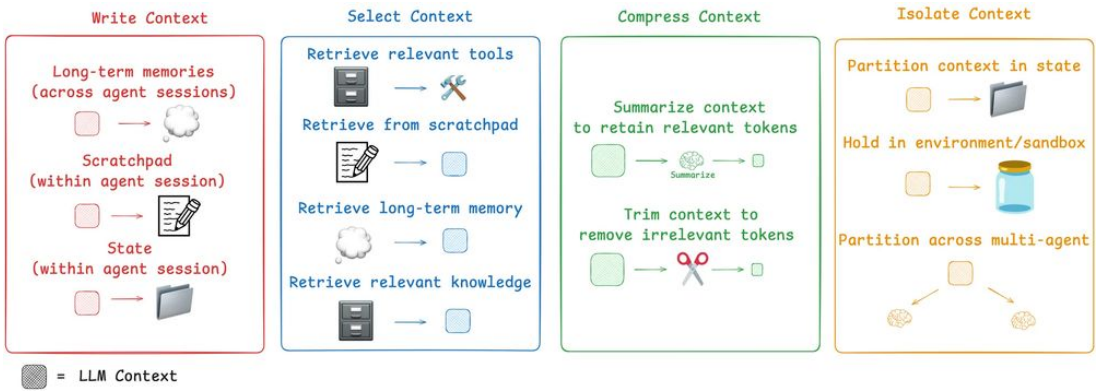

# RAG @ ETAL

Jose G Moreno
04/09/2025


# Prompt engineering is dead! Long live to Context engineering!

# Prompt engineering is dead! Long live to Context engineering!

- Common misconception:
  - Prompts are often thought of as brief task descriptions used in everyday interactions with large language models
- Reality in industrial-strength LLM apps:
  - Context engineering is a crucial, complex process
  - It’s the art and science of carefully filling the LLM's context window with the right data for optimal performance




Source: https://blog.langchain.com/context-engineering-for-agents/

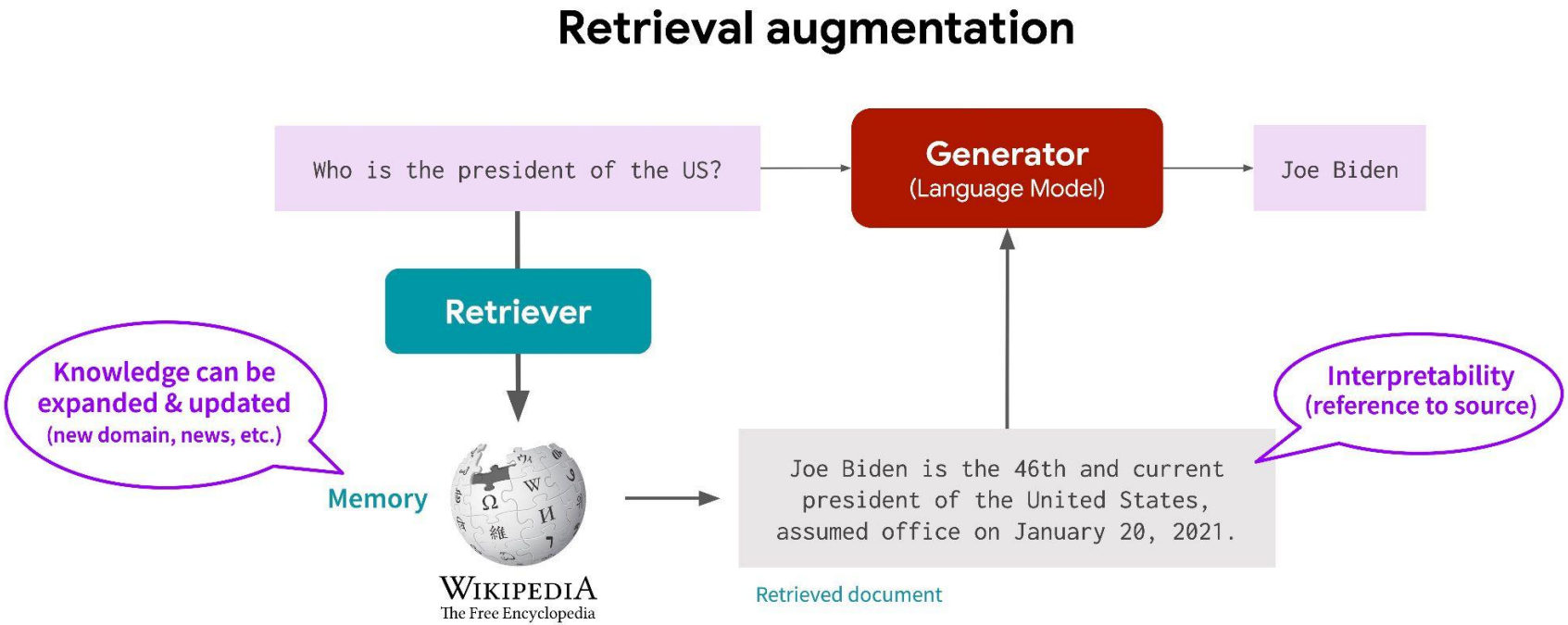
# Why is building an adequate context a challenging task?

- Involves:
  - Task descriptions and explanations
  - Few-shot examples
  - Retrieval-Augmented Generation (RAG)
  - Related (possibly multimodal) data
  - Tool outputs
  - System state and history
  - Data compacting and formatting
- Errors to avoid:
  - Too little or irrelevant info → poor performance
  - Too much or noisy info → increased cost, degraded output

# Why is building an adequate context a challenging task?

- Involves:
  - Task descriptions and explanations
  - Few-shot examples
  - **Retrieval-Augmented Generation (RAG)**
  - Related (possibly multimodal) data
  - Tool outputs
  - System state and history
  - Data compacting and formatting
- Errors to avoid:
  - **Too little or irrelevant info → poor performance**
  - **Too much or noisy info → increased cost, degraded output**

# What is RAG? (naive version)

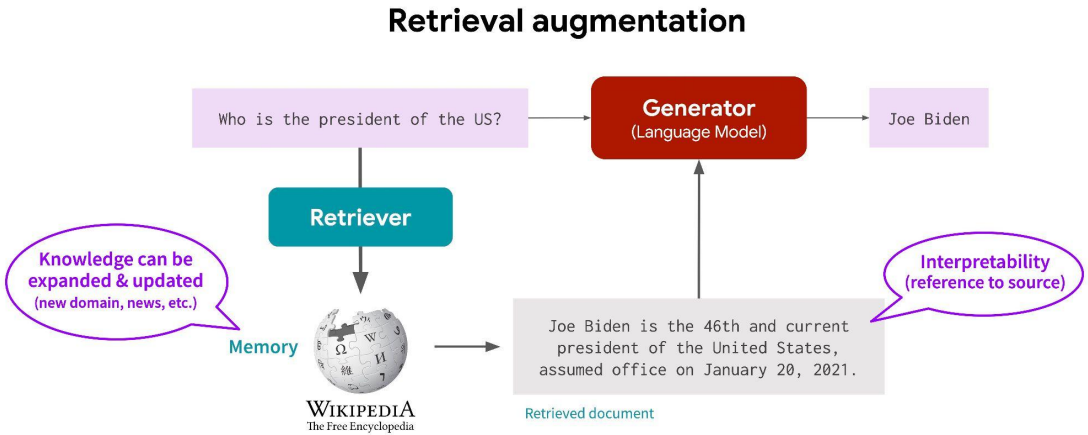



Source: https://cs.stanford.edu/~myasu/blog/racm3/


Words once generated cannot be taken back

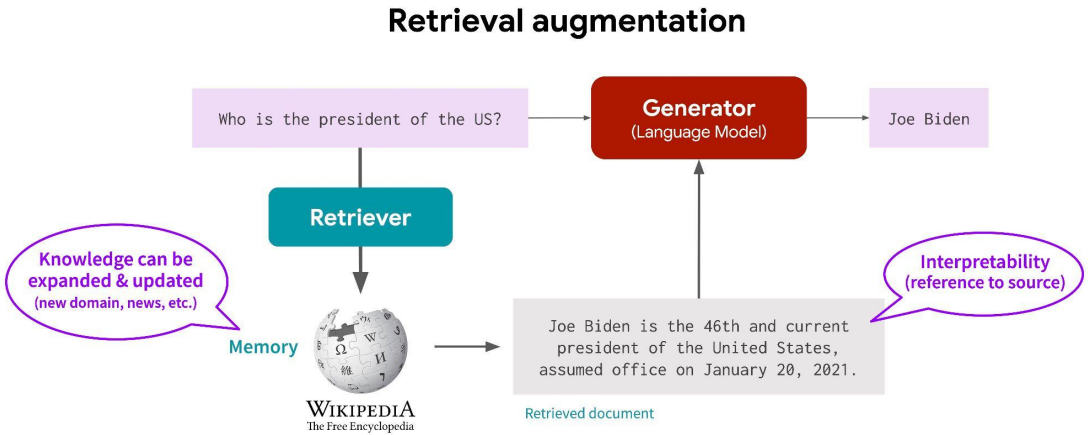
# RAG has emerged as a predominant solution because

