast at what stage

Agenda

- What is information retrieval?
- Retrieval modeling and pipeline
  - statistical and neural
- Evaluation
- Retrieval-Augmented Generation
- Research problems

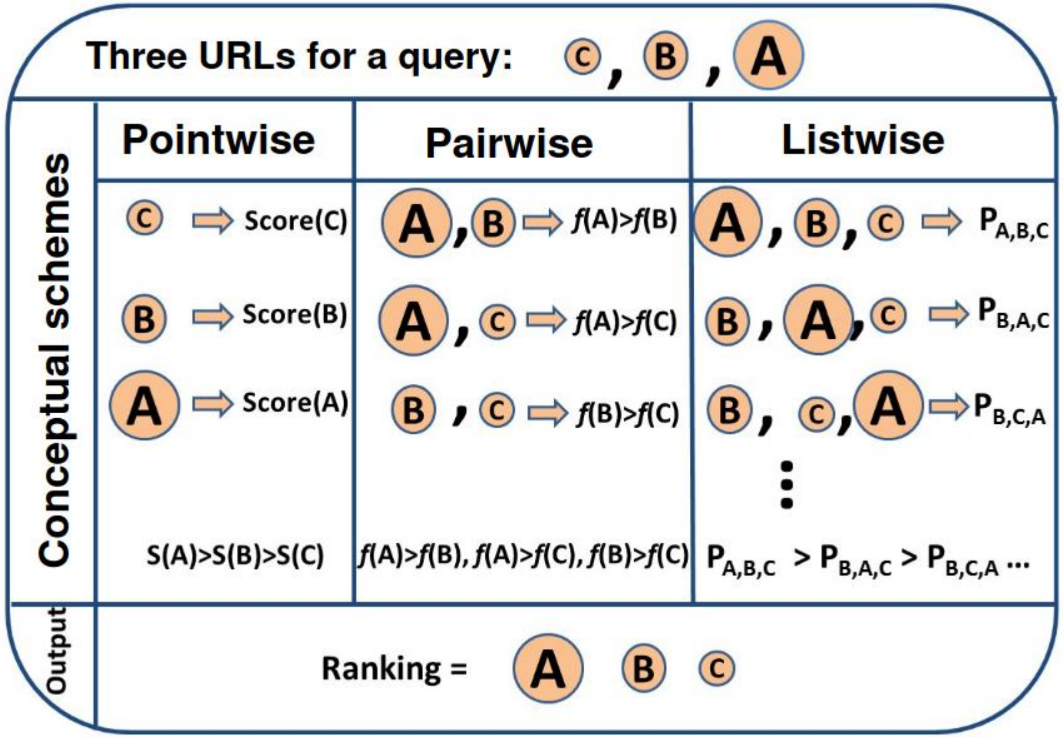
Retrieval Modeling and Pipeline

Modeling relevancy and do it fast

Three main modeling strategies

- **Pointwise**
- Pairwise
- Listwise

- And combinations of them



https://medium.com/vptech/learning-to-rank-at-veepee-ed420fd828e5

Statistical Models

$$\text{score}(D, Q) = \underbrace{\sum}_{\text{For each query term}} \boxed{\text{How important the term is}} \times \boxed{\text{How often the term appear in the D}}$$

$$\text{score}(D, Q) = \underbrace{\sum}_{\text{For each query term}} \boxed{\text{Inverted document frequency}} \times \boxed{\text{Term frequency}}$$

TF-IDF

$$\text{score}(D, Q) = \sum_{i=1}^{n} log \frac{N}{n_t} \times \log(f(q_i, D) + 1)$$

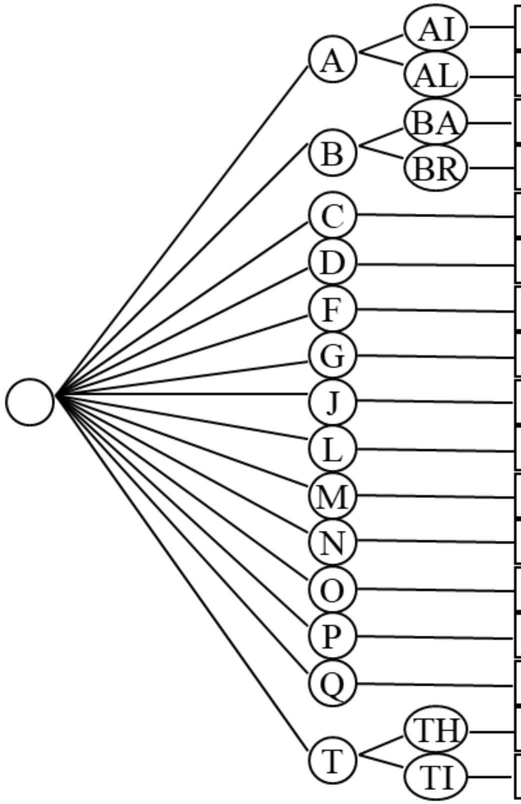
BM25

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

How to make it fast?

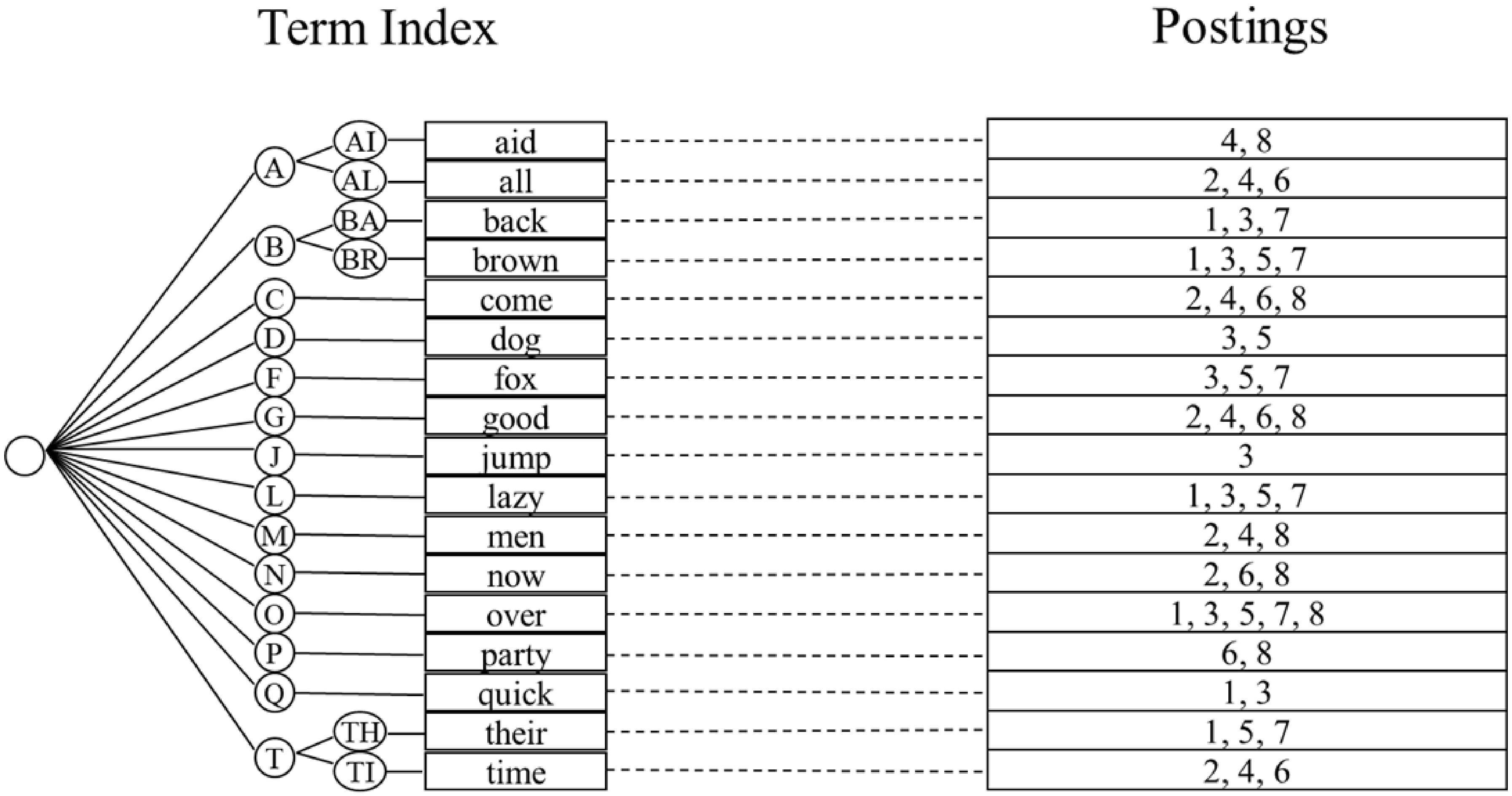
- “Fast” in responding to queries
- Better data structure
- Preprocess the data

Inverted Index

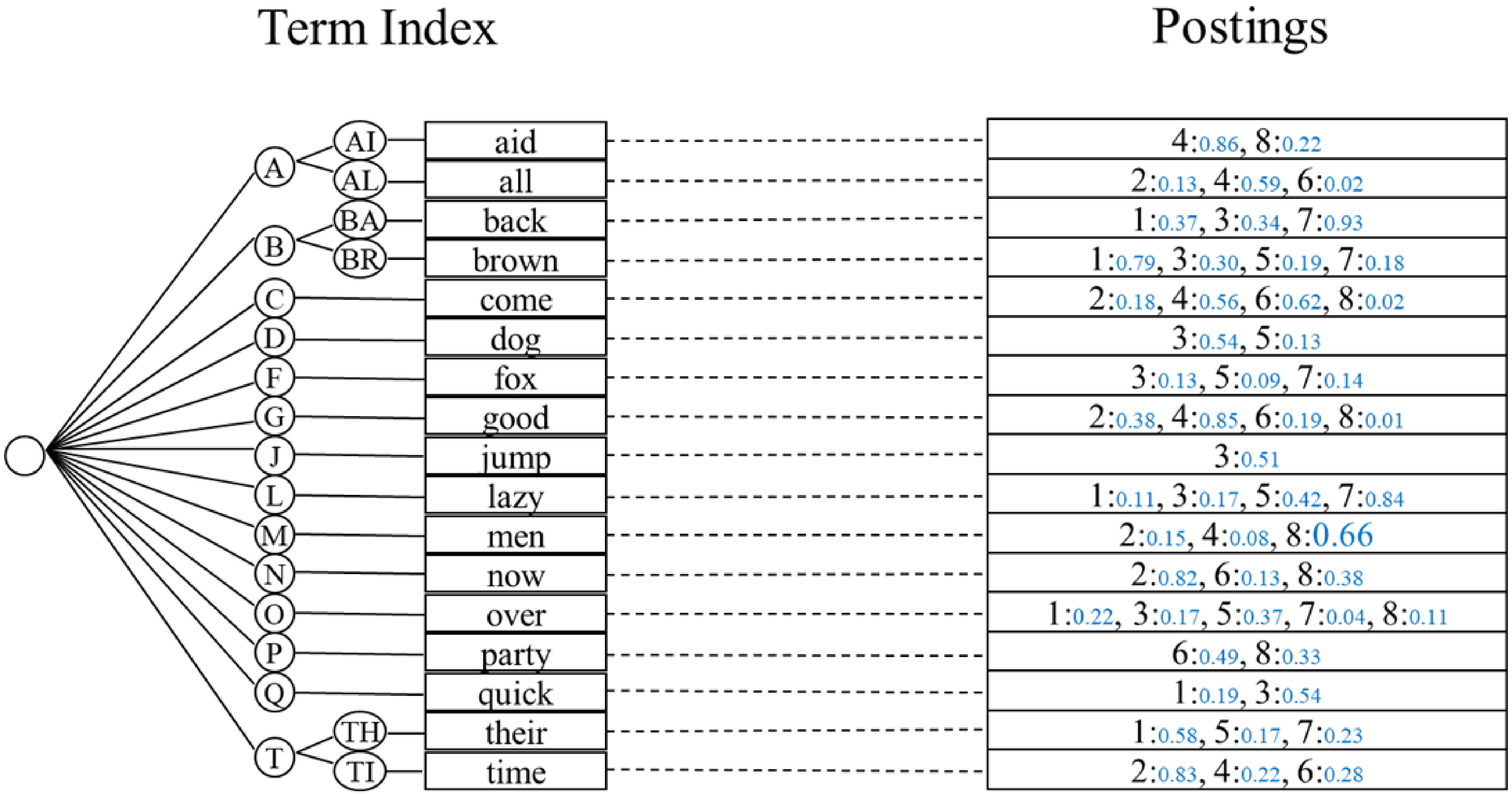


Term Index	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8	Postings
aid				1				1	4, 8
all		1		1		1			2, 4, 6
back	1		1				1		1, 3, 7
brown	1		1		1		1		1, 3, 5, 7
come		1		1		1		1	2, 4, 6, 8
dog			1		1				3, 5
fox			1		1		1		3, 5, 7
good		1		1		1		1	2, 4, 6, 8
jump			1						3
lazy	1		1		1		1		1, 3, 5, 7
men		1		1				1	2, 4, 8
now		1				1		1	2, 6, 8
over	1		1		1		1	1	1, 3, 5, 7, 8
party						1		1	6, 8
quick	1		1						1, 3
their	1				1		1		1, 5, 7
time		1		1		1			2, 4, 6