- Easy to implement
- Allows access to external knowledge
- Ensure interpretability
- Allows provenance tracking



# RAG has emerged as a predominant solution because

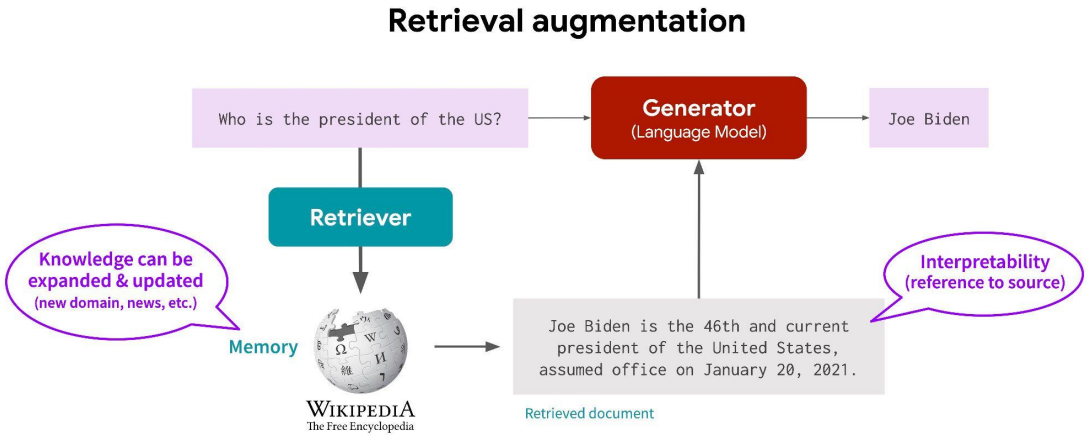
- Easy to implement
- Allows access to external knowledge
- Ensure interpretability
- Allows provenance tracking




**but RAG is dead!!!! long live to long context!!!**

# RAG has emerged as a predominant solution because

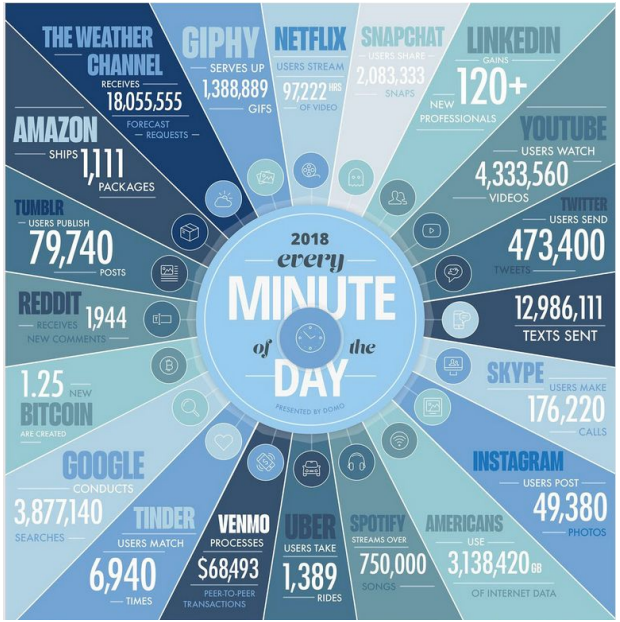
- Easy to implement
- Allows access to external knowledge
- Ensure interpretability
- Allows provenance tracking




**but RAG is dead!!!! long live to long context!!!**
**…can you imagine full web into a 1-2M prompt???**

Better to **remain silent** and be **thought a fool** than to **generate** and **remove all doubt**…

# Question answering / Information access - challenges



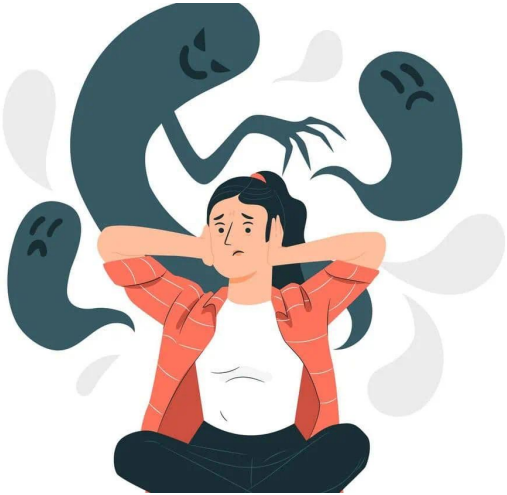


Can we update the model's knowledge without updating its parameters?

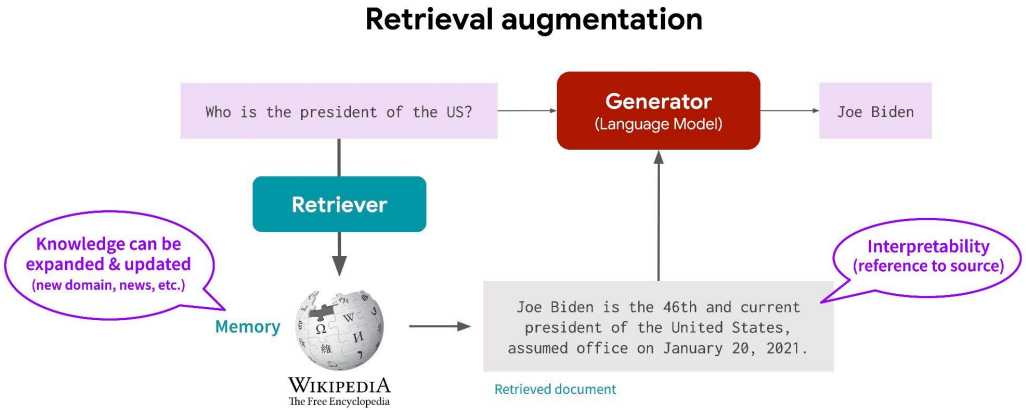
Image source:https://datavizblog.com/2018/06/17/infographic-how-much-data-is-generated-every-minute-of-the-day/

# Limitations of PLMs (& LLMs)

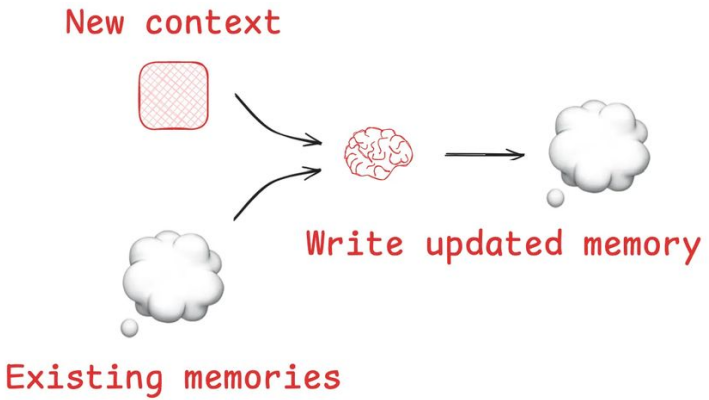
- Hallucination problem (e.g., factual errors)
- Long-tail knowledge (e.g., domain-specific) may not be well-represented in the model's pretraining corpus
- Cannot easily expand or update parameters after pretraining
  - knowledge learned during pretraining (parametric knowledge) is static
- Source of information is non-attributable



# How can we tackle these limitations?




# of course, RAG is one of the options!



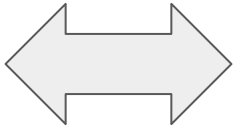
# How does Information Retrieval contribute to RAG?

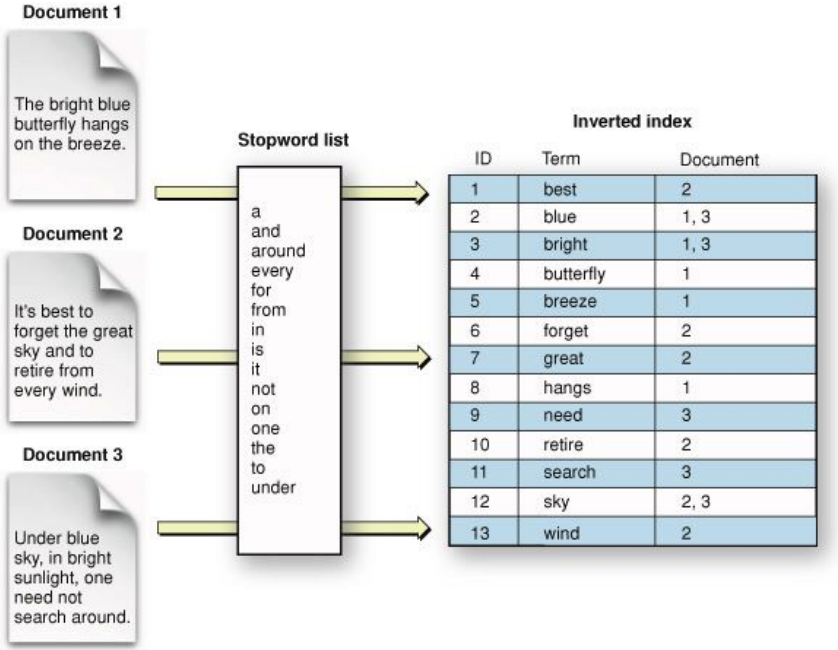
- Precise knowledge access mechanism
- Easy update to known knowledge (update the retrieval knowledge base)
- Neural Retrieval starting to outperform traditional IR
- Limitations:
  - No automatic way to setup and evaluate its contribution
  - It is a task specific way to integrate IR