Inverted Index

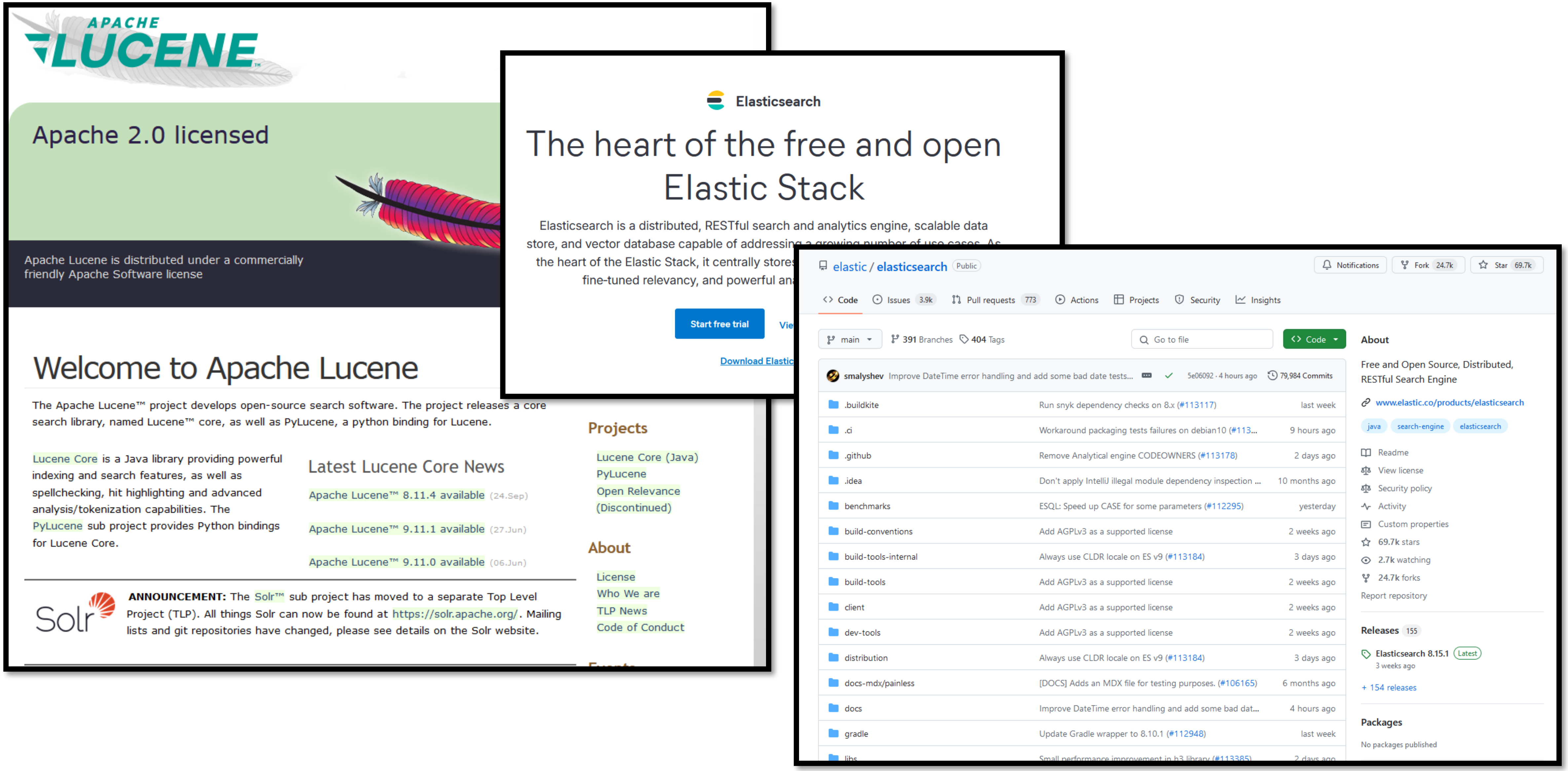


Inverted Index



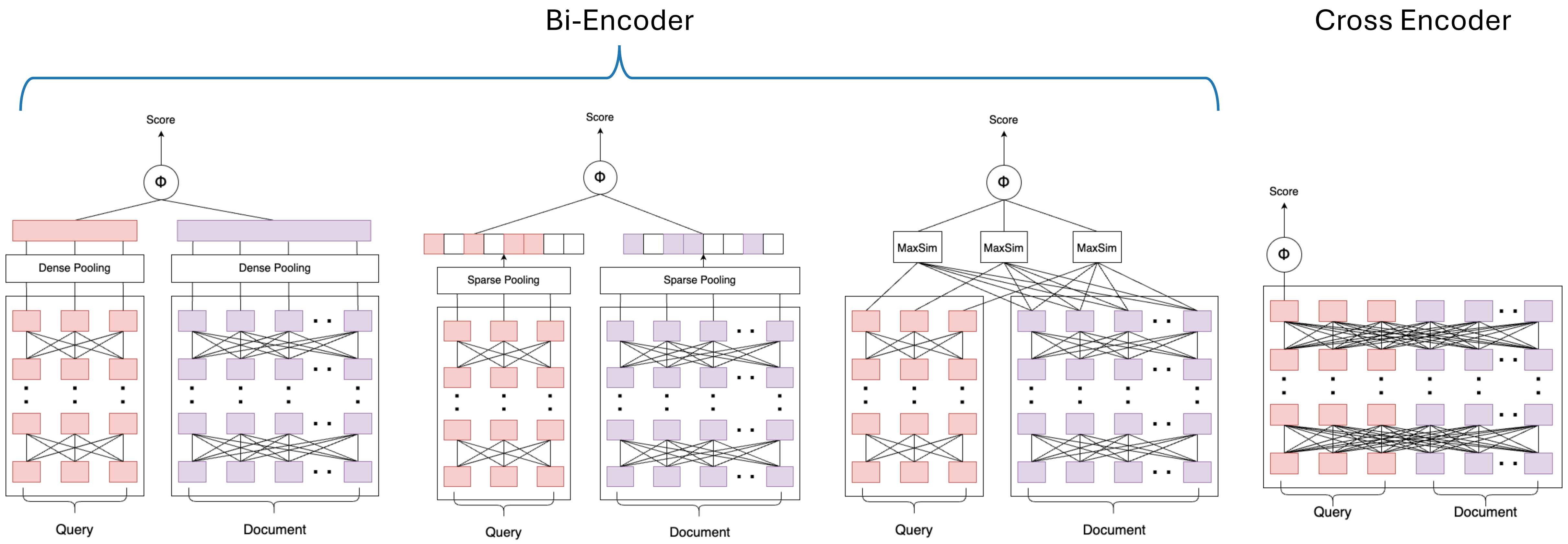
Two-Stage System

- Offline preprocessing and indexing
  - Define retrieval unit
  - Tokenization
  - Build the inverted index
- Online query serving
  - Traverse the inverted index and score it


APACHE LUCENE
Apache 2.0 licensed
Apache Lucene is distributed under a commercially friendly Apache Software license
Welcome to Apache Lucene
The Apache Lucene™ project develops open-source search software. The project releases a core search library, named Lucene™ core, as well as PyLucene, a python binding for Lucene.
Lucene Core is a Java library providing powerful indexing and search features, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities. The PyLucene sub project provides Python bindings for Lucene Core.
Latest Lucene Core News
Apache Lucene™ 8.11.4 available (24.Sep)
Apache Lucene™ 9.11.1 available (27.Jun)
Apache Lucene™ 9.11.0 available (06.Jun)
Solr
ANNOUNCEMENT: The Solr™ sub project has moved to a separate Top Level Project (TLP). All things Solr can now be found at https://solr.apache.org/. Mailing lists and git repositories have changed, please see details on the Solr website.
Projects
Lucene Core (Java)
PyLucene
Open Relevance (Discontinued)
About
License
Who We are
TLP News
Code of Conduct
Elasticsearch
The heart of the free and open Elastic Stack
Elasticsearch is a distributed, RESTful search and analytics engine, scalable data
Start free trial
elastic / elasticsearch Public
Notifications
Fork 24.7k
Star 69.7k
Code
Issues 3.9k
Pull requests 773
Actions
Projects
Security
Insights
main
391 Branches
404 Tags
Go to file
Code
smalyshev Improve DateTime error handling and add some bad date tests...
5e06092 · 4 hours ago
79,984 Commits
.buildkite
Run snyk dependency checks on 8.x (#113117)
last week
.ci
Workaround packaging tests failures on debian10 (#113...
9 hours ago
.github
Remove Analytical engine CODEOWNERS (#113178)
2 days ago
.idea
Don't apply IntelliJ illegal module dependency inspection ...
10 months ago
benchmarks
ESQL: Speed up CASE for some parameters (#112295)
yesterday
build-conventions
Add AGPLv3 as a supported license
2 weeks ago
build-tools-internal
Always use CLDR locale on ES v9 (#113184)
3 days ago
build-tools
Add AGPLv3 as a supported license
2 weeks ago
client
Add AGPLv3 as a supported license
2 weeks ago
dev-tools
Add AGPLv3 as a supported license
2 weeks ago
distribution
Always use CLDR locale on ES v9 (#113184)
3 days ago
docs-mdx/painless
[DOCS] Adds an MDX file for testing purposes. (#106165)
6 months ago
docs
Improve DateTime error handling and add some bad dat...
4 hours ago
gradle
Update Gradle wrapper to 8.10.1 (#112948)
last week
About
Free and Open Source, Distributed, RESTful Search Engine
www.elastic.co/products/elasticsearch
java
search-engine
elasticsearch
Readme
View license
Security policy
Activity
Custom properties
69.7k stars
2.7k watching
24.7k forks
Report repository
Releases 155
Elasticsearch 8.15.1 Latest
3 weeks ago
+ 154 releases
Packages
No packages published

Can we go beyond surface forms?

neural language models

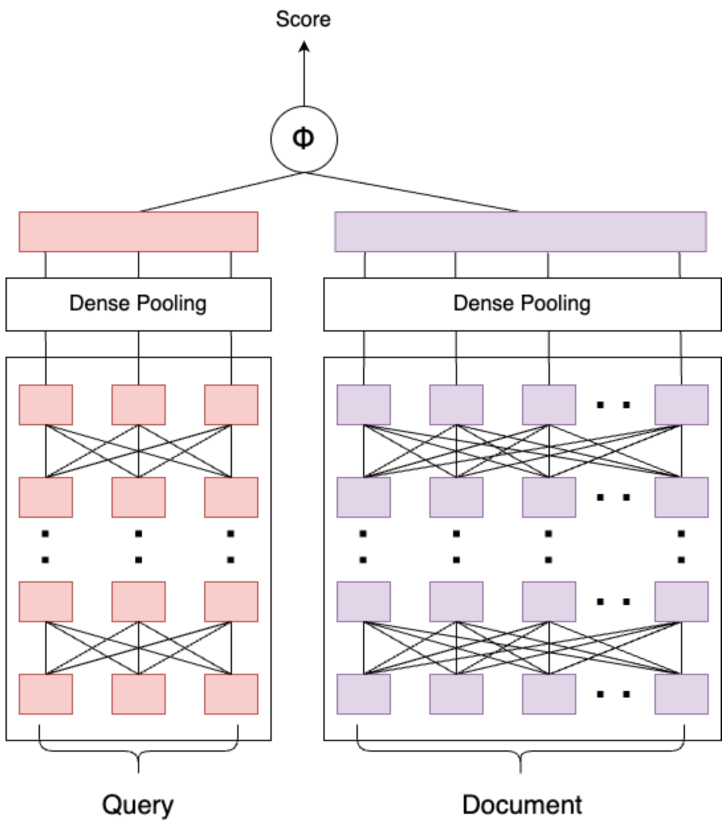


One Dense Vector
Per Sequence
e.g., DPR

One **Sparse** Vector
Per Sequence
e.g., SPLADE

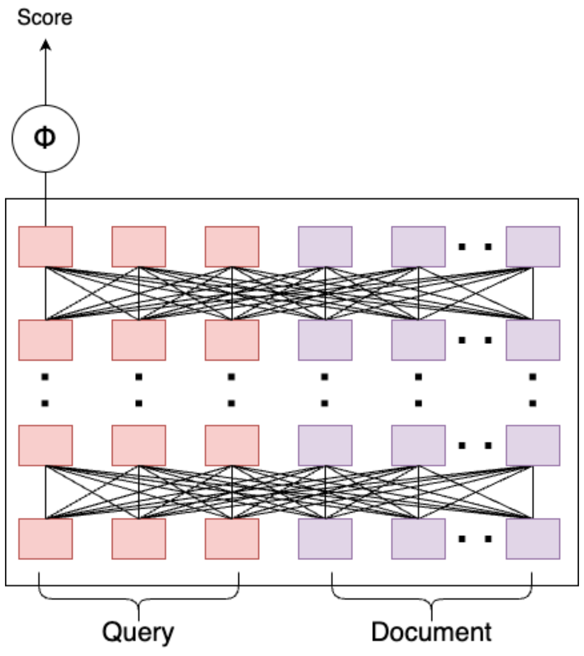
Multiple Dense Vectors
Per Sequence
e.g., ColBERT