# Basic concepts in RAG - Section overview

- Information Retrieval
  - Classic retrieval
    - Software / Datasets / Metrics
  - Neural retrieval
- Retrieval-augmented models
  - Retrieval-augmented PLMs/LLMs
  - Taxonomy of RAG foundations
- Generation with extra context
  - Advantages
  - Current and future enhancements

# Finding a document in a large document collection

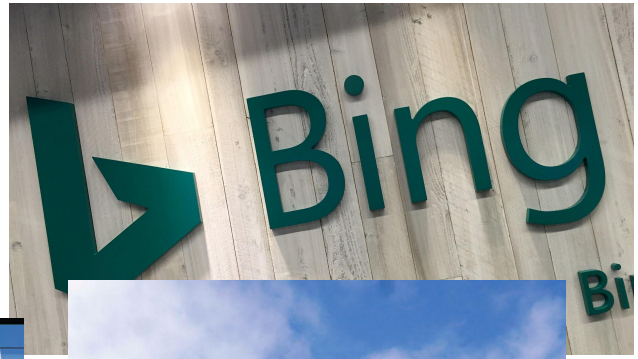




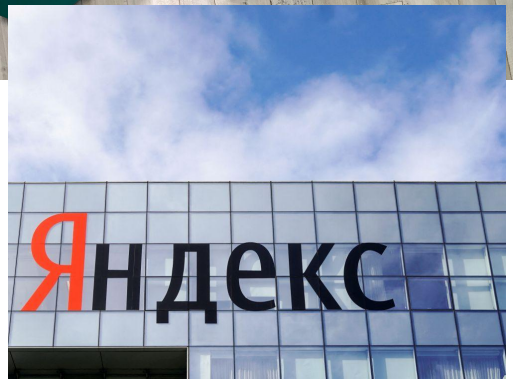
# Software for IR

(Generation n-1)

# The classics... but not open source ;)




Qwant

# Free “traditional” Software for information retrieval

## Pages in category "Free search engine software"

The following 19 pages are in this category, out of 19 total. This list may not reflect recent changes.

**A**

- Anna's Archive
- Apache Lucene
- Apache Nutch
- Apache Solr

**D**

- DocFetcher

**G**

- Gigablast
- Grub (search engine)

**H**

- Ht-//Dig

**I**

- Isearch

**M**

- MnoGoSearch

**O**

- OpenSearchServer

**S**

- Searx
- Sphinx (search engine)
- StrangeSearch
- SWISH-E

**T**

- Terrier (search engine)

**X**

- Xapian

**Y**

- YaCy

**Z**

- Zettair

Source: https://en.wikipedia.org/wiki/Category:Free_search_engine_software

# Free “traditional” Software for information retrieval

## Pages in category "Free search engine software"

The following 19 pages are in this category, out of 19 total. This list may not reflect recent changes.

**A**

- Anna's Archive
- Apache Lucene
- Apache Nutch
- Apache Solr

**D**

- DocFetcher

**G**

- Gigablast
- Grub (search engine)

**H**

- Ht-//Dig

**I**

- Isearch

**M**

- MnoGoSearch

**O**

- OpenSearchServer

**S**

- Searx
- Sphinx (search engine)
- StrangeSearch
- SWISH-E

**T**

- Terrier (search engine)

**X**

- Xapian

**Y**

- YaCy

**Z**

- Zettair

Source: https://en.wikipedia.org/wiki/Category:Free_search_engine_software

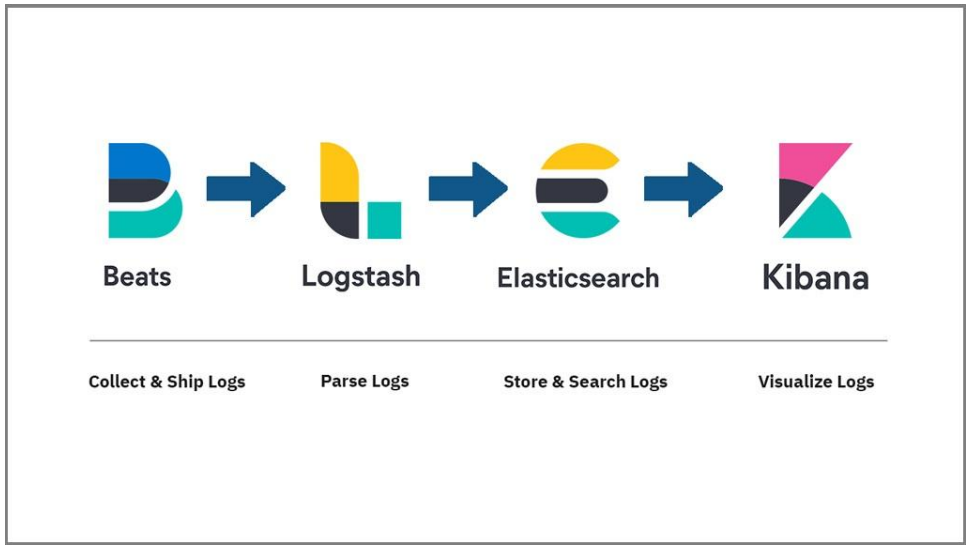
# Elasticsearch

Elasticsearch is a distributed RESTful search and analytics engine designed to address a wide range of use cases.

Elasticsearch was launched in February 2010.

It was developed by Shay Banon.

Elasticsearch is licensed under Apache 2.0.



# Elasticsearch

Elasticsearch is a search engine based on **Apache Lucene** that stores, extracts, and manages text-based and/or semi-structured data.

It enables **real-time search and analysis of textual,** numerical, or geospatial data, whether structured or unstructured.

Current version: 9.1.2.

This software is designed for use on powerful servers, so there is no default graphical module.

# In the shoulder of giants

The latest JDK 24 is recommended as the software required to run Elasticsearch on your device

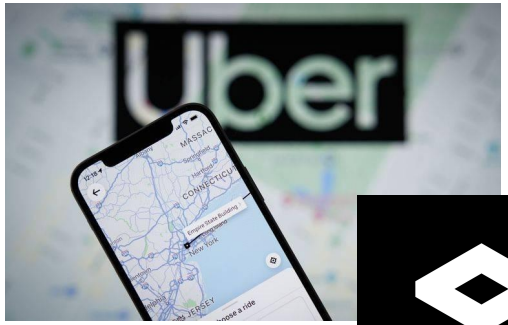
This is because Elasticsearch is based on Lucene, which is written in Java

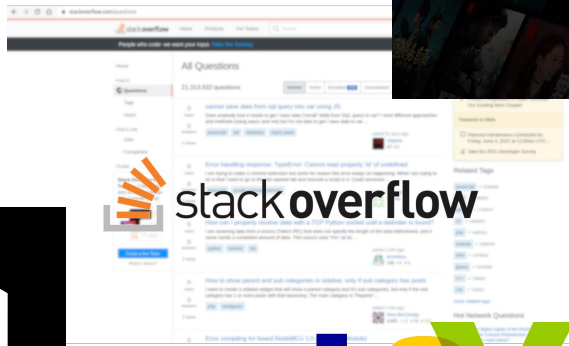
Lucene is the common core of several search engines (Terrier, Solr, etc.)

# Elasticsearch is used by…

slack



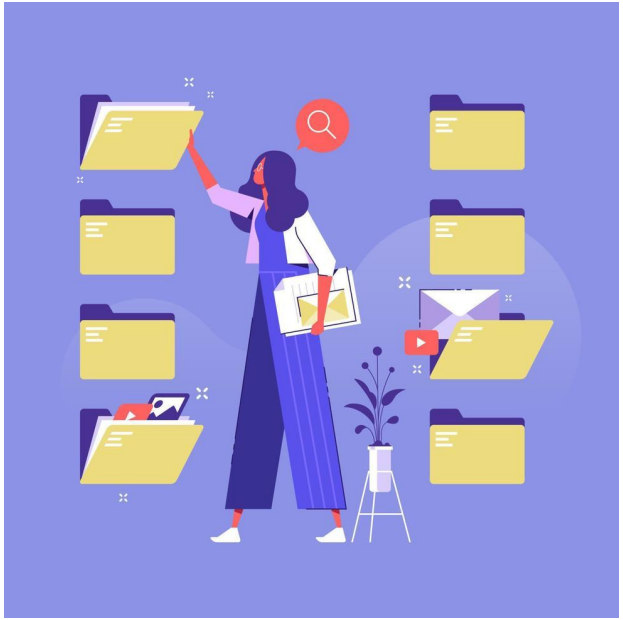



Many SE are being replaced by VectorDB but, we are not yet there in this lecture…