Joint Encoder
e.g., monoBERT



Online

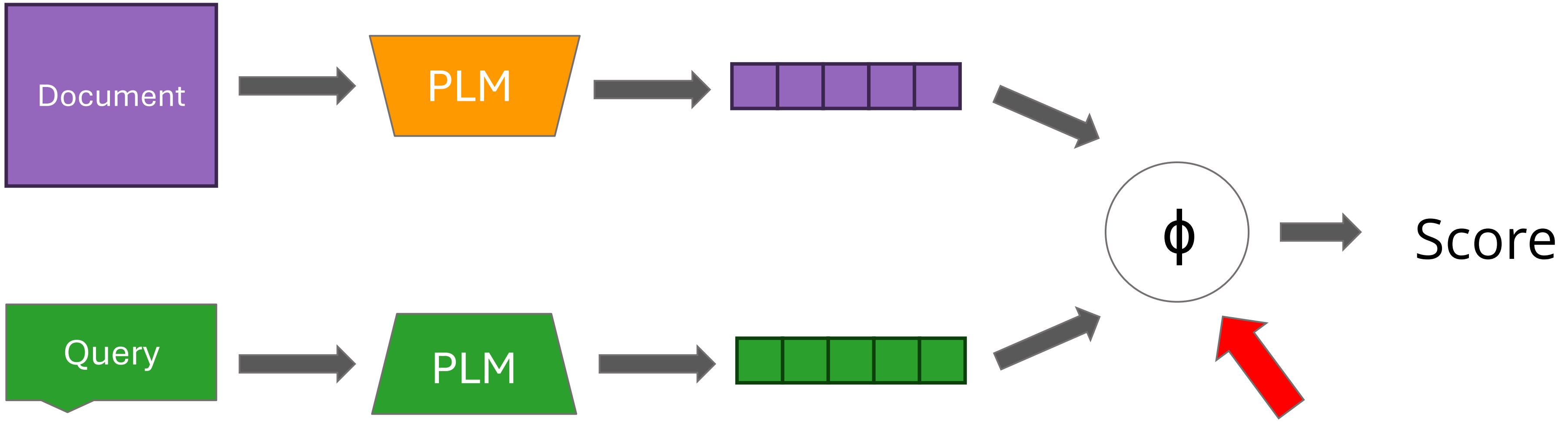
Offline

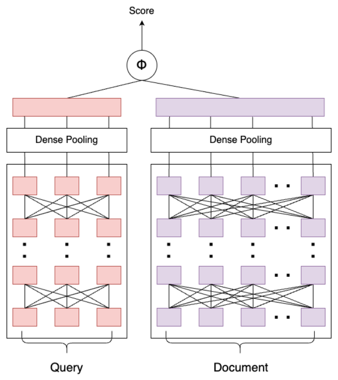


Both Online

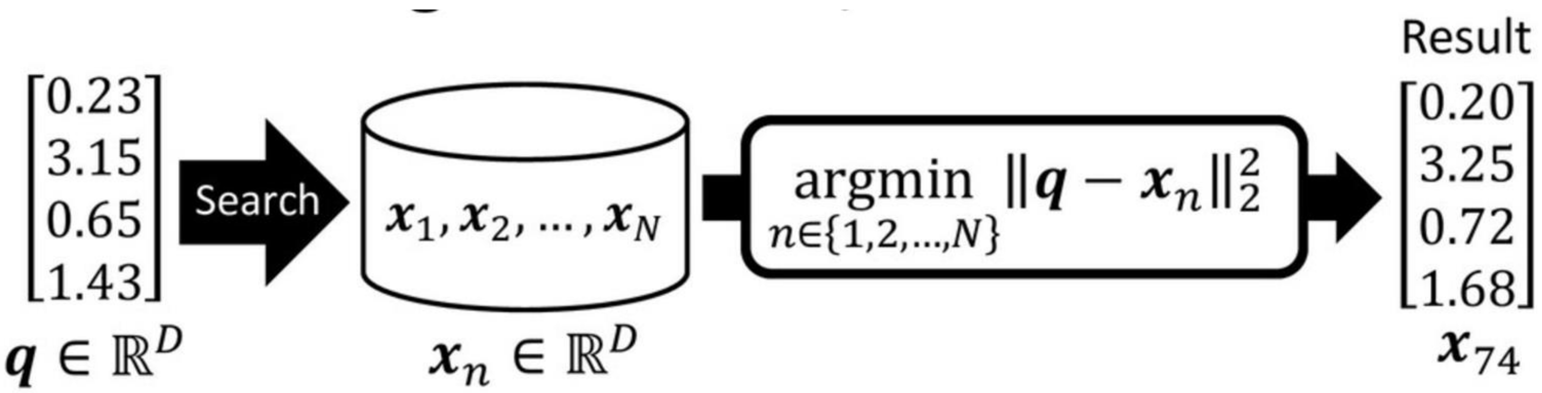
Separate query and document processing

One Vector per Query, One Vector per Document



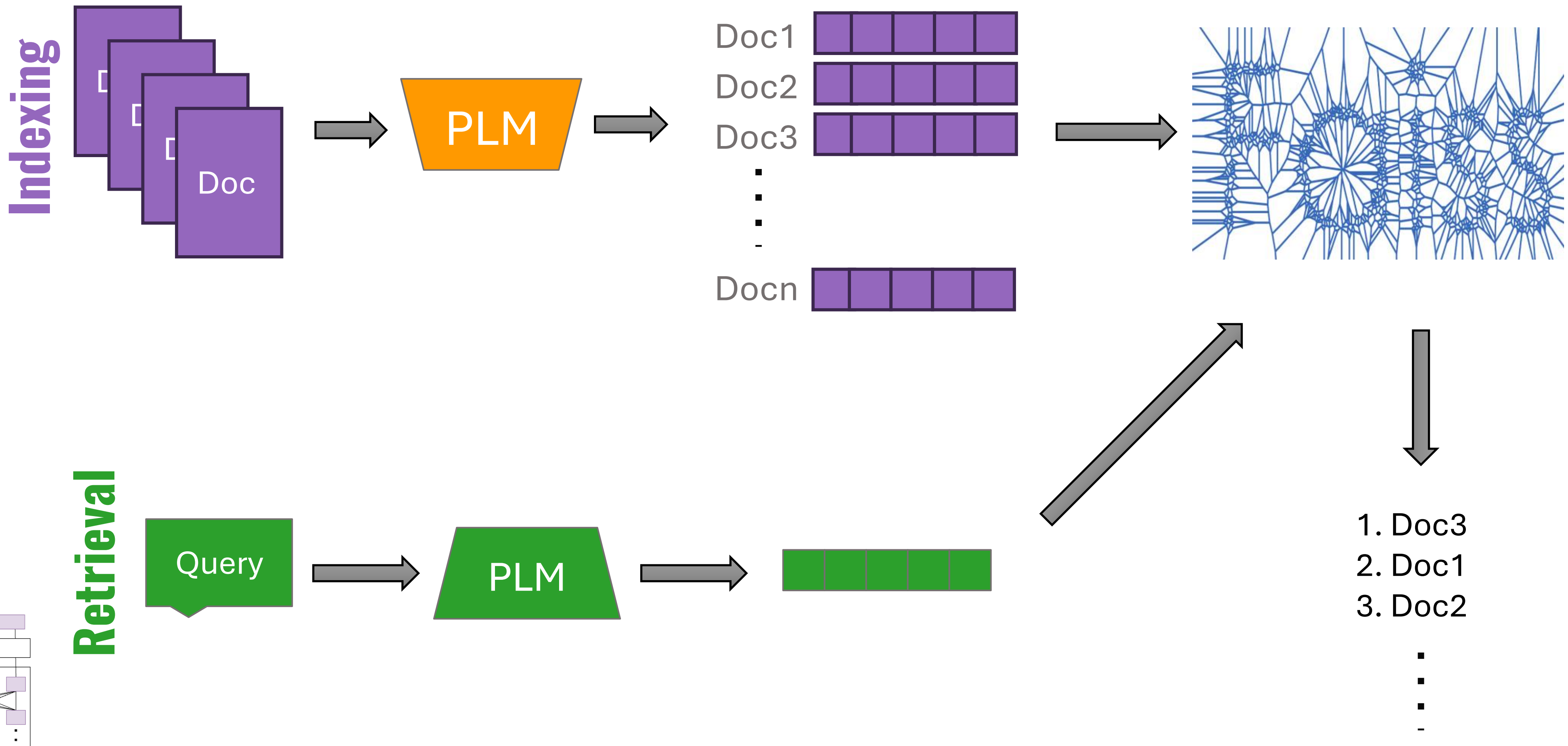


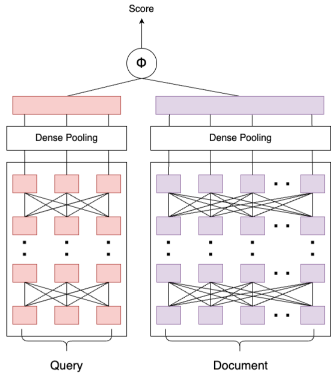
Nearest Vectors aka Neighbors

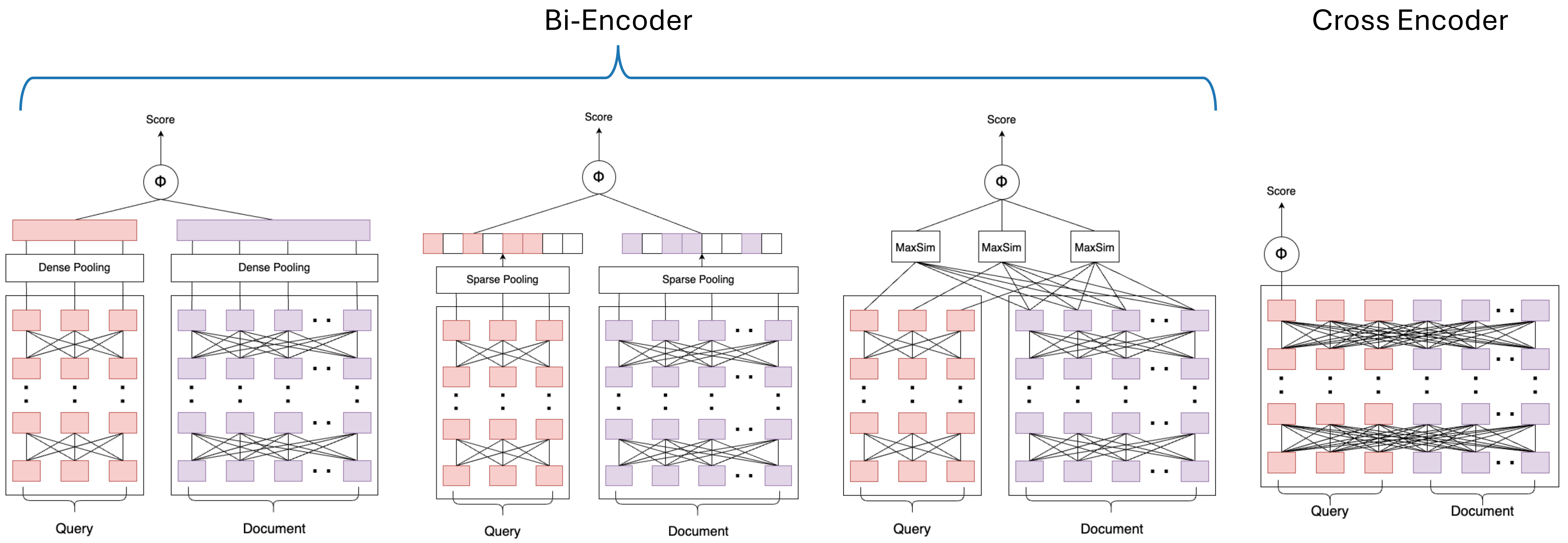


- Linear Search
    - Slow (scales linearly in size of document collection)
- Approximate Methods (e.g., Product Quantization) → **ANN**
    - Faster Search
- Runtime Efficiency vs Effectiveness

DPR Indexing and Retrieval







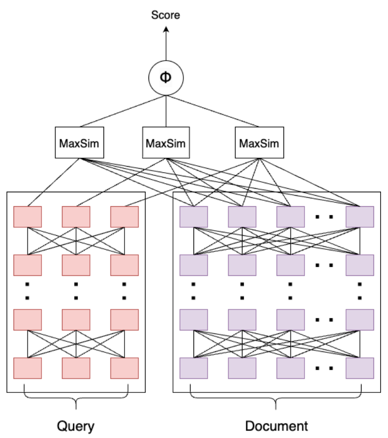
One Dense Vector
Per Sequence
e.g., DPR

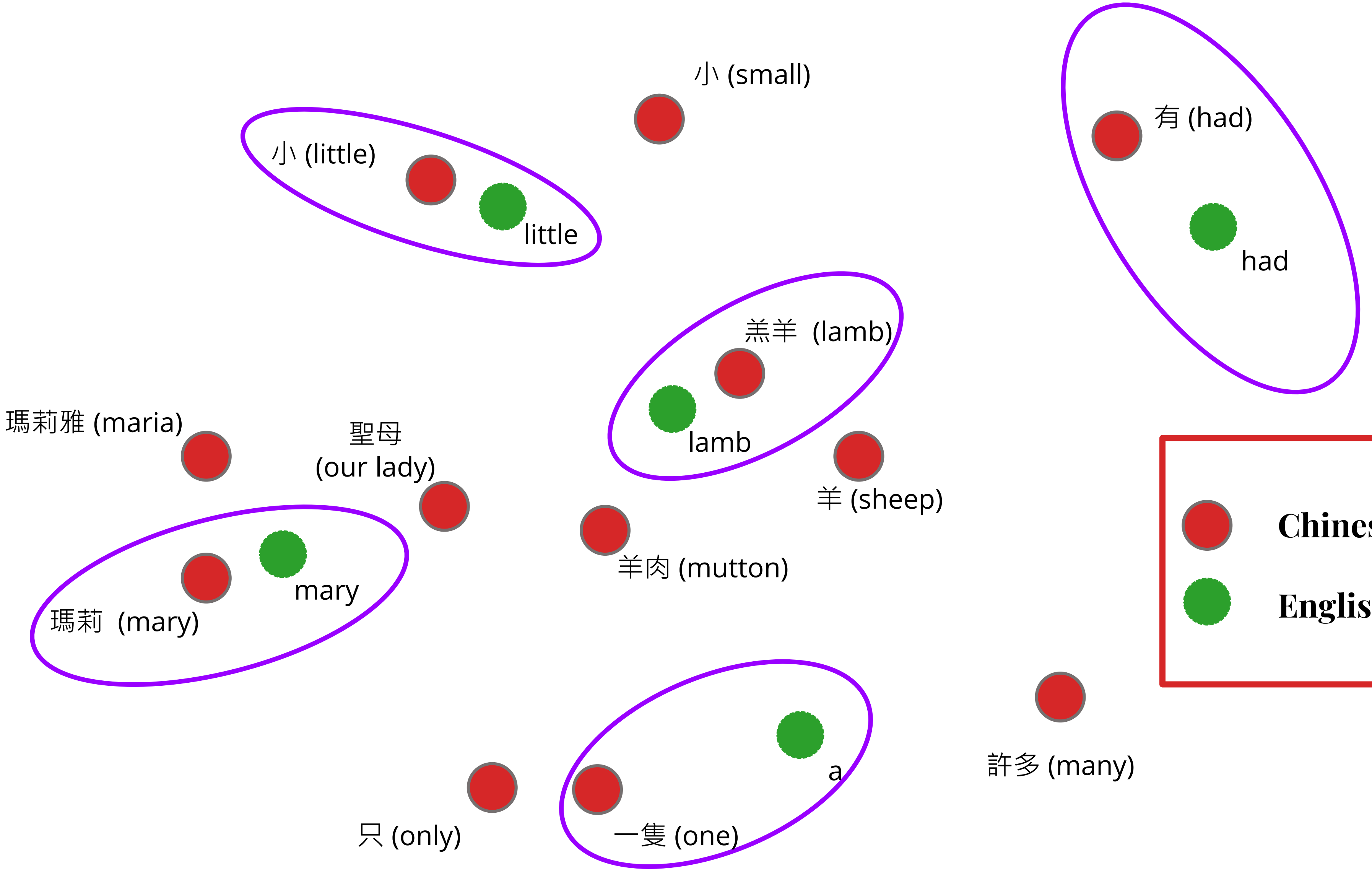
One **Sparse** Vector
Per Sequence
e.g., SPLADE

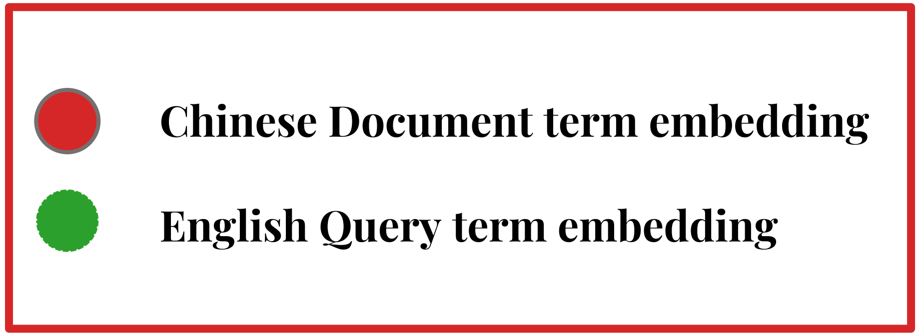
Multiple Dense Vectors
Per Sequence
e.g., ColBERT