# Main operations performed by a SE

- Indexing a document
- Searching for documents
- Updating documents
- Deleting documents



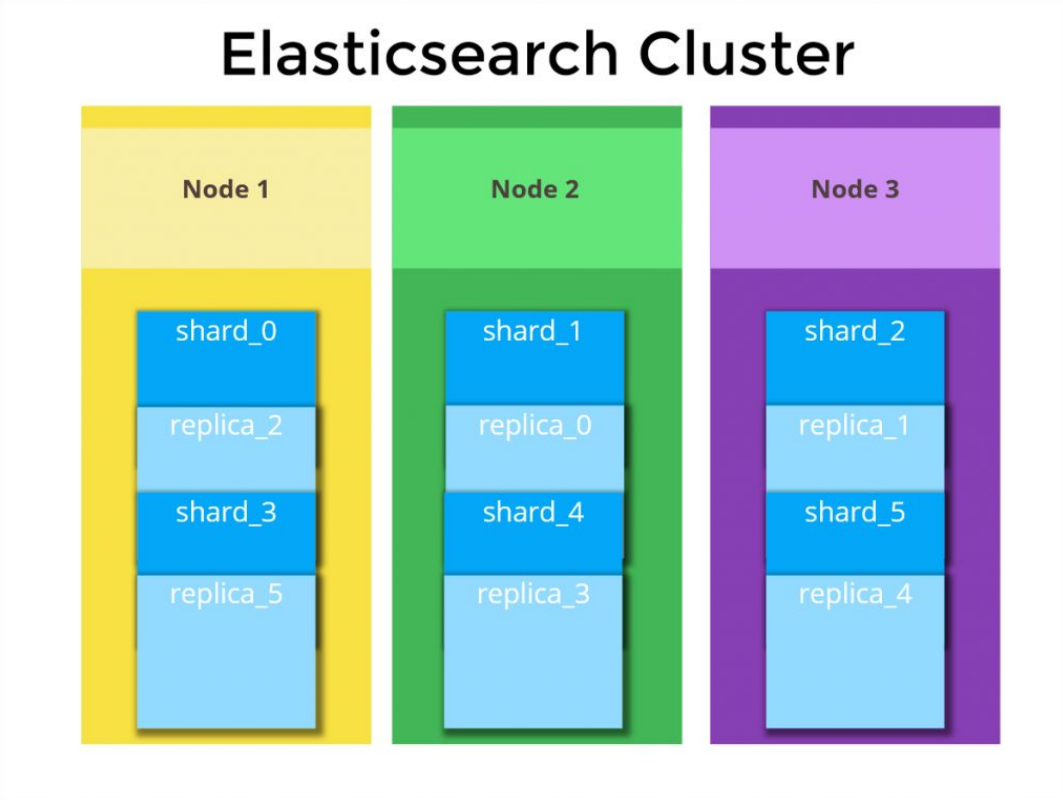
# Clusters and indexes on Elasticsearch

A cluster is a group of one or more connected node instances responsible for distributing tasks, searching, and indexing across all nodes

An Elasticsearch cluster can contain **multiple indexes**, which are databases similar to a relational database. These indexes contain multiple types (tables)

Types (tables) contain **multiple documents (records/rows)**, and these documents contain properties (columns)

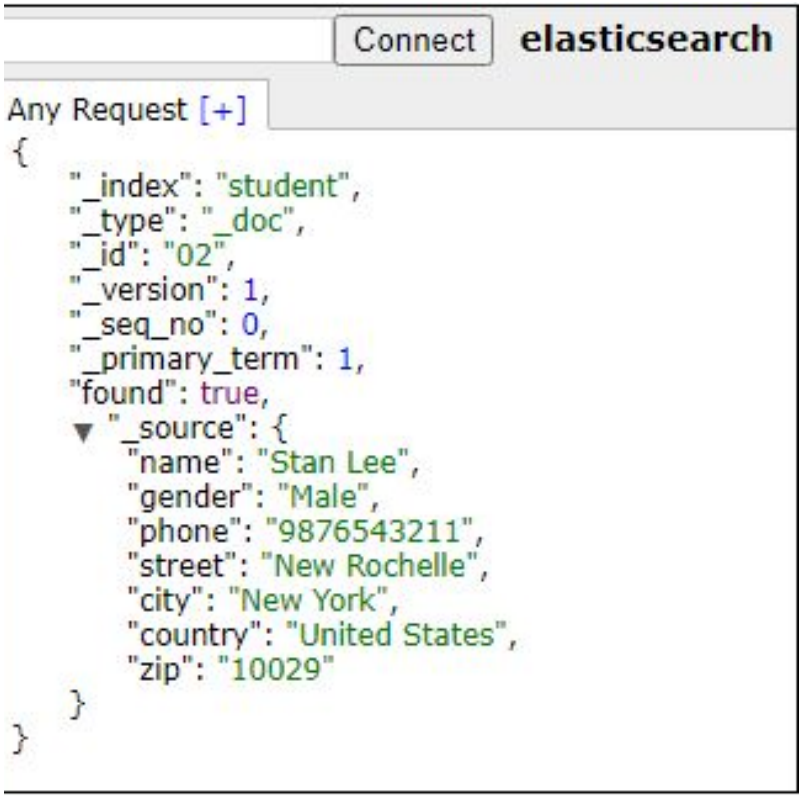
# Shards et replicas



# The document in the context of Elasticsearch

A document is a JSON document that is stored in Elasticsearch

A mapping defines how a document is indexed, how its fields are indexed and stored by Lucene



# Example (terminal)

# Installation

https://www.elastic.co/fr/downloads/elasticsearch

elasticsearch

## 1 Download and unzip Elasticsearch

**Choose platform:**

Linux x86_64

Linux x86_64 | sha | asc

**Package managers:**

yum, dnf, or zypper | apt-get

**Containers:**

Docker →

Elasticsearch can also be installed from our package repositories using apt or yum. See ***Repositories*** in the Guide.

## 2 Start Elasticsearch

Run `bin/elasticsearch` (or `bin\elasticsearch.bat` on Windows) to start Elasticsearch with security enabled.

## 3 Dive in

See our documentation for the latest up-to-date information.

# Indexation - Elasticsearch syntax

```
curl -X POST "localhost:9200/my-index-000001/_doc/?pretty" -H 'Content-Type: application/json' -d'
{
  "@timestamp": "2099-11-15T13:12:00",
  "message": "GET /search HTTP/1.1 200 1070000",
  "user": {
    "id": "kimchy"
  }
}
'
```

```
{
  "_index" : "my-index-000001",
  "_type" : "_doc",
  "_id" : "KVueLIYBiLBGst-w-BOy",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

# Indexing

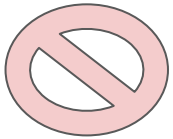
```
curl -X POST "http://localhost:9200/educative/_doc/?pretty" -H 'Content-Type: application/json' -d'
{ "articleName" : "elasticsearch-intro" }
'
```

```
curl -X POST "http://localhost:9200/_bulk?pretty" -H 'Content-Type: application/json' -d'
{ "index" : { "_index" : "educative"} }
{ "articleName" : "elasticsearch-intro" }
{ "index" : { "_index" : "educative"} }
{ "articleName" : "elasticsearch-insert-data" }
{ "index" : { "_index" : "educative"} }
{ "articleName" : "elasticsearch-query-data" }
'
```

```
curl -s -H "Content-Type: application/x-ndjson" -XPOST http://localhost:9200/_bulk --data-binary "@data.txt"
```

# Searching

curl -X GET "localhost:9200/my-index-000001/_doc/0?_source=false&pretty"



curl -X GET "localhost:9200/my-index-000001/_doc/KVueLIYBiLBGst-w-BOy?_source=false&pretty"

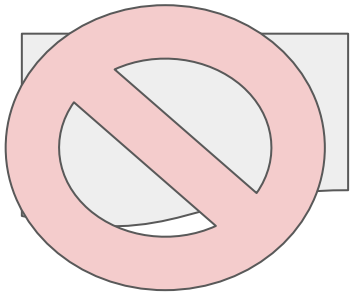
KVueLIYBiLB
Gst-w-BOy

```
curl -X GET "localhost:9200/_search?pretty" -H 'Content-Type: application/json' -d'
{
  "query": {
    "query_string": {
      "query": "content:Je cherche quelque chose"
    }
  }
}
'
```

```
curl -X GET "localhost:9200/_search?pretty" -H 'Content-Type: application/json' -d'
{
  "query": {
    "query_string": {
      "query": "(content:this OR name:this) AND (content:that OR name:that)"
    }
  }
}
'
```

# Deleting

```
curl -X DELETE
"localhost:9200/my-index-000001/_doc/1?pretty"
```



# Updating