Joint Encoder
e.g., monoBERT

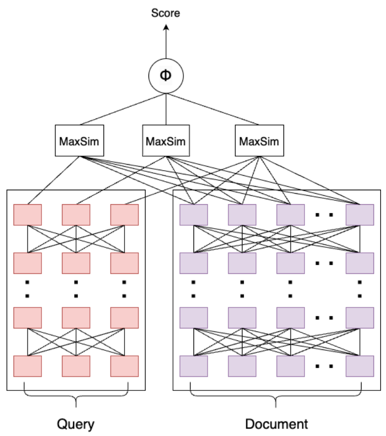
One Vector per Term: MaxSim

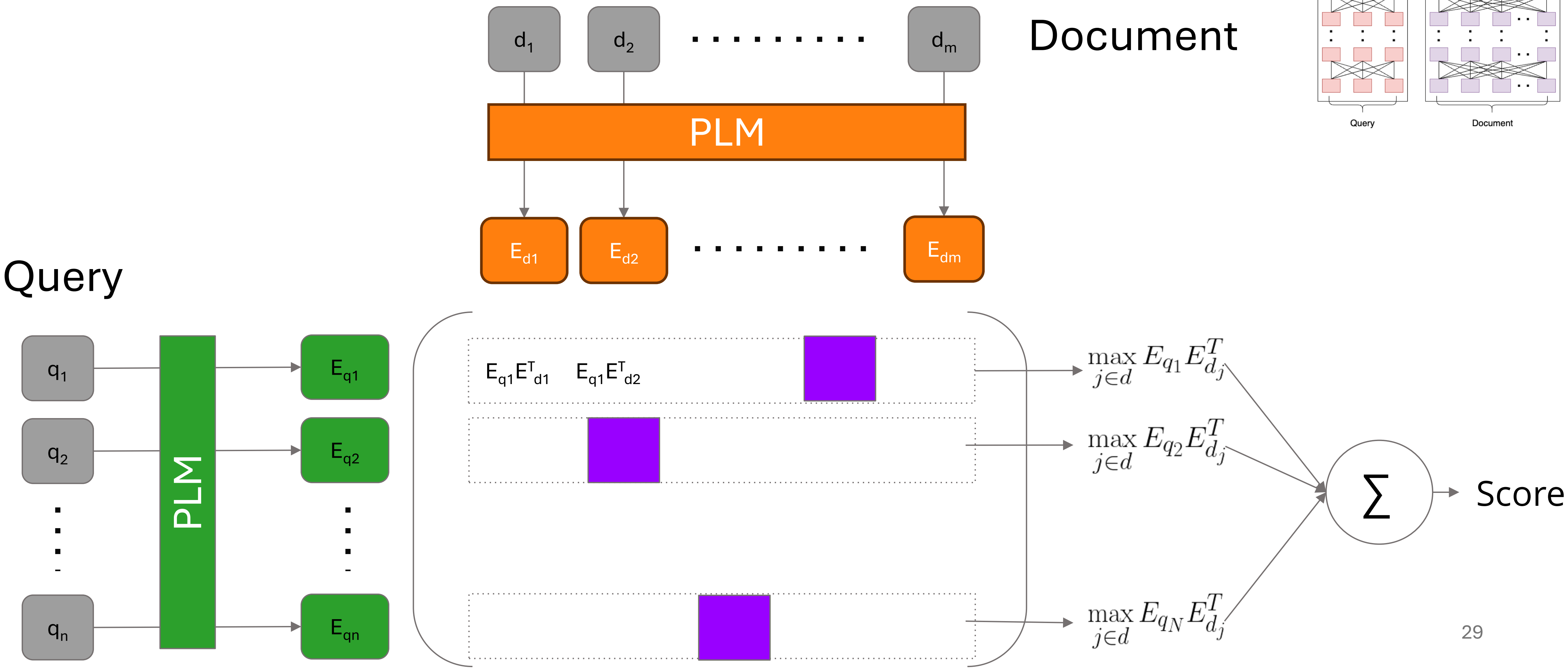




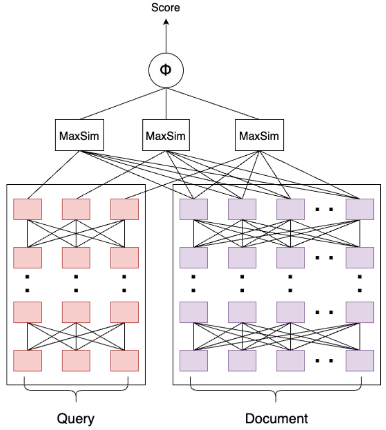


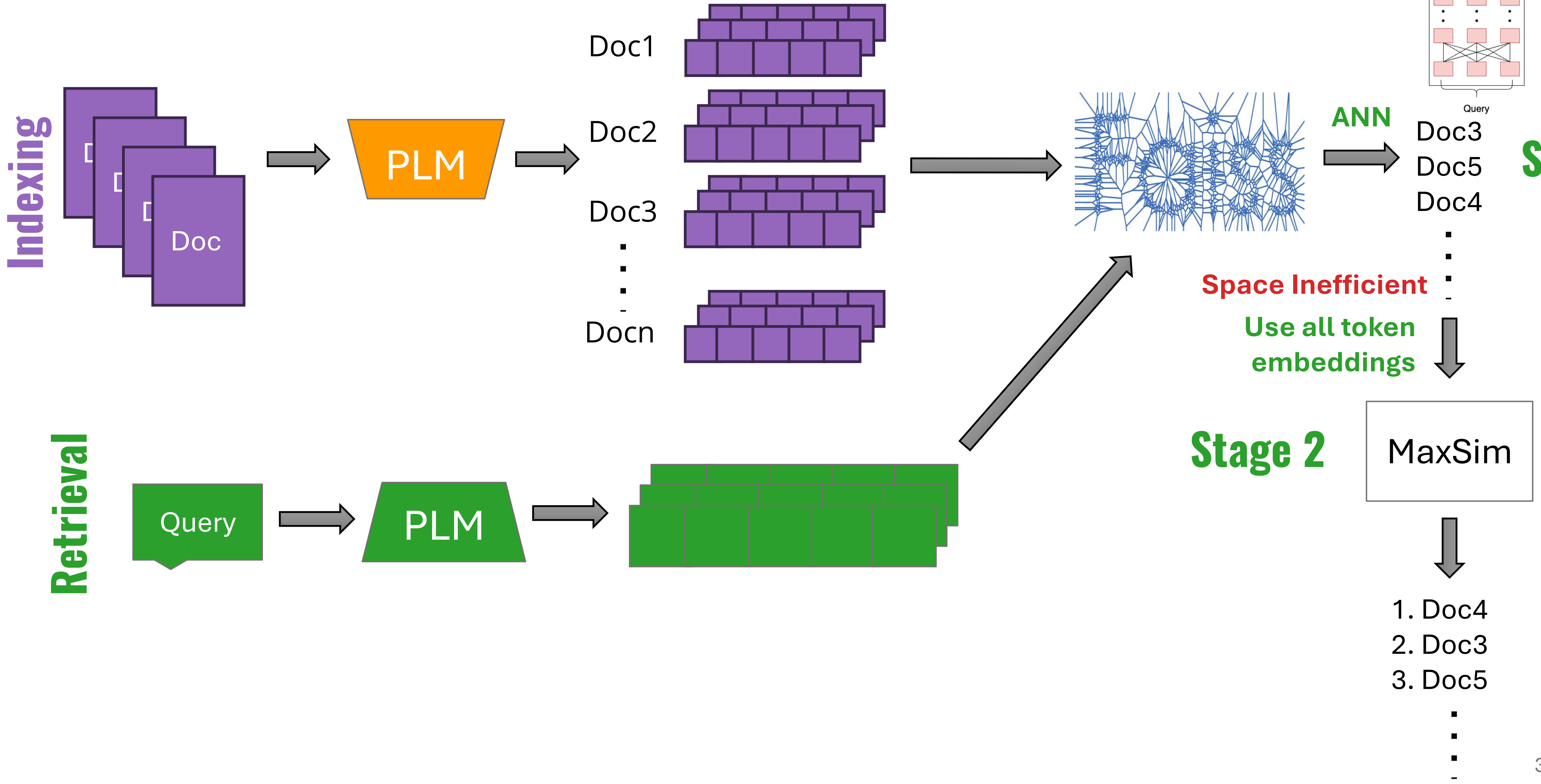
MaxSim in Action -- ColBERT

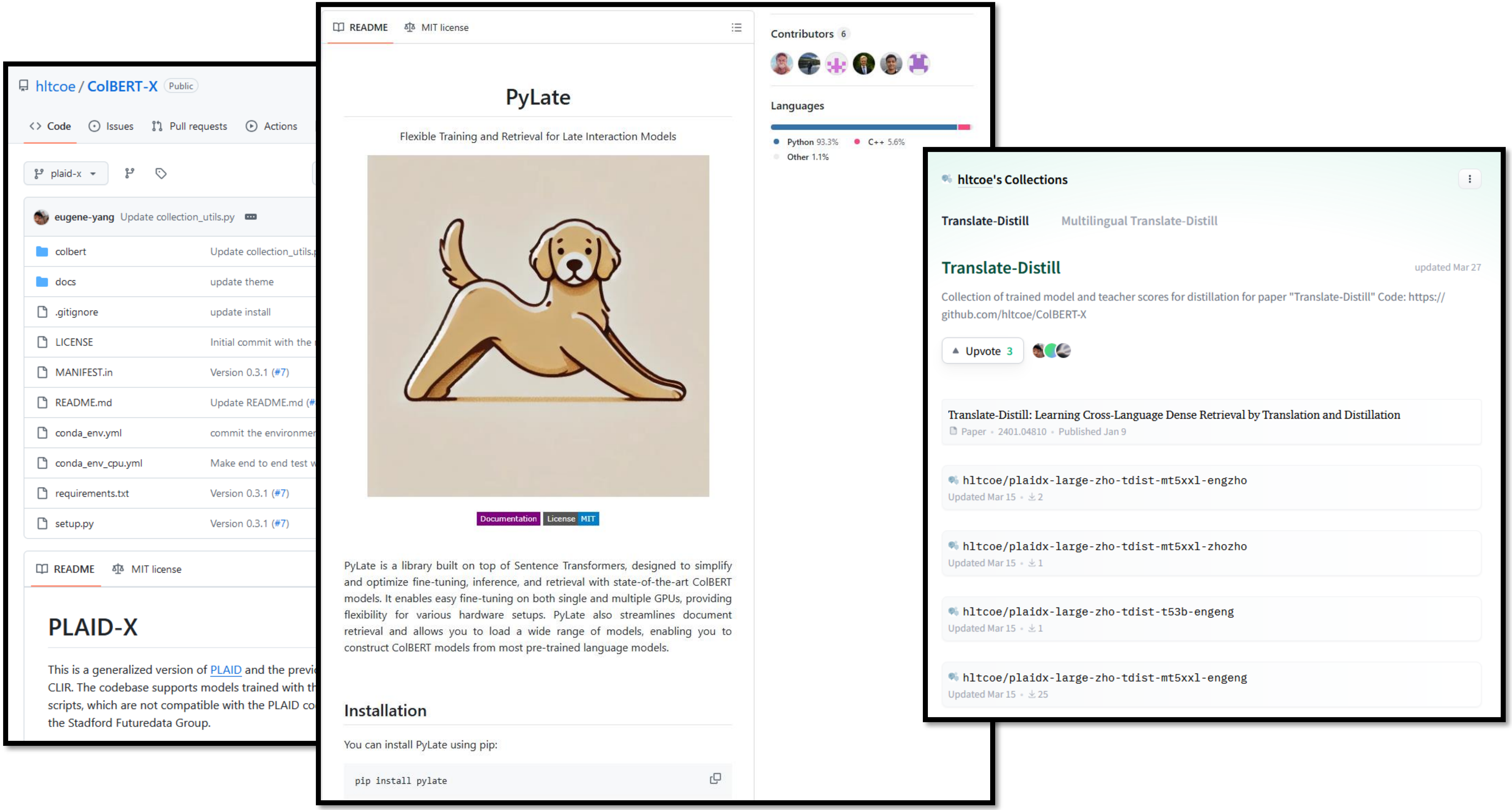




One Vector per term: Multi-stage Retrieval

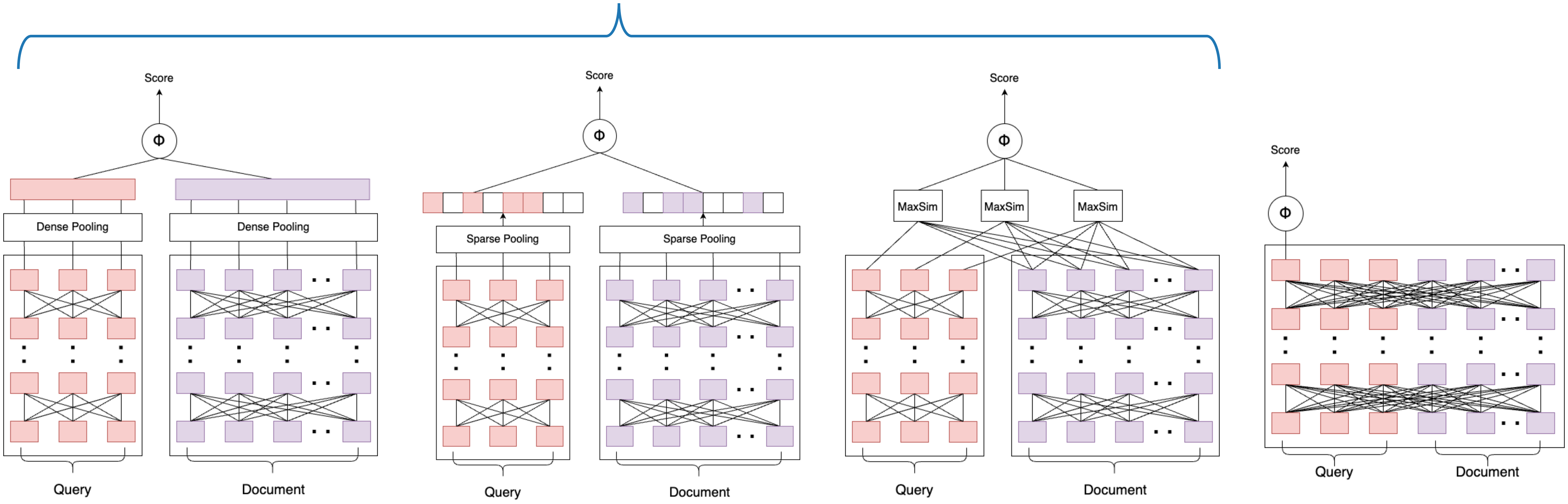





hltcoe / ColBERT-X Public
Code
Issues
Pull requests
Actions
plaid-x
eugene-yang Update collection_utils.py
colbert Update collection_utils.
docs update theme
.gitignore update install
LICENSE Initial commit with the
MANIFEST.in Version 0.3.1 (#7)
README.md Update README.md (#
conda_env.yml commit the environmer
conda_env_cpu.yml Make end to end test w
requirements.txt Version 0.3.1 (#7)
setup.py Version 0.3.1 (#7)
README
MIT license
PLAID-X
This is a generalized version of PLAID and the previ
CLIR. The codebase supports models trained with th
scripts, which are not compatible with the PLAID co
the Stadford Futuredata Group.
README
MIT license
PyLate
Flexible Training and Retrieval for Late Interaction Models
Documentation
License MIT
PyLate is a library built on top of Sentence Transformers, designed to simplify and optimize fine-tuning, inference, and retrieval with state-of-the-art ColBERT models. It enables easy fine-tuning on both single and multiple GPUs, providing flexibility for various hardware setups. PyLate also streamlines document retrieval and allows you to load a wide range of models, enabling you to construct ColBERT models from most pre-trained language models.
Installation
You can install PyLate using pip:
pip install pylate
Contributors 6
Languages
Python 93.3%
C++ 5.6%
Other 1.1%
hltcoe's Collections
Translate-Distill
Multilingual Translate-Distill
Translate-Distill
updated Mar 27
Collection of trained model and teacher scores for distillation for paper "Translate-Distill" Code: https://github.com/hltcoe/ColBERT-X
Upvote 3
Translate-Distill: Learning Cross-Language Dense Retrieval by Translation and Distillation
Paper • 2401.04810 • Published Jan 9
hltcoe/plaidx-large-zho-tdist-mt5xxl-engzho
Updated Mar 15 • 2
hltcoe/plaidx-large-zho-tdist-mt5xxl-zhozho
Updated Mar 15 • 1
hltcoe/plaidx-large-zho-tdist-t53b-engeng
Updated Mar 15 • 1
hltcoe/plaidx-large-zho-tdist-mt5xxl-engeng
Updated Mar 15 • 25

Bi-Encoder

Cross Encoder



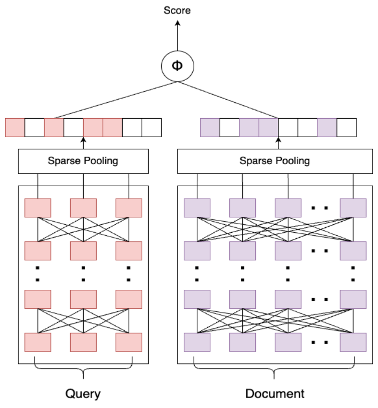
One Dense Vector
Per Sequence
e.g., DPR

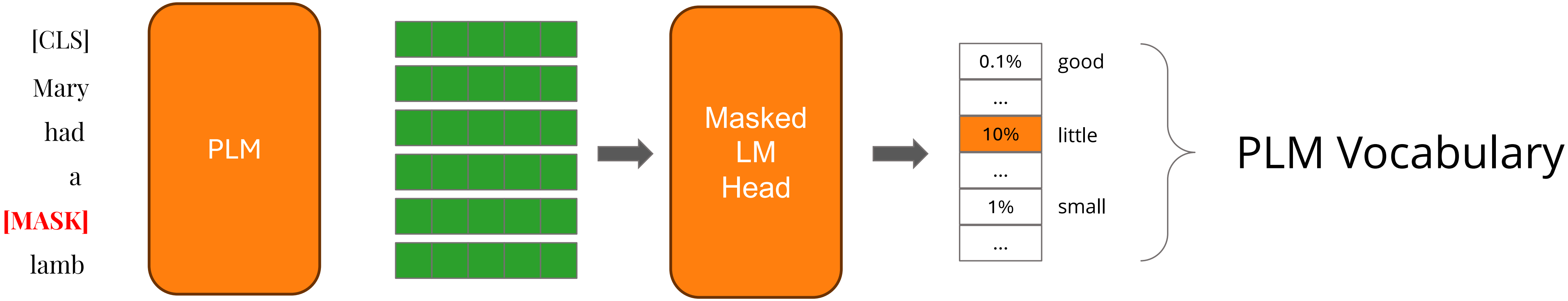
One **Sparse** Vector
Per Sequence
e.g., SPLADE

Multiple Dense Vectors
Per Sequence
e.g., ColBERT

Joint Encoder
e.g., monoBERT

High-dimensional Vector: Masked LM





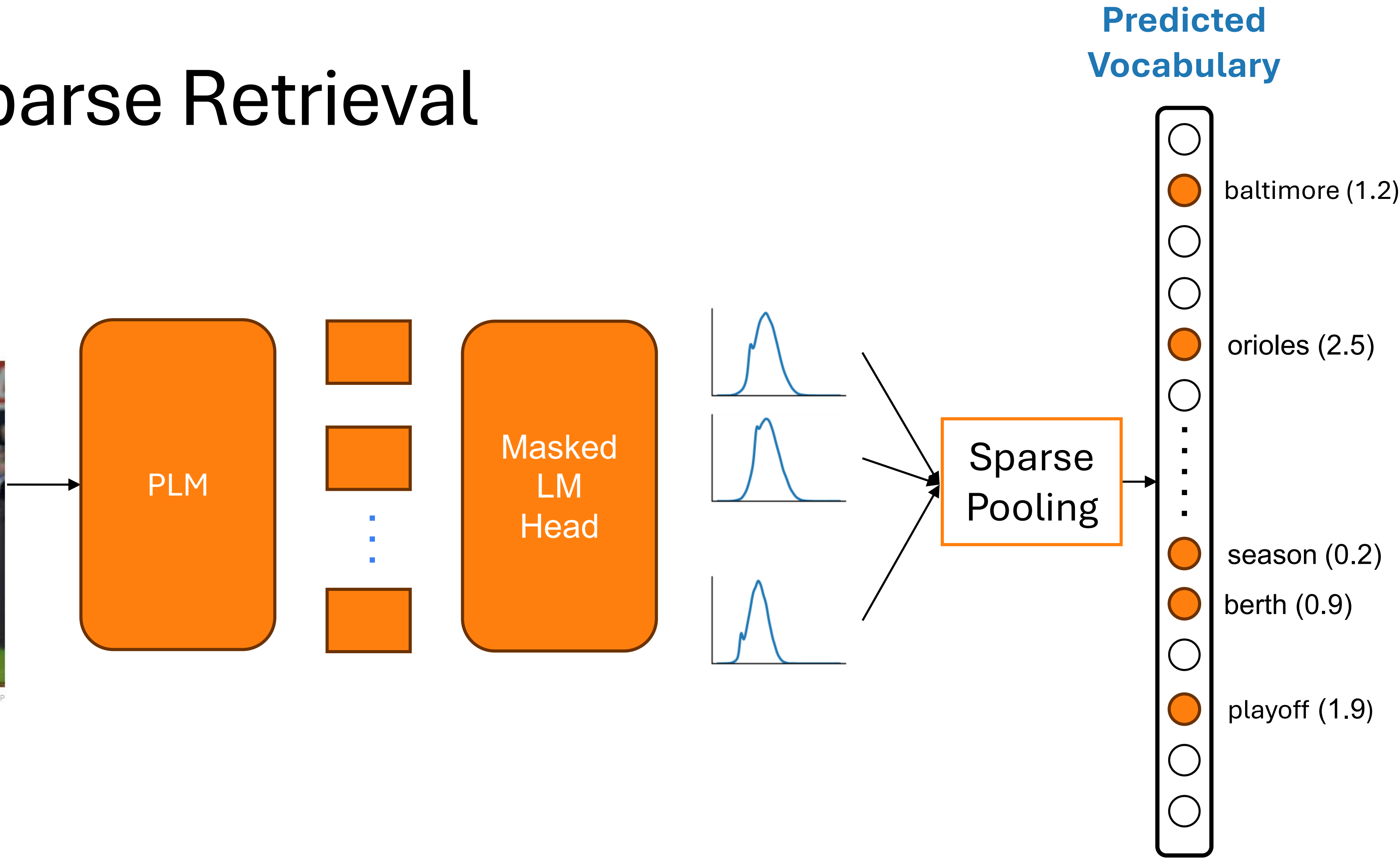
Learned Sparse Retrieval


Baltimore Orioles clinch playoff berth for 2nd straight season

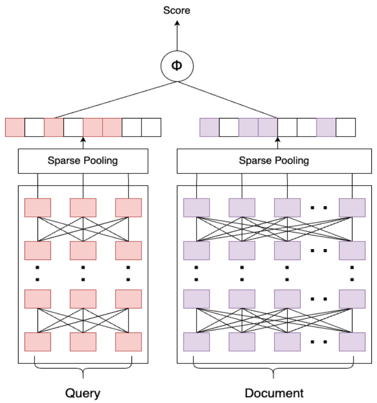
The Baltimore Orioles are headed to the playoffs in consecutive years for the first time since the 1990s, clinching no worse than a wild-card berth with a 5-3 win over the New York Yankees paired with Minnesota's loss to Miami

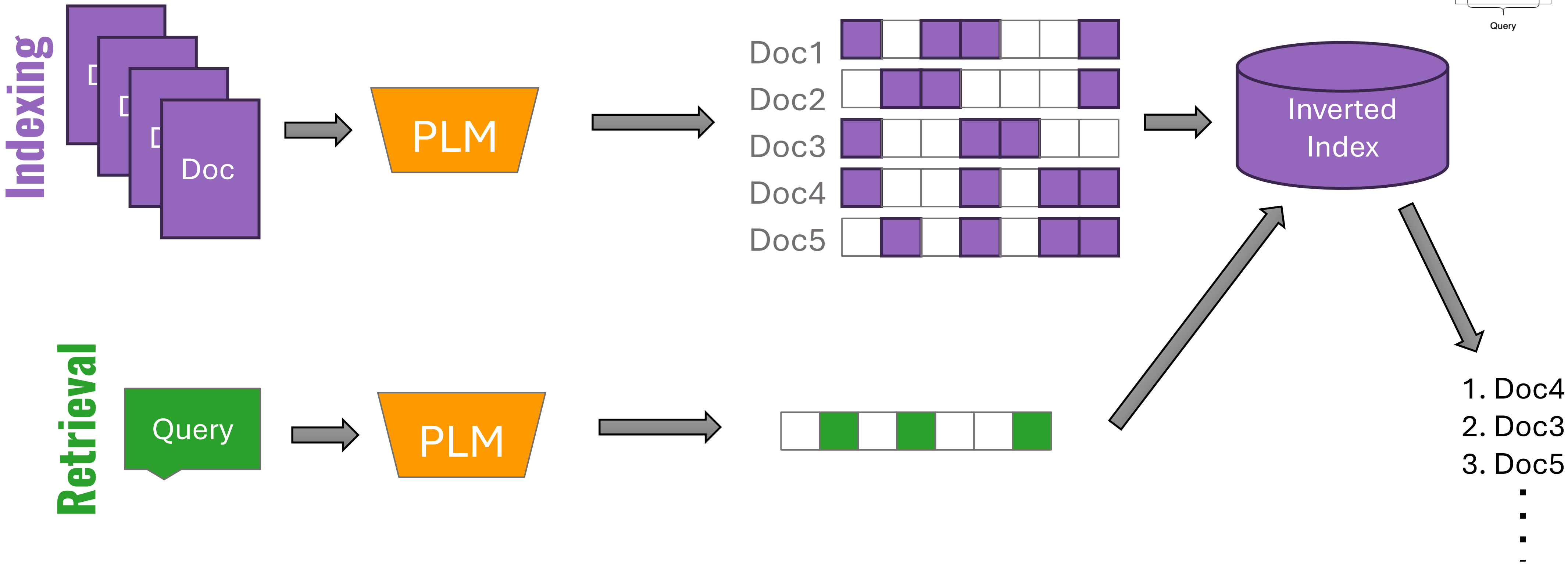
Credit: AP

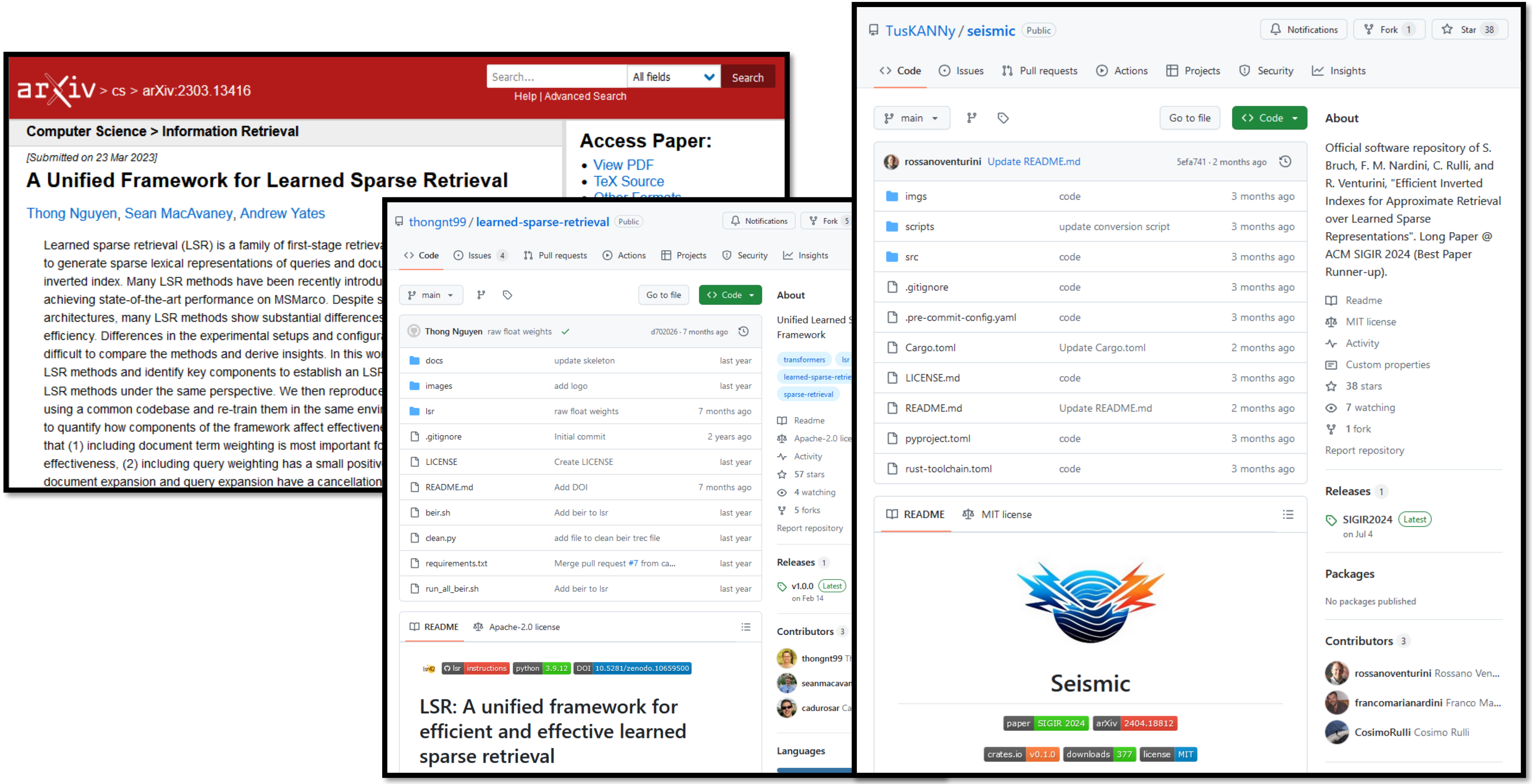
Baltimore Orioles' Anthony Santander runs the bases after hitting a home run during the sixth inning of a baseball game against the New York Yankees, Tuesday, Sept. 24, 2024, in New York. (AP Photo/Bryan Woolston)



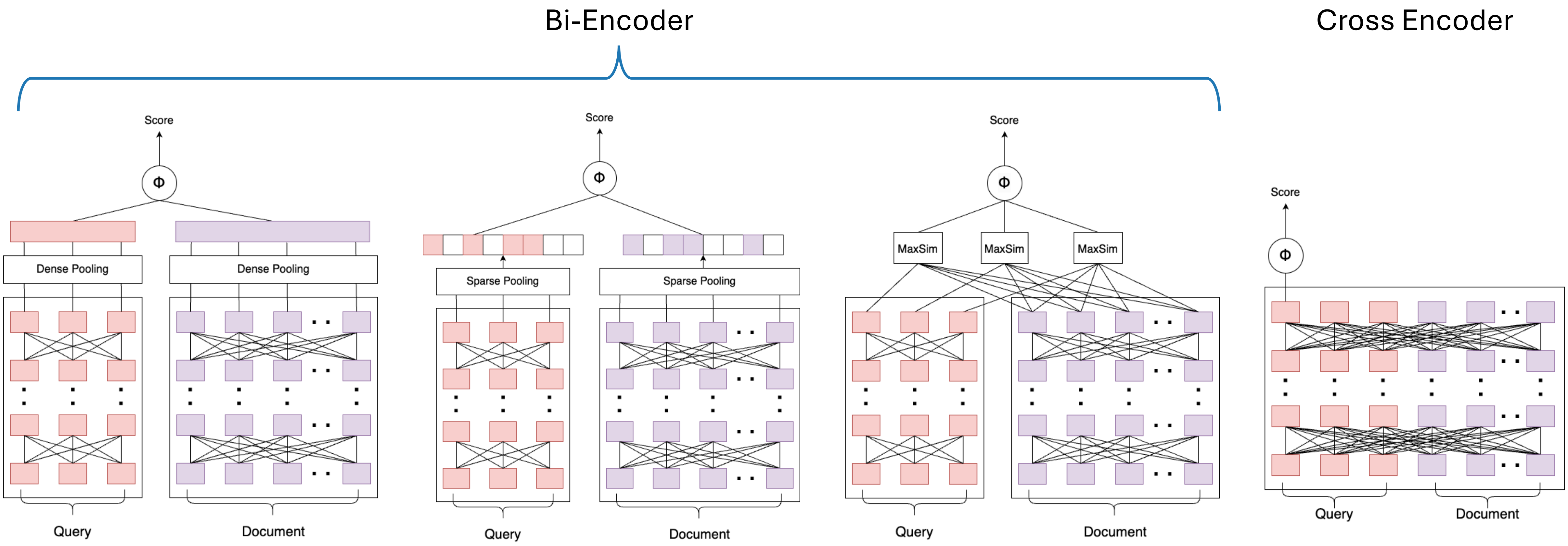
SPLADE Search Pipeline







arXiv > cs > arXiv:2303.13416
Search... All fields Search
Help | Advanced Search
Computer Science > Information Retrieval
[Submitted on 23 Mar 2023]
A Unified Framework for Learned Sparse Retrieval
Thong Nguyen, Sean MacAvaney, Andrew Yates
Access Paper:
View PDF
TeX Source
thongnt99 / learned-sparse-retrieval Public
Notifications Fork 5
Code Issues 4 Pull requests Actions Projects Security Insights
main Go to file Code
Thong Nguyen raw float weights d702026 · 7 months ago
docs update skeleton last year
images add logo last year
lsr raw float weights 7 months ago
.gitignore Initial commit 2 years ago
LICENSE Create LICENSE last year
README.md Add DOI 7 months ago
beir.sh Add beir to lsr last year
clean.py add file to clean beir trec file last year
requirements.txt Merge pull request #7 from ca... last year
run_all_beir.sh Add beir to lsr last year
README Apache-2.0 license
lsr instructions python 3.9.12 DOI 10.5281/zenodo.10659500
LSR: A unified framework for efficient and effective learned sparse retrieval
About
Readme
Activity
57 stars
4 watching
5 forks
Report repository
Releases 1
v1.0.0 Latest on Feb 14
Contributors 3
thongnt99
seanmacavan
cadurosar
Languages
TusKANNy / seismic Public
Notifications Fork 1 Star 38
Code Issues Pull requests Actions Projects Security Insights
main Go to file Code
rossanoventurini Update README.md 5efa741 · 2 months ago
imgs code 3 months ago
scripts update conversion script 3 months ago
src code 3 months ago
.gitignore code 3 months ago
.pre-commit-config.yaml code 3 months ago
Cargo.toml Update Cargo.toml 2 months ago
LICENSE.md code 3 months ago
README.md Update README.md 2 months ago
pyproject.toml code 3 months ago
rust-toolchain.toml code 3 months ago
README MIT license
Seismic
paper SIGIR 2024 arXiv 2404.18812
crates.io v0.1.0 downloads 377 license MIT
About
Official software repository of S. Bruch, F. M. Nardini, C. Rulli, and R. Venturini, "Efficient Inverted Indexes for Approximate Retrieval over Learned Sparse Representations". Long Paper @ ACM SIGIR 2024 (Best Paper Runner-up).
Readme
MIT license
Activity
Custom properties
38 stars
7 watching
1 fork
Report repository
Releases 1
SIGIR2024 Latest on Jul 4
Packages
No packages published
Contributors 3
rossanoventurini Rossano Ven...
francomarianardini Franco Ma...
CosimoRulli Cosimo Rulli



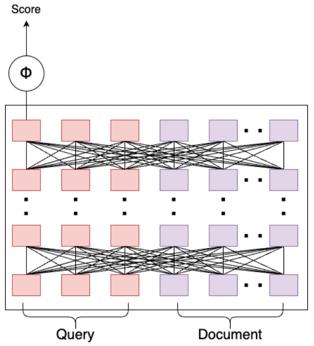
One Dense Vector
Per Sequence
e.g., DPR

One **Sparse** Vector
Per Sequence
e.g., SPLADE

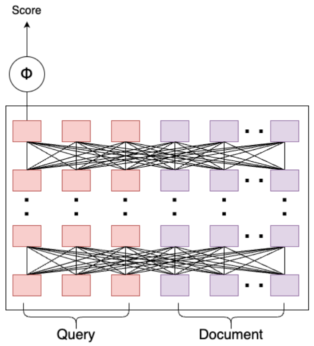
Multiple Dense Vectors
Per Sequence
e.g., ColBERT

Joint Encoder
e.g., monoBERT

Cross-Encoder as Reranker



Using Generative Models



Query: What does Mary have
Doc: Mary had a little lamb.
Relevant:

Generative
(L)LM

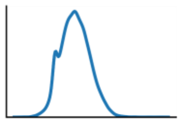
e.g, Qwen3

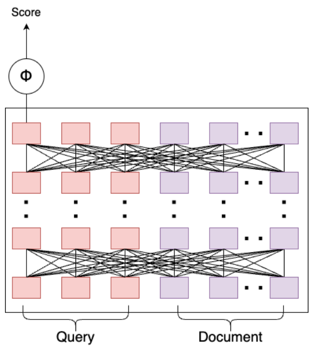
Not a number!