```
curl -X POST "localhost:9200/my-index-000001/_update/1?pretty"
-H 'Content-Type: application/json' -d'
{
  "script" : {
    "source": "ctx._source.counter += params.count",
    "lang": "painless",
    "params" : {
      "count" : 4
    }
  }
}
'
```

of course, there are Python libraries...

# Python for elasticsearch

- elasticsearch-py (official)
- cherche (developed by Renault in collaboration with UT)
- Haystack
- …

Haystack is an end-to-end framework that allows you to build powerful, production-ready pipelines for various information retrieval use cases

It is useful for performing tasks such as question answering (QA) or semantic document search

Haystack is built in a modular way so that you can combine with other open source projects, such as Hugging Face transformers, Elasticsearch, or Milvus

Used by Airbus, Alcatel-Lucent, BetterUp, Deepset, Etalab, Infineon, Sooth.ai…

# Example haystack

# If you have a text collection

```
1 wksimpledf = wksimpledf.sample(10000)
2 wksimpledf
```

| | id | url | title | text |
|---|---|---|---|---|
| **120028** | 549186 | https://simple.wikipedia.org/wiki/R%C3%BCschegg | Rüschegg | Rüschegg is a municipality in the administrative district of Bern-Mittelland... |
| **5910** | 19138 | https://simple.wikipedia.org/wiki/The%20X-Files | The X-Files | The X-Files is an American science fiction television series set in Maryland... |
| **190535** | 852833 | https://simple.wikipedia.org/wiki/Island%20of%20Death | Island of Death | Island of Death (Greek: Τα Παιδιά Του Διαβόλου, Ta pediá tou Diavólou, ), al... |
| **192318** | 858473 | https://simple.wikipedia.org/wiki/Montano%20Lucino | Montano Lucino | Montano Lucino is a comune in the Province of Como in the Italian region of ... |
| **71004** | 301066 | https://simple.wikipedia.org/wiki/Mis%20Mejores%20Canciones%20-%2017%20Super... | Mis Mejores Canciones - 17 Super Éxitos | Mis Mejores Canciones - 17 Super Exitos (English: My Best Songs - 17 Super H... |
| **...** | ... | ... | ... | ... |
| **3639** | 11039 | https://simple.wikipedia.org/wiki/Lady%20Frances%20Brandon | Lady Frances Brandon | Lady Frances Brandon (16 July 1517 – 20 November 1559) was the daughter of M... |
| **22210** | 84252 | https://simple.wikipedia.org/wiki/Fregi%C3%A9court | Fregiécourt | Fregiécourt was a municipality, in the new municipality of La Baroche and th... |
| **170356** | 784692 | https://simple.wikipedia.org/wiki/Ron%20Tudor | Ron Tudor | Ronald Stewart Tudor MBE (18 May 1924 – 21 August 2020) was an Australian pr... |
| **44724** | 157991 | https://simple.wikipedia.org/wiki/Dino%20Baggio | Dino Baggio | Dino Baggio (born 24 July 1971) is a former Italian football player. He has ... |
| **50911** | 185244 | https://simple.wikipedia.org/wiki/Cotyledon | Cotyledon | A cotyledon, or seed leaf, is a leaf that is stored in a seed. When the seed... |

10000 rows × 4 columns

# Indexing the collection

```python
import os
from haystack.document_stores import ElasticsearchDocumentStore

# Get the host where Elasticsearch is running, default to localhost
host = os.environ.get("ELASTICSEARCH_HOST", "localhost")
document_store = ElasticsearchDocumentStore(host=host, username="", password="", index="document")
```

```python
docs = [{"content": row["text"], "id": row["id"],
             "meta":{"item_id": row["id"], "url": row["url"], "title": row["title"]}}
         for  ,row in wksimpledf.iterrows()]
document_store.write_documents(documents=docs, index="document")

print(f"Loaded {document_store.get_document_count()} documents")
```

The documents are indexed and everything is ready for searching

# Searching

```python
from haystack.nodes.retriever import BM25Retriever

bm25_retriever = BM25Retriever(document_store=document_store)
```

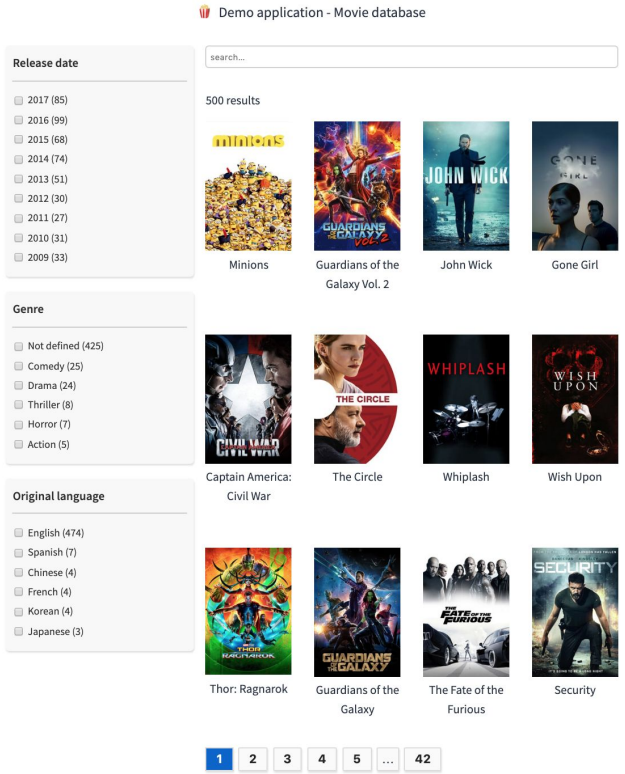
```python
query = "toulouse"
retrieved_docs = bm25_retriever.retrieve(
    query=query, top_k=10)
```

```python
pd.DataFrame(retrieved_docs)
```

| | id | content | content_type | meta | id_hash_keys | score | embedding |
|---|---|---|---|---|---|---|---|
| 0 | 825253 | Aeroscopia is a museum in Blagnac, France, close to Toulouse. It was create... | text | {'item_id': '825253', 'title': 'Aeroscopia', 'url': 'https://simple.wikipedi... | [content] | 0.802462 | None |
| 1 | 98186 | Comte Henri Marie Raymond de Toulouse-Lautrec-Monfa (24 November 1864 – 9 Se... | text | {'item_id': '98186', 'title': 'Henri de Toulouse-Lautrec', 'url': 'https://s... | [content] | 0.798555 | None |
| 2 | 347952 | Jean-Michel Vernhes (born in 1950 at Mazamet) is a French aerospace engineer... | text | {'item_id': '347952', 'title': 'Jean-Michel Vernhes', 'url': 'https://simple... | [content] | 0.759506 | None |
| 3 | 342892 | The Groupement des écoles d'aéronautique (GEA France) (in English French avi... | text | {'item_id': '342892', 'title': 'Groupement des écoles d'aéronautique', 'url'... | [content] | 0.748334 | None |
| 4 | 22060 | Casablanca (classical Arabic name: الدار البيضاء, "the white house"; Spanish... | text | {'item_id': '22060', 'title': 'Casablanca', 'url': 'https://simple.wikipedia... | [content] | 0.745086 | None |
| 5 | 539261 | Kerima (born February 10, 1925) is a French movie actress. She was best know... | text | {'item_id': '539261', 'title': 'Kerima (actress)', 'url': 'https://simple.wi... | [content] | 0.727866 | None |
| 6 | 537308 | Marie Thérèse Félicité d'Este (; October 6, 1726 – April 30, 1754) was born ... | text | {'item_id': '537308', 'title': 'Maria Teresa Felicitas d'Este', 'url': 'http... | [content] | 0.699737 | None |
| 7 | 410005 | The European Rugby Champions Cup is a major European rugby competition invol... | text | {'item_id': '410005', 'title': 'European Rugby Champions Cup', 'url': 'https... | [content] | 0.699335 | None |
| 8 | 453216 | Fernando Lúcio da Costa (18 March 1978 – 7 June 2014), better known as Ferna... | text | {'item_id': '453216', 'title': 'Fernandão', 'url': 'https://simple.wikipedia... | [content] | 0.689056 | None |
| 9 | 342523 | \nYear 1249 (MCCXLIX) was a common year starting on Friday of the Julian cal... | text | {'item_id': '342523', 'title': '1249', 'url': 'https://simple.wikipedia.org/... | [content] | 0.686201 | None |

# More useful example…

- The example can easily be transformed into a question-answer problem
- Re-ranking using transformer-based models can be easily configured
- There are web interfaces for Elasticsearch, such as React Elasticsearch, elastic/search-ui, but Kibana is its most professional web component.



and for research experiments?

# Terrier IR

Terrier IR is a modular, open-source software package for the rapid development of large-scale information retrieval applications

Terrier was developed by members of the Information Retrieval Research Group in the Department of Computer Science at the University of Glasgow

Terrier is written in Java


# pyTerrier

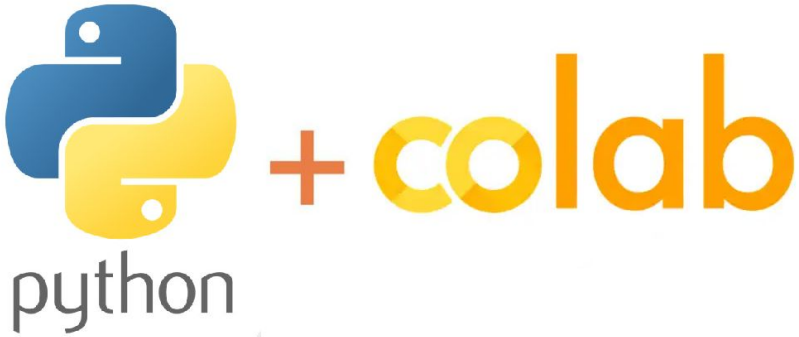
Similarly than for Elasticsearch, using a Python library can simplify the various steps.

We will use pyTerrier, a grapper for Terrier.

Using pyTerrier will simplify interactions with Terrier, but Terrier is still used, which means that Java will still be used (same concept for Elasticsearch).

The main advantage is easy integration with other software (such as haystack) or modules for using recent models, using annotated collections, and evaluation.

# Practical example




# Installation and configuration