Yes

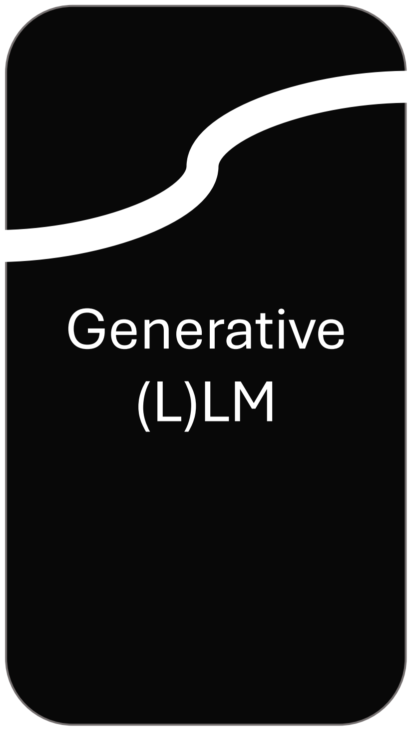
Pradeep, Ronak, Rodrigo Nogueira, and Jimmy Lin. "The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models." arXiv preprint arXiv:2101.05667 (2021).

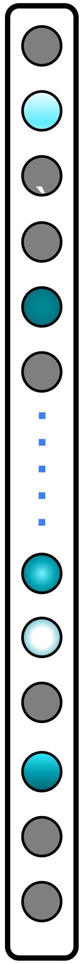
Using Generative Models





Query: What does Mary have
Doc: Mary had a little lamb.
Relevant:





Yes (0.08)

No (0.001)

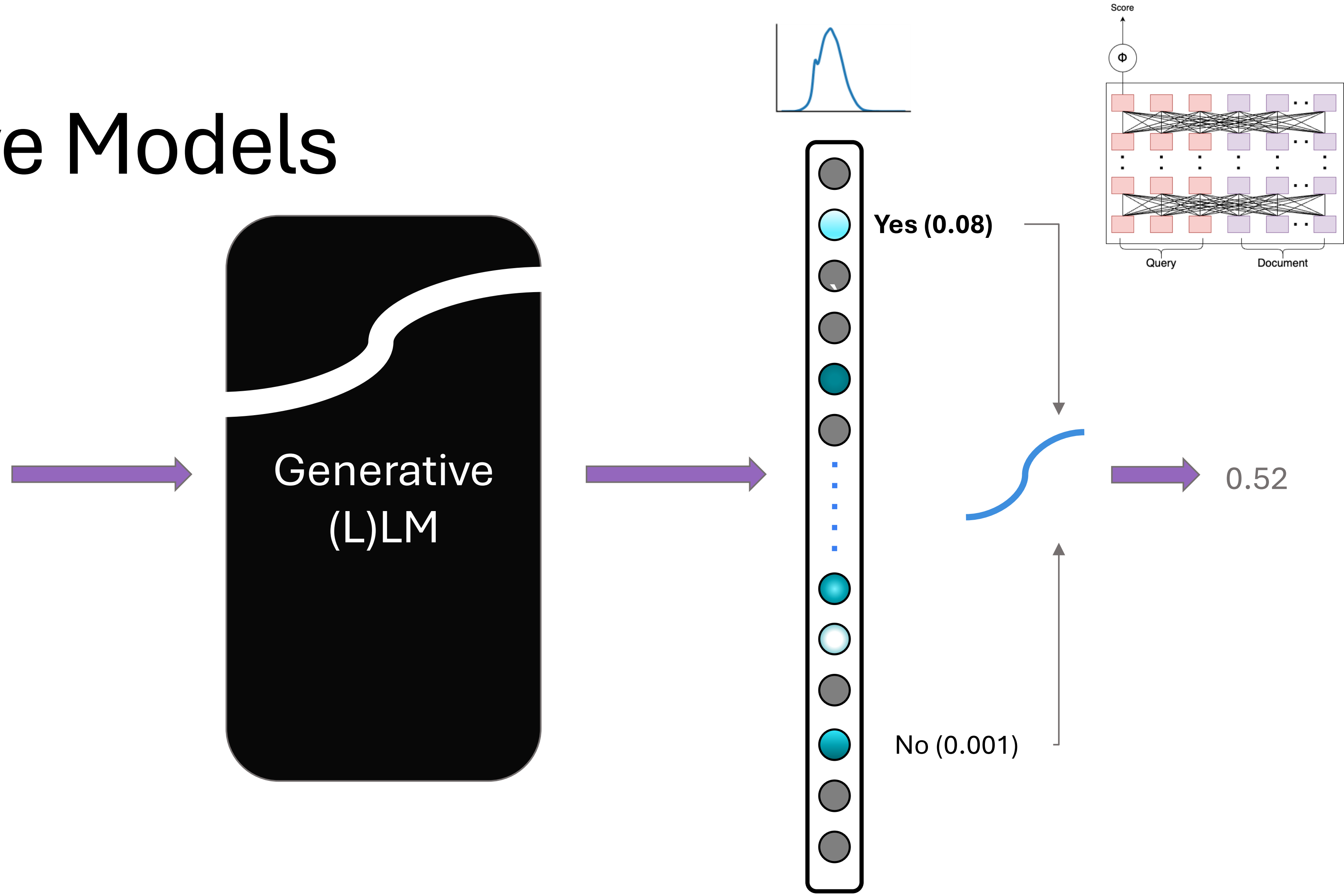
Yes

Pradeep, Ronak, Rodrigo Nogueira, and Jimmy Lin. "The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models." arXiv preprint arXiv:2101.05667 (2021).

Using Generative Models

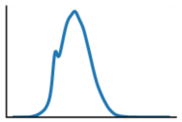
Pointwise score

Query: What does Mary have
Doc: Mary had a little lamb.
Relevant:



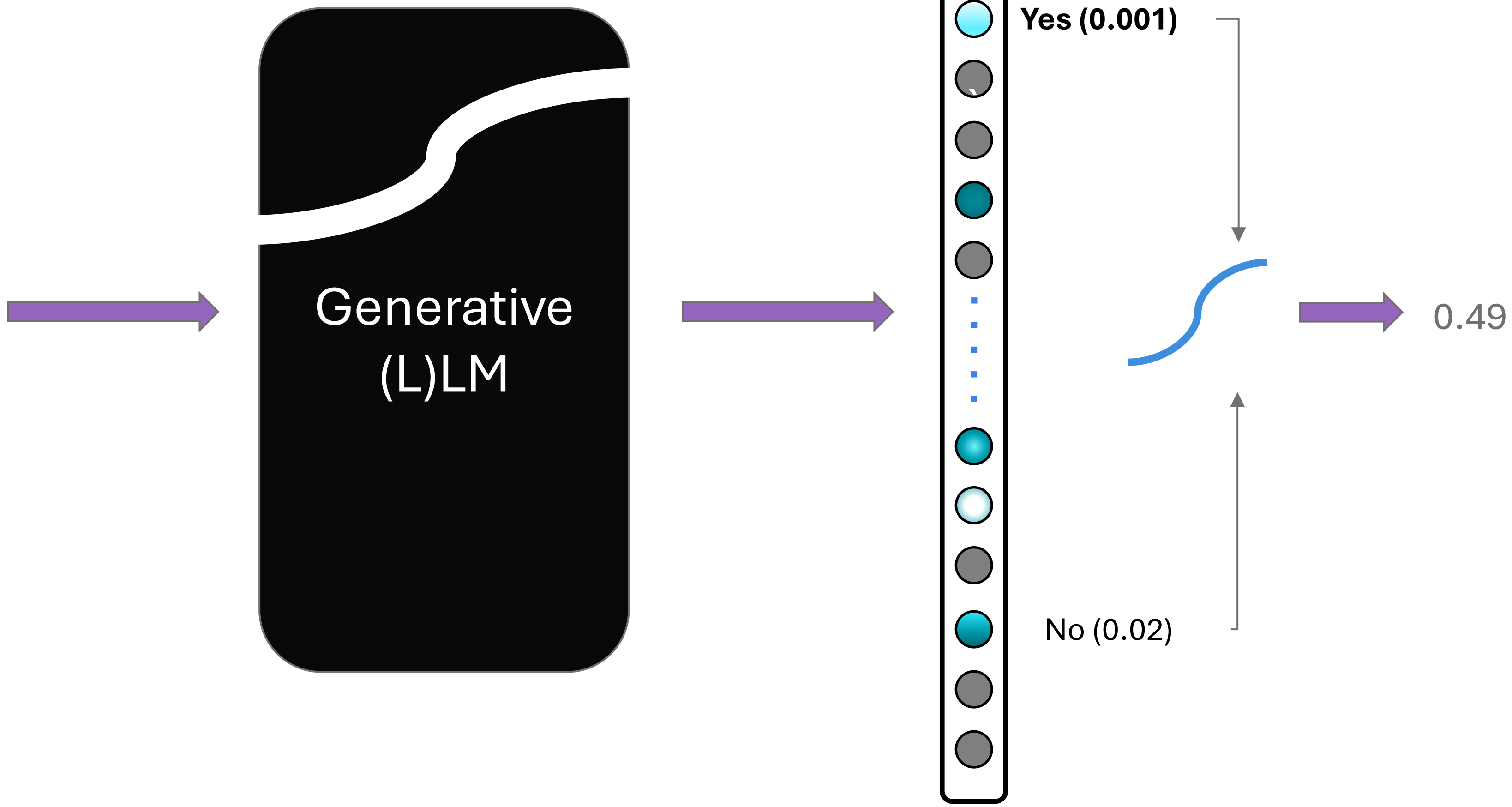
Pradeep, Ronak, Rodrigo Nogueira, and Jimmy Lin. "The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models." arXiv preprint arXiv:2101.05667 (2021).

Using Generative Models



Pairwise score

Query: What does Mary have
Doc0: JHU is in Baltimore
Doc1: Mary had a little lamb.
Relevant:



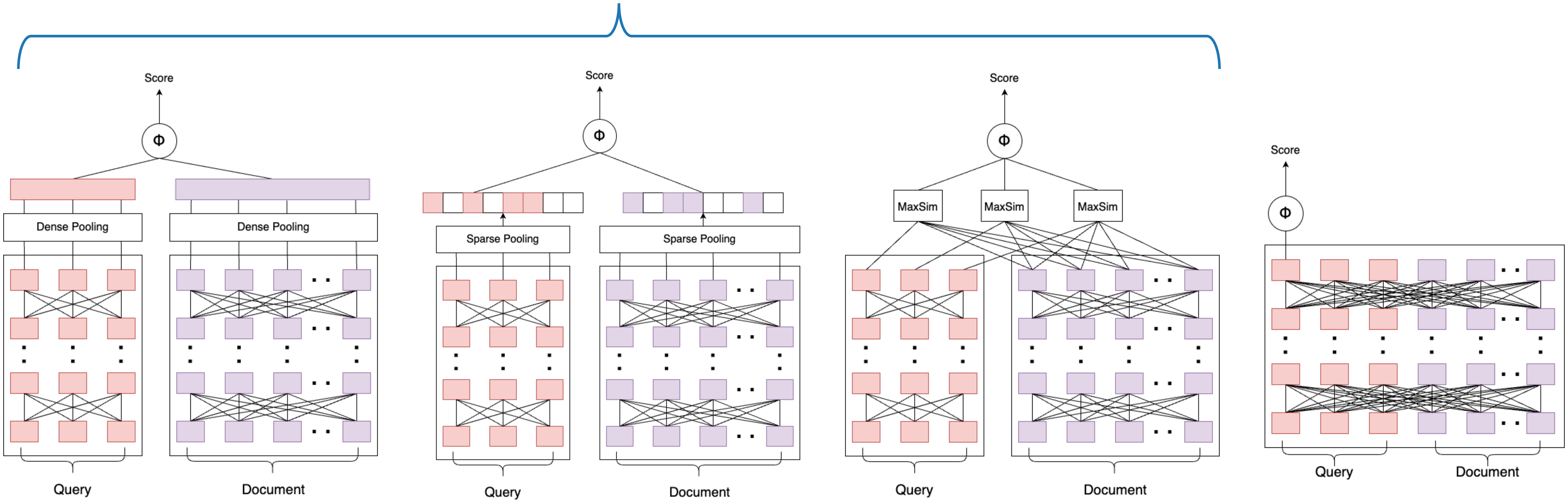
Pradeep, Ronak, Rodrigo Nogueira, and Jimmy Lin. "The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models." arXiv preprint arXiv:2101.05667 (2021).

Many kinds of LLM-based Rerankers

	Generative Model	Reasoning Model
Pointwise	MonoT5 RankLlama	Rank1
Pairwise	DuoT5	
Listwise/Setwise	RankGPT RankZephyr	Rank-K ReasonRank

Bi-Encoder

Cross Encoder



One Dense Vector
Per Sequence
e.g., DPR

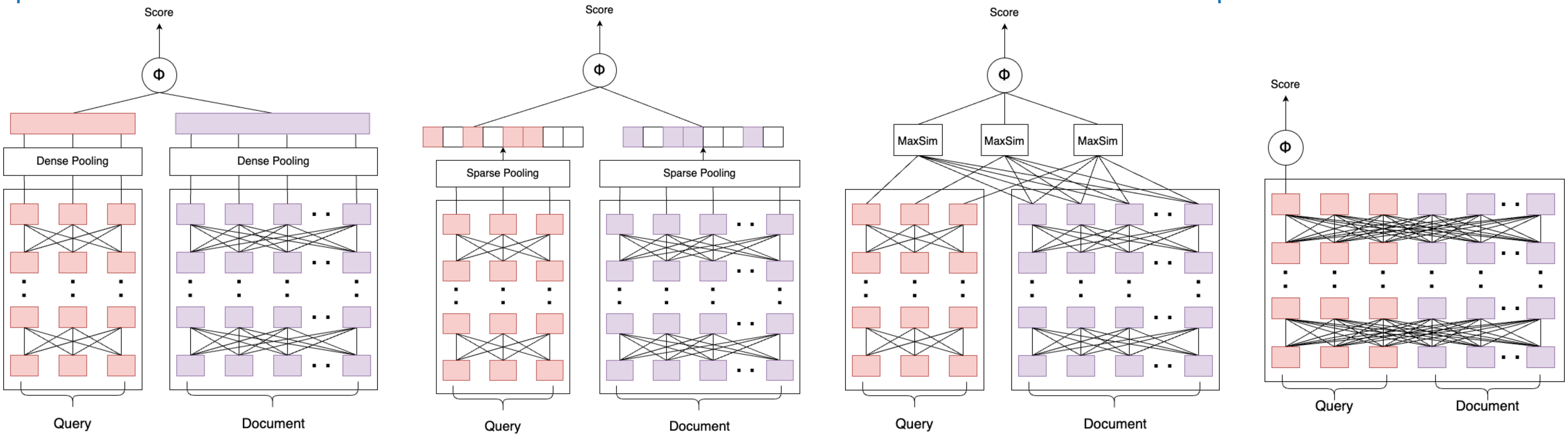
One **Sparse** Vector
Per Sequence
e.g., SPLADE

Multiple Dense Vectors
Per Sequence
e.g., ColBERT

Joint Encoder
e.g., monoBERT

Bi-Encoder

Cross Encoder



One Dense Vector
Per Sequence
e.g., DPR

One **Sparse** Vector
Per Sequence
e.g., SPLADE

Multiple Dense Vectors
Per Sequence
e.g., ColBERT

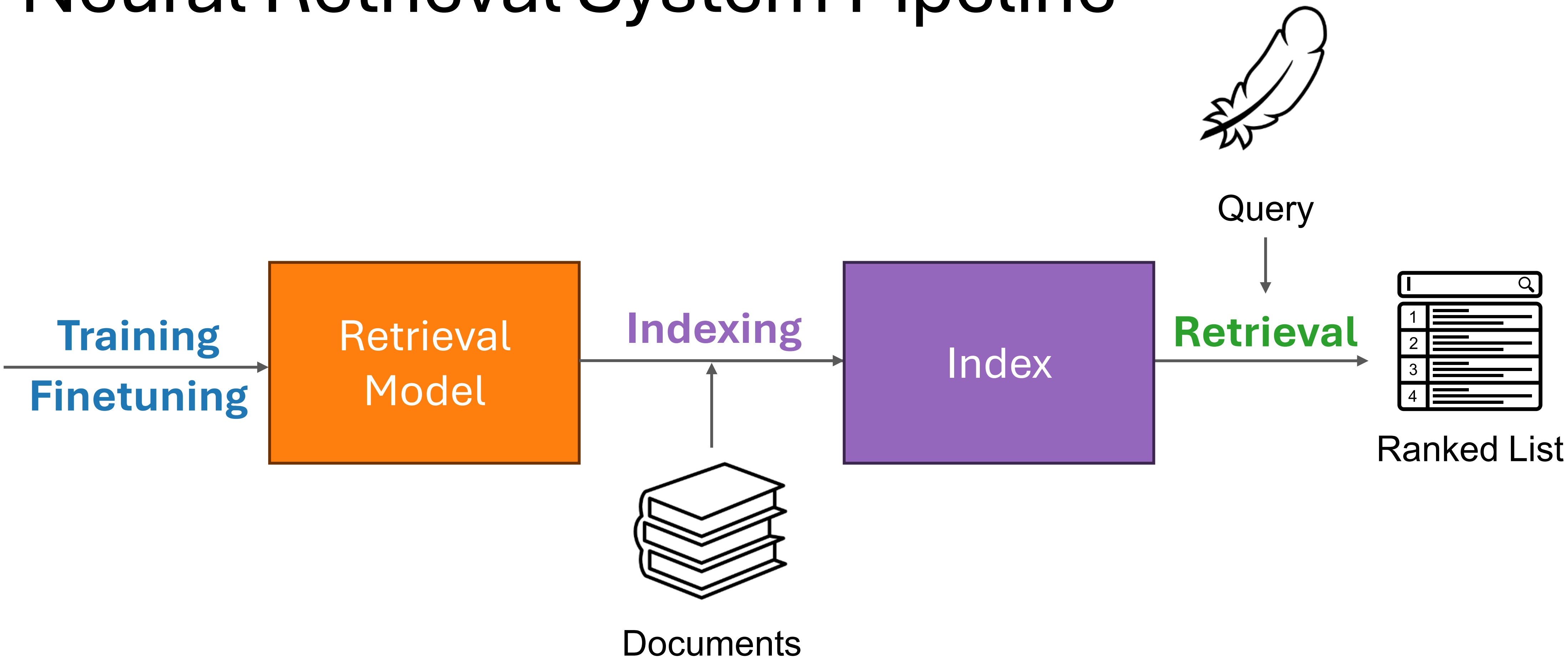
Joint Encoder
e.g., Rank1

More Effective

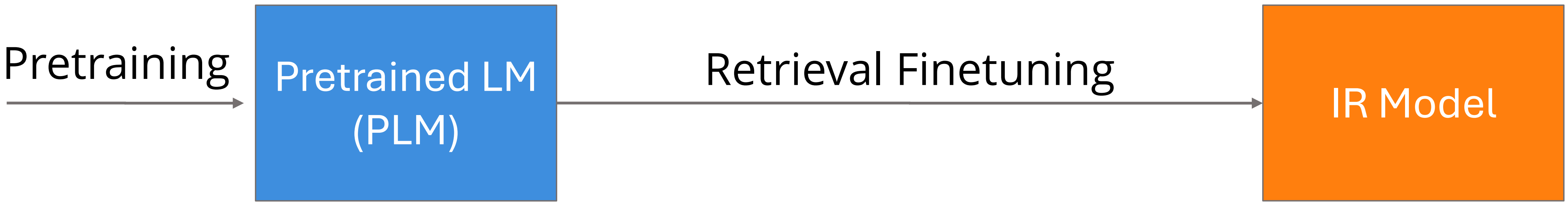
More Efficient (at Query Time)

Retrieve-and-Rerank System Combinations

Neural Retrieval System Pipeline



PLM to IR Model



- Align the representation
- Model "relevancy"

Evaluation

Which system is better?

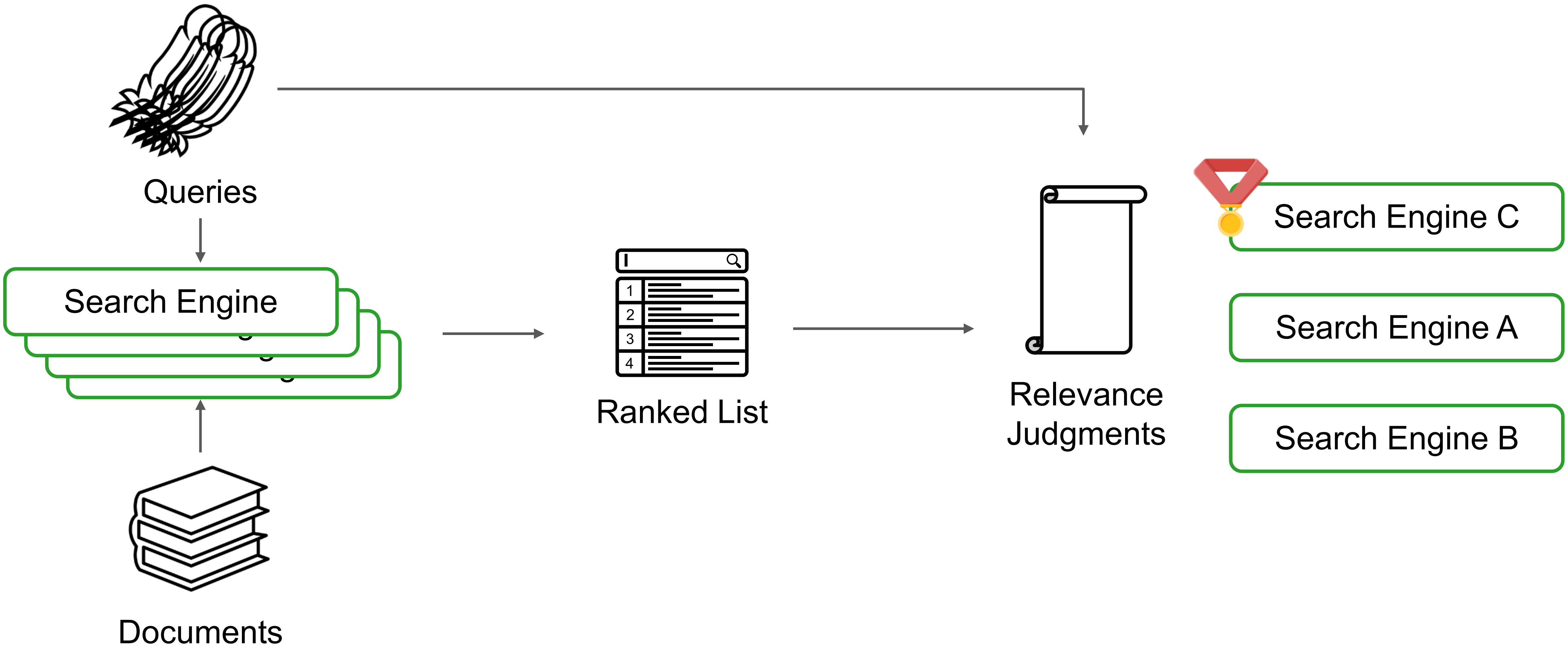
What is Information Retrieval?

(relevant)

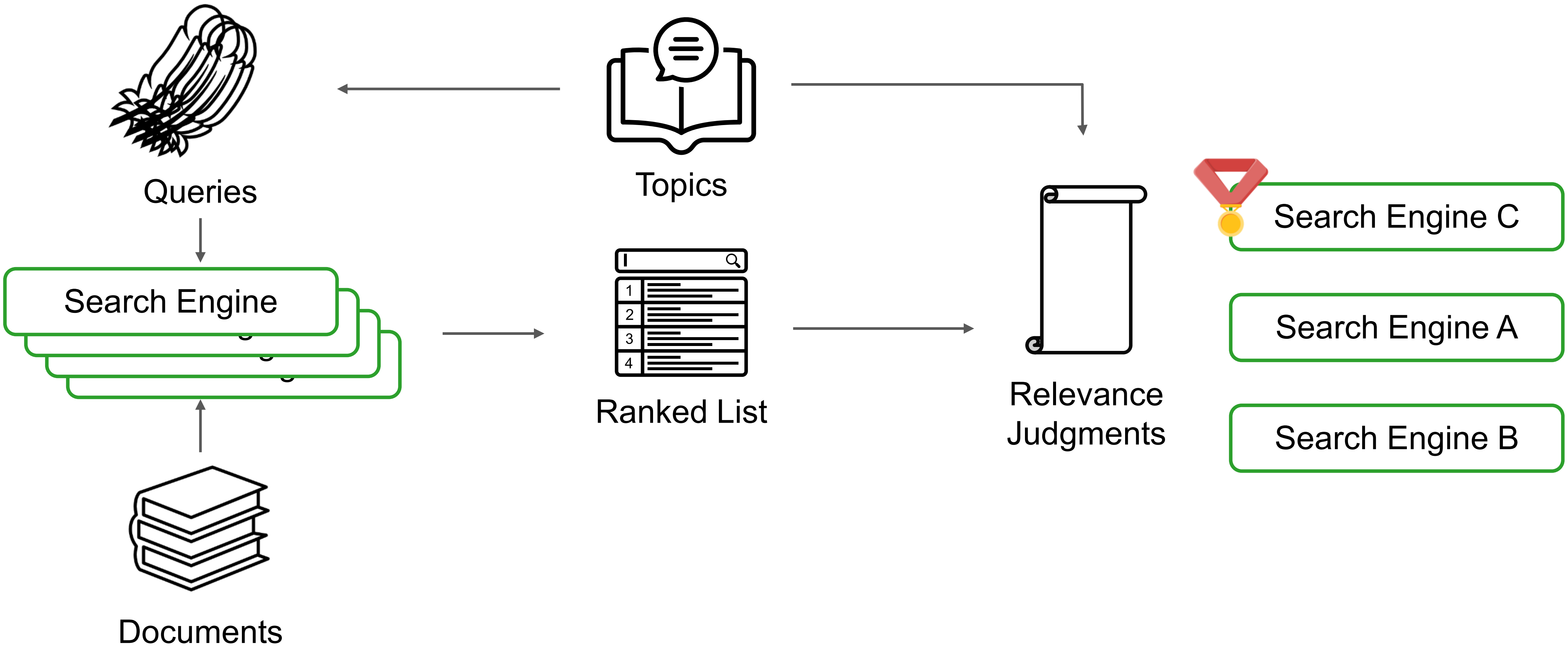
Retrieve information from a storage based on user's information need

Which system retrieve more relevant information?

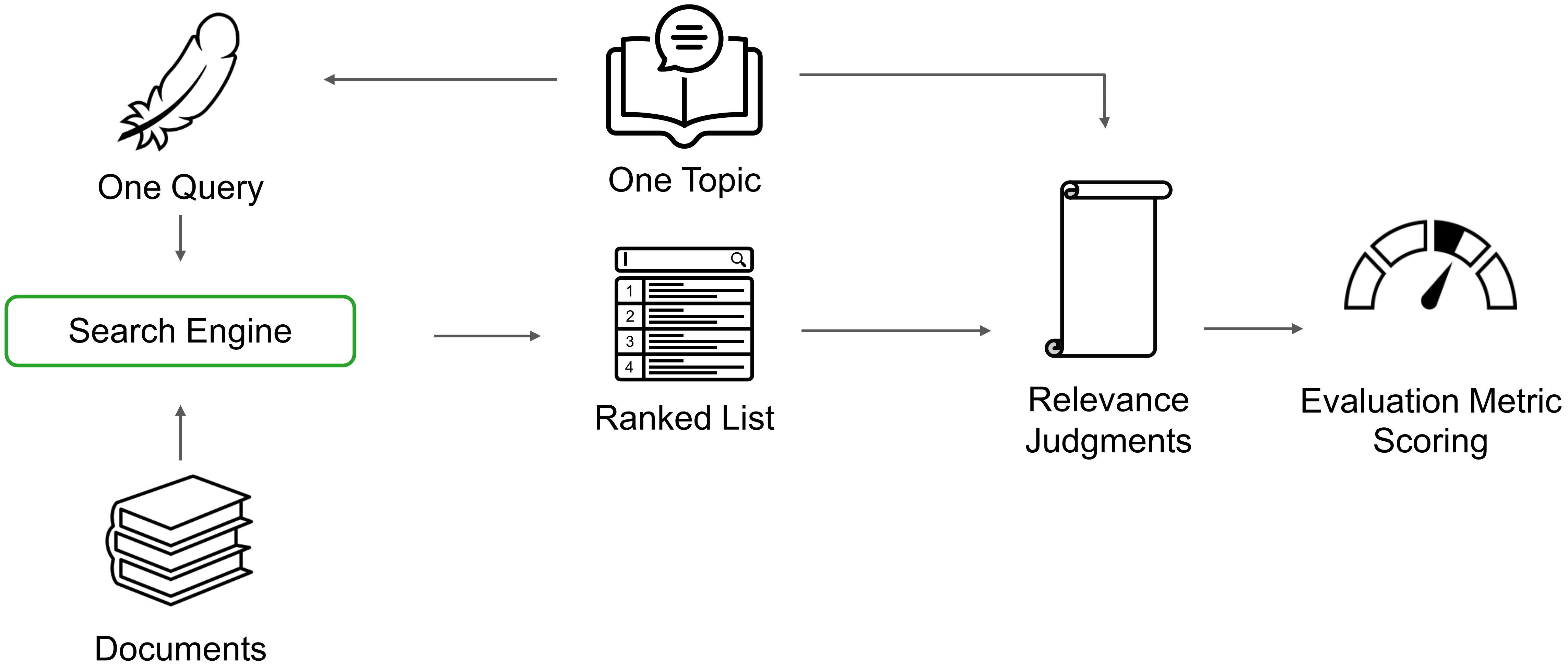
Cranfield Paradigm Evaluation



Cranfield Paradigm Evaluation



Cranfield Paradigm Evaluation



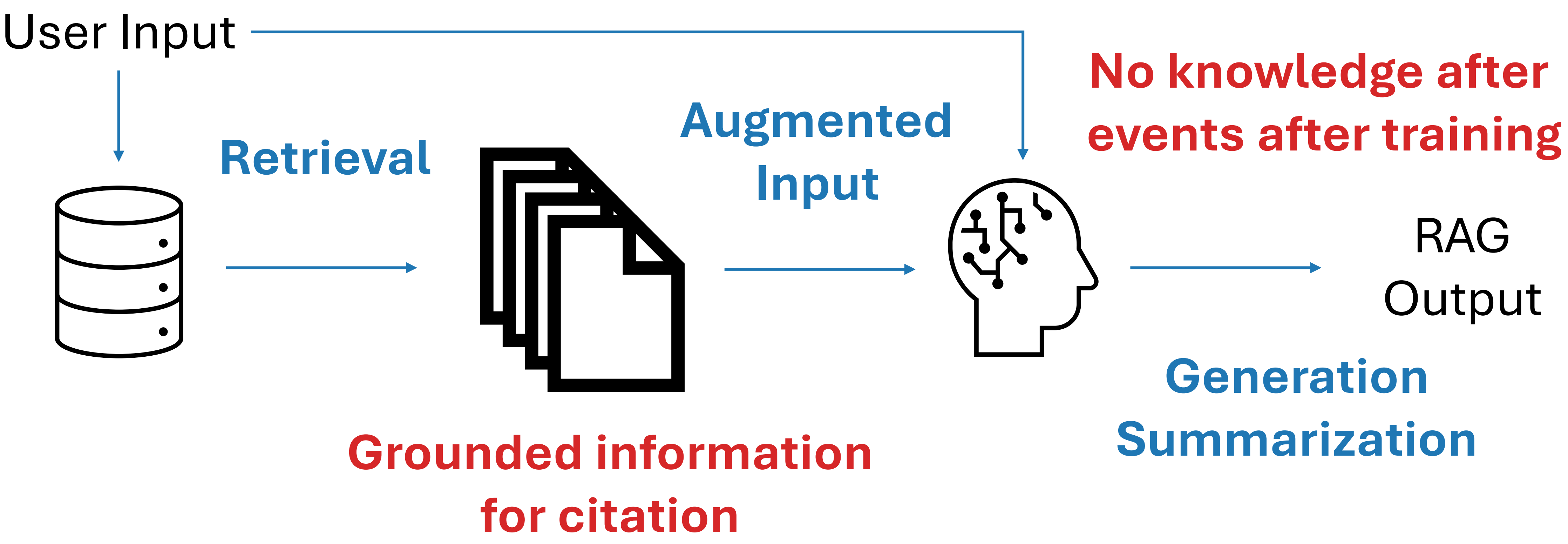
IR-Specific Issues

- Topics vs Queries
  - Clear intent vs an expression of such intent
- Relevant vs related
  - Fulfilling the information need or not
- Relevance Judgements vs Labels
  - Opinion vs “fact”
- Ranked retrieval metrics
  - Measuring the quality/effectiveness of a ranked list