```
# déclaration de la variable JAVA_HOME
import os
os.environ['JAVA_HOME'] = '/usr/lib/jvm/java-11-openjdk-amd64'
!export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

```
#installation de pyterrier avec pip
!pip install --upgrade git+https://github.com/terrier-org/pyterrier.git#egg=python-terrier
```

Installation, importing libraries, and creating the object for indexing

```
#Initialization de JVM
import pyterrier as pt
if not pt.started():
  pt.init()
```

```
terrier-assemblies 5.7 jar-with-dependencies not foun
Done
terrier-python-helper 0.0.7 jar not found, downloadin
Done
PyTerrier 0.9.2 has loaded Terrier 5.7 (built by crai
```

```
#Création du dossier pour le stockage des indexes
!rm -rf ./pd_index
pd_indexer = pt.DFIndexer("./pd_index")
```

# Downloading a text collection - Wikipedia

```
1 import pandas as pd
2 from datasets import load_dataset
```

```
1 wksimple = load_dataset("wikipedia", "20220301.simple")
2 df = pd.DataFrame(wksimple['train'])
3 df.columns = ['docno','url','title','text']
4 df
```

```
WARNING:datasets.builder:Found cached dataset wikipedia (/root/.cache/huggingface/datasets/wikipedia/20220301.simple/2.0.0/aa542ed919c
100% 1/1 [00:00<00:00, 26.04it/s]
```

| | docno | url | title | text |
|---|---|---|---|---|
| **0** | 1 | https://simple.wikipedia.org/wiki/April | April | April is the fourth month of the year in the J... |
| **1** | 2 | https://simple.wikipedia.org/wiki/August | August | August (Aug.) is the eighth month of the year ... |
| **2** | 6 | https://simple.wikipedia.org/wiki/Art | Art | Art is a creative activity that expresses imag... |
| **3** | 8 | https://simple.wikipedia.org/wiki/A | A | A or a is the first letter of the English alph... |
| **4** | 9 | https://simple.wikipedia.org/wiki/Air | Air | Air refers to the Earth's atmosphere. Air is a... |
| **...** | ... | ... | ... | ... |
| **205323** | 910281 | https://simple.wikipedia.org/wiki/Noticiero%20... | Noticiero Univision | Noticiero Univision is the flagship daily even... |
| **205324** | 910287 | https://simple.wikipedia.org/wiki/Bachhan%20Pa... | Bachhan Paandey | Bachchhan Paandey is an upcoming Indian Hindi-... |
| **205325** | 910294 | https://simple.wikipedia.org/wiki/Repdigit | Repdigit | In recreational math, a repdigit or a monodigi... |
| **205326** | 910309 | https://simple.wikipedia.org/wiki/Lady%20in%20... | Lady in a Cage | Lady in a Cage is a 1964 American psychologica... |
| **205327** | 910312 | https://simple.wikipedia.org/wiki/Noah%20flood... | Noah flood and Nakhchivan | Nakhchivan is one of the oldest cities in Azer... |

205328 rows × 4 columns

# Indexing and searching - toy example

```
1 indexref2 = pd_indexer.index(df["text"], df["docno"], df["url"], df["title"])
```

create the files in the pd_index folder

```
1 pt.BatchRetrieve(indexref2, wmodel="TF_IDF", metadata=["docno","title","url"]).search("france")
```

searching for “france” on the collection

but there are other options, such as Anserini…

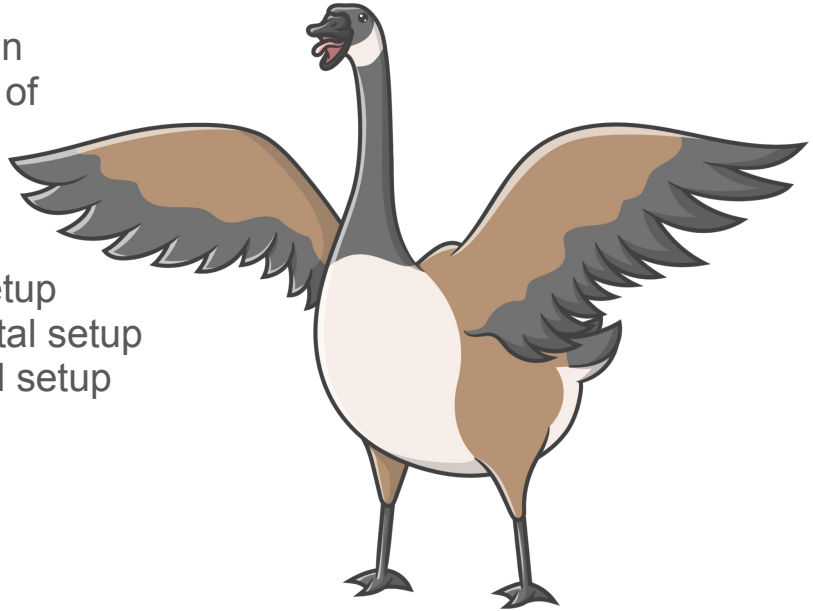
# Anserini

Anserini is a toolkit for reproducible information retrieval research.

By building on **Lucene**, we aim to bridge the gap between academic information retrieval research and the practice of building real-world search applications.

Anserini is a great IR tool for:

- **Repeatability** = same team, same experimental setup
- **Reproducibility** = different team, same experimental setup
- Replicability = different team, different experimental setup



Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the Use of Lucene for Information Retrieval Research. SIGIR 2017.

but IR is nothing without datasets…isn’t it?

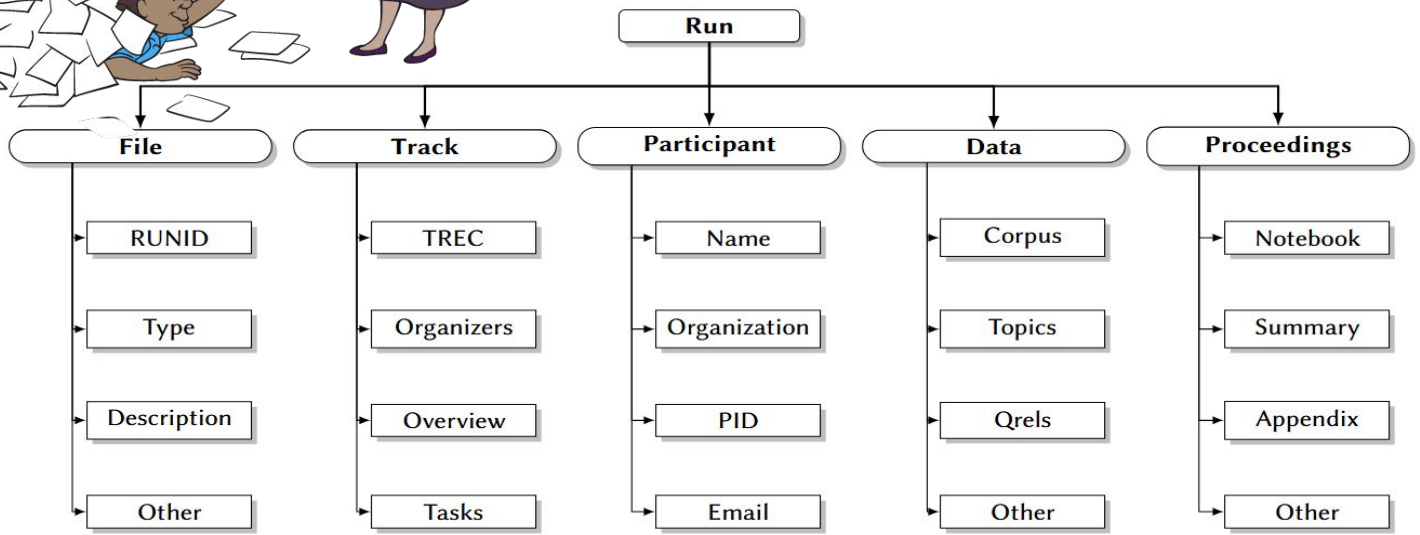
# Datasets for IR

# TREC

The Text Retrieval Conference

... encouraging research in information retrieval from large text collections.

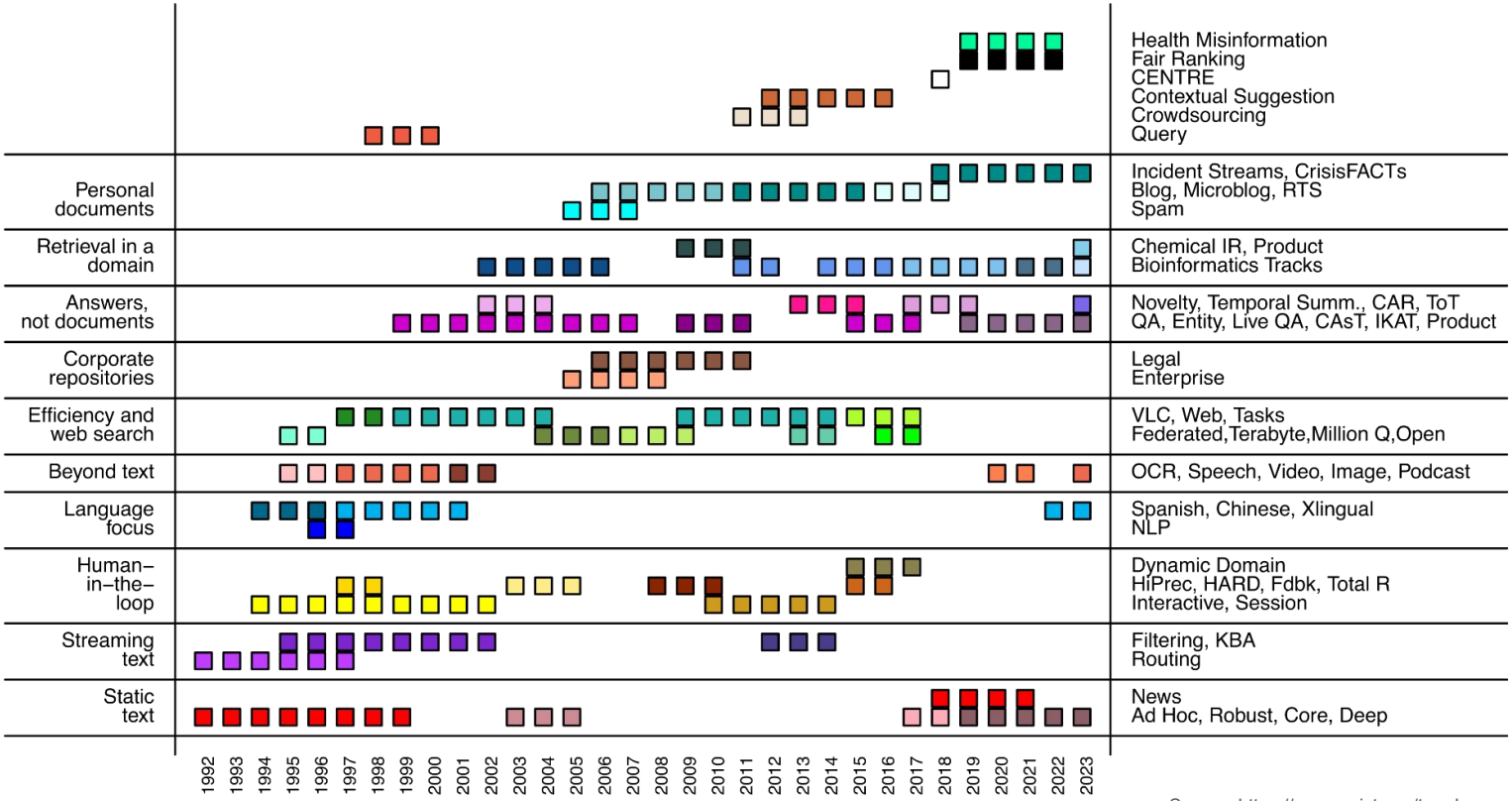
TREC is an evaluation workshop series for measuring the effectiveness of search algorithms and other technologies that help us find information.






Breuer, T., Voorhees, E. M., & Soboroff, I. Browsing and searching metadata of TREC. SIGIR2024

# TREC Browser




Source: https://pages.nist.gov/trec-browser/

# BEIR

BioASD
14.91M

MsMARCO
8.8M

FiQA
57k

NFCorpus
3.5k

**NanoBEIR** 🍺: A collection of smaller versions of BEIR datasets with 50 queries and up to 10K documents each.

| Dataset | Website | BEIR-Name | Public? | Type | Queries | Corpus | Rel D/Q | Down-load |
|---|---|---|---|---|---|---|---|---|
| MSMARCO | Homepage | msmarco | ✅ | train dev test | 6,980 | 8.84M | 1.1 | Link |
| TREC-COVID | Homepage | trec-covid | ✅ | test | 50 | 171K | 493.5 | Link |
| NFCorpus | Homepage | nfcorpus | ✅ | train dev test | 323 | 3.6K | 38.2 | Link |
| BioASQ | Homepage | bioasq | ❌ | train test | 500 | 14.91M | 4.7 | No |
| NQ | Homepage | nq | ✅ | train test | 3,452 | 2.68M | 1.2 | Link |
| HotpotQA | Homepage | hotpotqa | ✅ | train dev test | 7,405 | 5.23M | 2.0 | Link |
| FiQA-2018 | Homepage | fiqa | ✅ | train dev test | 648 | 57K | 2.6 | Link |
| Signal-1M(RT) | Homepage | signal1m | ❌ | test | 97 | 2.86M | 19.6 | No |
| TREC-NEWS | Homepage | trec-news | ❌ | test | 57 | 595K | 19.6 | No |
| Robust04 | Homepage | robust04 | ❌ | test | 249 | 528K | 69.9 | No |
| ArguAna | Homepage | arguana | ✅ | test | 1,406 | 8.67K | 1.0 | Link |
| Touche-2020 | Homepage | webis-touche2020 | ✅ | test | 49 | 382K | 19.0 | Link |
| CQADupstack | Homepage | cqadupstack | ✅ | test | 13,145 | 457K | 1.4 | Link |
| Quora | Homepage | quora | ✅ | dev test | 10,000 | 523K | 1.6 | Link |
| DBPedia | Homepage | dbpedia-entity | ✅ | dev test | 400 | 4.63M | 38.2 | Link |
| SCIDOCS | Homepage | scidocs | ✅ | test | 1,000 | 25K | 4.9 | Link |
| FEVER | Homepage | fever | ✅ | train dev test | 6,666 | 5.42M | 1.2 | Link |
| Climate-FEVER | Homepage | climate-fever | ✅ | test | 1,535 | 5.42M | 3.0 | Link |
| SciFact | Homepage | scifact | ✅ | train test | 300 | 5K | 1.1 | Link |

Thakur, Nandan, et al. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. NeurIPS 2021.

# ir_datasets

## ir_datasets: Catalog

`ir_datasets` provides a common interface to many IR ranking datasets.

### Getting Started

Install with pip:

```
pip install --upgrade ir_datasets
```

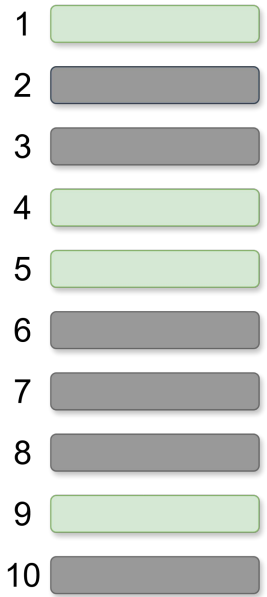
Guides:

- Colab Tutorials: python, CLI
- Python API Documentation (beta version)
- CLI Documentation
- Download Dashboard
- Dataset Counts
- Adding new datasets
- ir_datasets SIGIR resource paper
- Using `ir_datasets` with... PyTerrier · ir-measures · trec_eval · Experimaestro
- Design Documentation

| Dataset | docs | queries | qrels | /q | scoreddocs | /q | docpairs | /q | qlogs |
|---|---|---|---|---|---|---|---|---|---|
| mmarco/v2/fr/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/fr/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 6.8M | 978.8 | | | |
| mmarco/v2/fr/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/hi | 8.8M | | | | | | | | |
| mmarco/v2/hi/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/hi/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 7.0M | 997.4 | | | |
| mmarco/v2/hi/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/id | 8.8M | | | | | | | | |
| mmarco/v2/id/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/id/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 6.8M | 973.0 | | | |
| mmarco/v2/id/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/it | 8.8M | | | | | | | | |
| mmarco/v2/it/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/it/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 7.0M | 996.1 | | | |
| mmarco/v2/it/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/ja | 8.8M | | | | | | | | |
| mmarco/v2/ja/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/ja/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 6.8M | 976.7 | | | |
| mmarco/v2/ja/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/pt | 8.8M | | | | | | | | |
| mmarco/v2/pt/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/pt/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 7.0M | 999.3 | | | |
| mmarco/v2/pt/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/ru | 8.8M | | | | | | | | |
| mmarco/v2/ru/dev | 8.8M | 101K | 59K | 0.6 | | | | | |
| mmarco/v2/ru/dev/small | 8.8M | 7.0K | 7.4K | 1.1 | 6.9M | 993.1 | | | |
| mmarco/v2/ru/train | 8.8M | 809K | 533K | 0.7 | | | 40M | 49.2 | |
| mmarco/v2/vi | 8.8M | | | | | | | | |
| mmarco/v2/vi/dev | 8.8M | 101K | 59K | 0.6 | | | | | |