IR Metrics

- Effective Metrics
  - Mean Average Precision
  - Normalized Discounted Cumulative Gain
  - Recall@k
- Efficiency Metrics
  - Indexing time
  - Index disk space
  - Query latency (average search time per query)

Retrieval Augmented Generation


User Input
Retrieval
Augmented
Input
No knowledge after
events after training
RAG
Output
Grounded information
for citation
Generation
Summarization

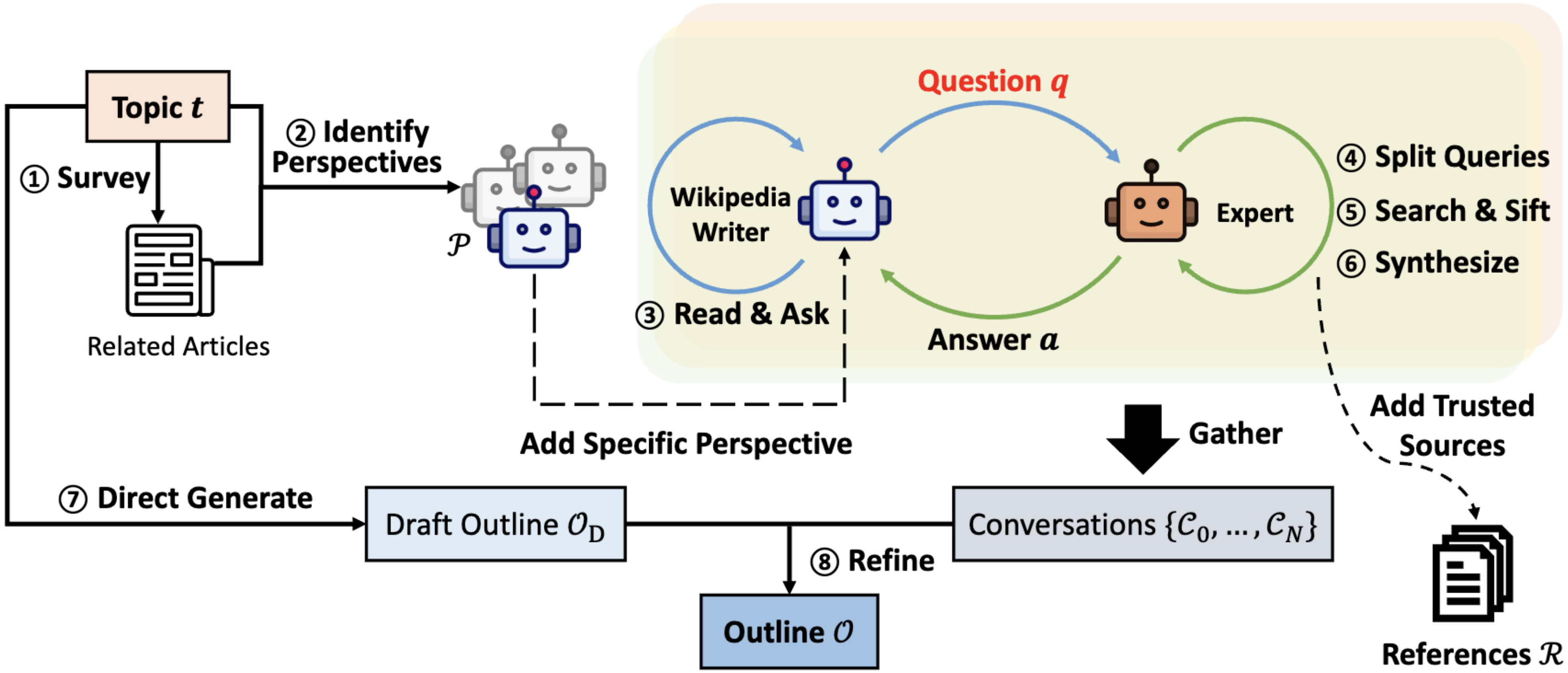
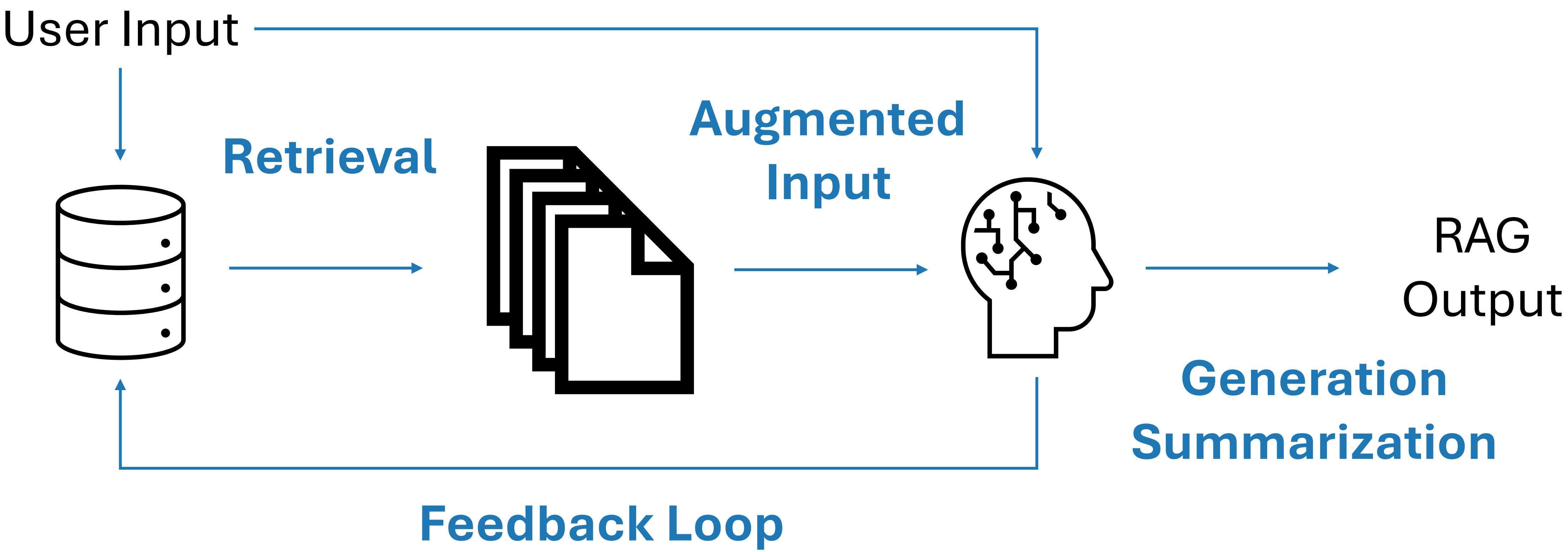


Figure 2: The overview of STORM that automates the pre-writing stage. Starting with a given topic, STORM identifies various perspectives on covering the topic by surveying related Wikipedia articles (①-②). It then simulates conversations between a Wikipedia writer who asks questions guided by the given perspective and an expert grounded on trustworthy online sources (③-⑥). The final outline is curated based on the LLM's intrinsic knowledge and the gathered conversations from different perspectives (⑦-⑧).

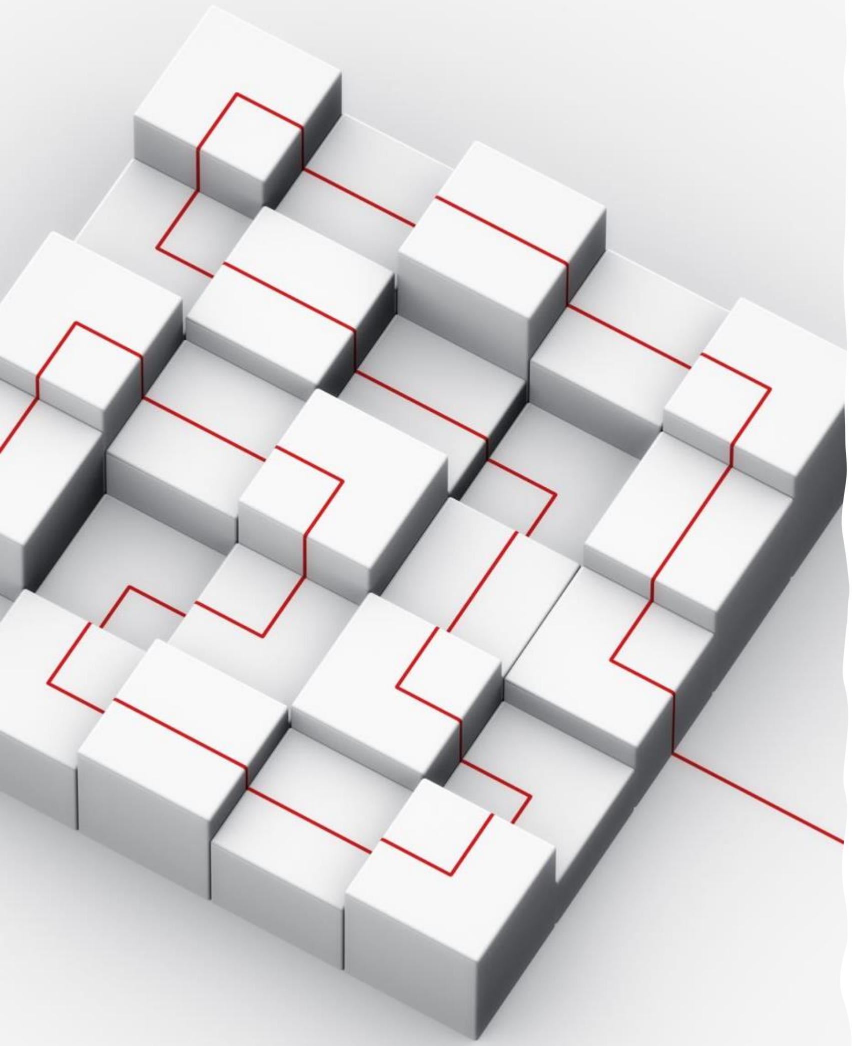
Shao, Yijia, et al. "Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models." Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). 2024.

A Green Field

- How the pipeline should be designed?
  - Predefined one or agentic system?
- Short vs long input from user
- How to make the response to be more useful?
  - Graphical or just text?
- How to interact with human?

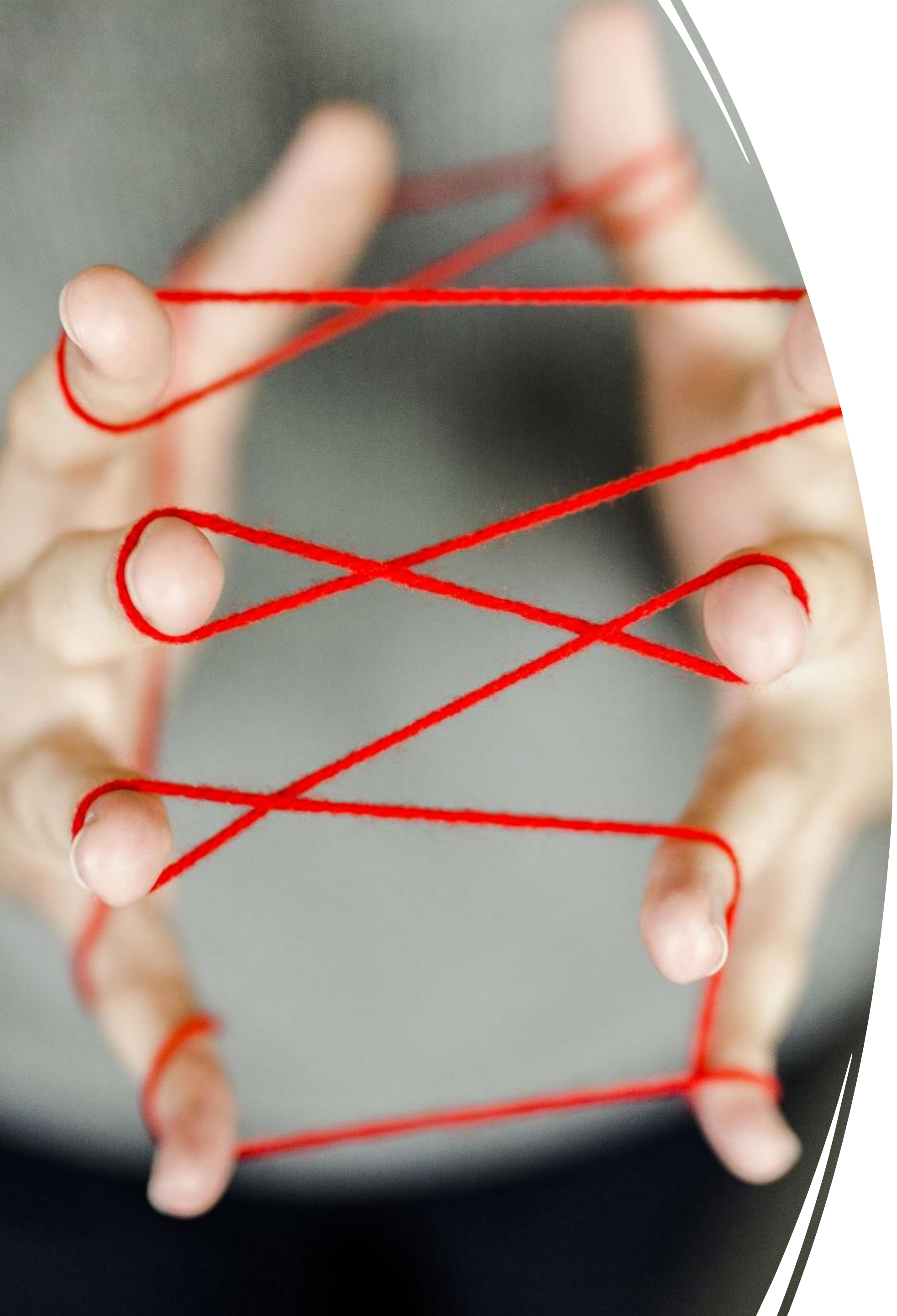

User Input
Retrieval
Augmented
Input
RAG
Output
Generation
Summarization
Feedback Loop

State of IR Research



Better Retrieval Models

- More effective
  - Better/larger neural models
  - Better architecture?
  - Under harder setup, e.g., scholar search, multilingual, cross-modal, etc
- More efficient
  - Faster at query time
  - Less resource footprint, e.g., memory, storage, compute, etc
- Other qualities
  - Fairness, diversity, etc



Other Retrieval Problems

- Conversational
  - Guessing intent, finding the “right” information to serve
- Iterative/interactive/human-in-the-loop
  - Rounds of interactions
- Generative
  - Returning a piece of text

Evaluation

- What to measure
  - and when would it fail
- How to measure
  - Generative text? Citations?
- “Better” evaluation collection
  - Not necessarily larger