and what about metrics???
Let’s briefly talk about metrics for IR

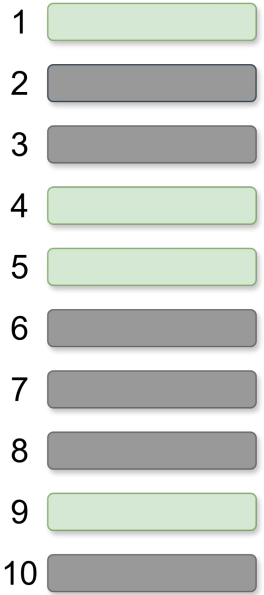
# QRELs: a core evaluation file in IR



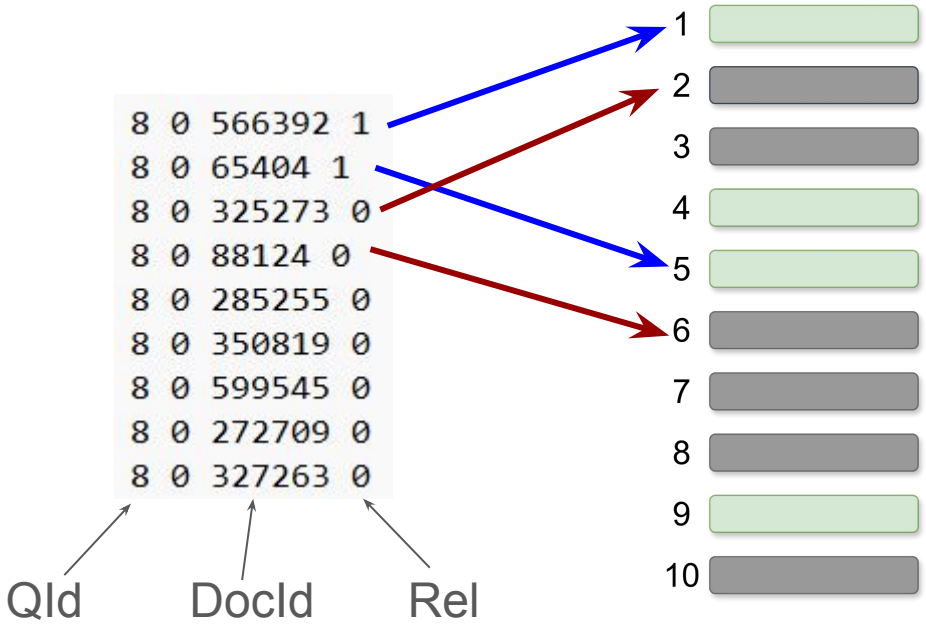
# QRELs: a core evaluation file in IR

```
8 0 566392 1
8 0 65404 1
8 0 325273 0
8 0 88124 0
8 0 285255 0
8 0 350819 0
8 0 599545 0
8 0 272709 0
8 0 327263 0
```

QId DocId Rel

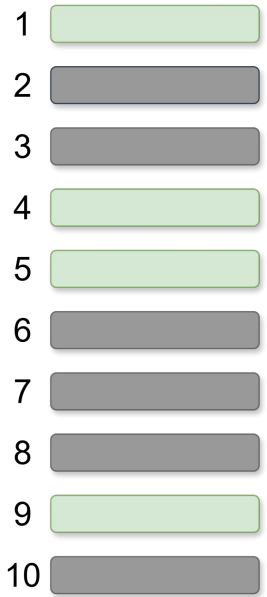


# QRELs: a core evaluation file in IR



# Common IR metrics

| Measure | Name/Description |
|---|---|
| (M)AP | (Mean) Average Precision |
| Bpref | Binary Preference |
| ERR | Expected Reciprocal Rank |
| infAP | Inferred Average Precision |
| IPrec@r | Interpolated Precision at r |
| Judged@k | Judgement rate at cutoff k |
| nDCG | Normalised Discounted Cumulative Gain |
| NumQ | Number of queries |
| NumRel | Number of relevant documents (from qrels) |
| NumRet | Number of results returned |
| P(recision)@k | Precision at cutoff k |
| R(ecall)@k | Recall at cutoff k |
| RBP | Rank Biased Precision |
| Rprec | Precision at R (total number of rel docs) |
| (M)RR | (Mean) Reciprocal Rank |
| SetAP | Unranked Set Average Precision |
| SetF | Set F-measure |
| SetP | Set Precision |
| SetR | Set Recall |
| Success@k | Relevant document found in top k |




$$\text{nDCG}_\text{k} = \frac{\text{DCG}_\text{k}}{\text{IDCG}_\text{k}}$$

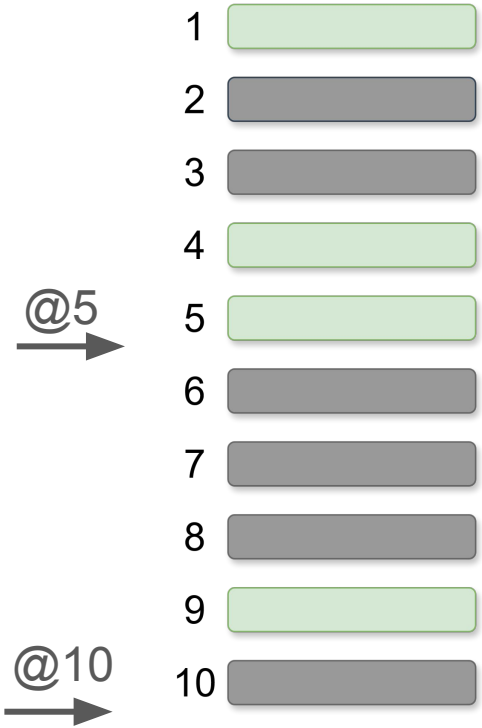
$$\text{DCG}_\text{k} = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)} \qquad \text{IDCG}_\text{k} = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}$$

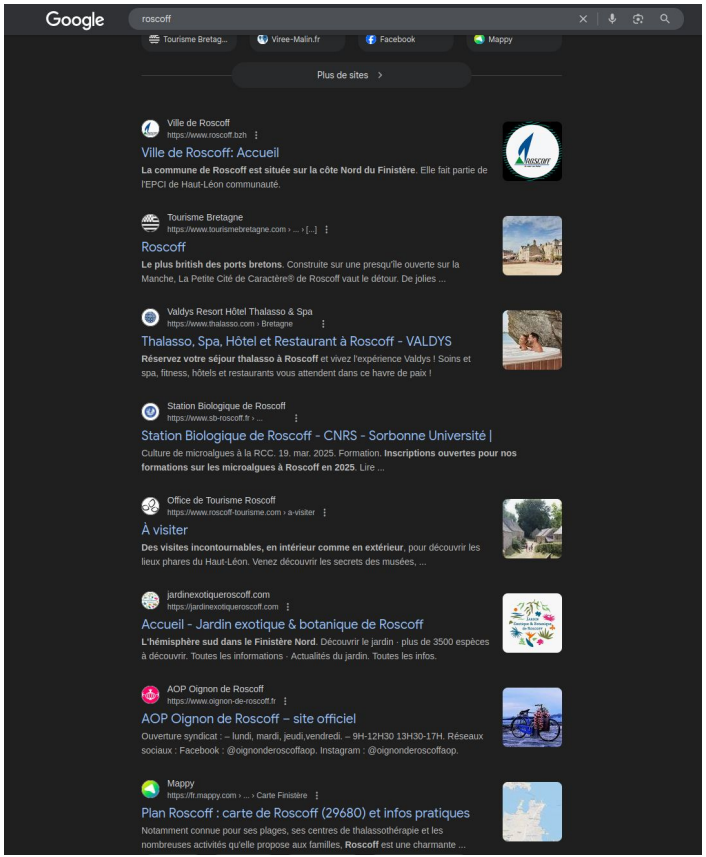
$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

MacDonald et al. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. CIKM2020.

# Is the top-k relevant?



# trec_eval

```
8 0 566392 1
8 0 65404 1
8 0 325273 0
8 0 88124 0
8 0 285255 0
8 0 350819 0
8 0 599545 0
8 0 272709 0
8 0 327263 0
```

1 2 3 4 5 6 7 8 9 10

1 2 3 4 5 6 7 8 9 10

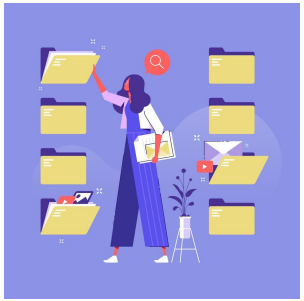
```
recall_1000             all     0.8578
ndcg_cut_10             all     0.3801
```

# **A naive** (*but relevant*) **IR experiment**

# Inputs

## Collection

## IR system (Representation Model)



## Queries

## QREL

```
8 0 566392 1
8 0 65404 1
8 0 325273 0
8 0 88124 0
8 0 285255 0
8 0 350819 0
8 0 599545 0
8 0 272709 0
8 0 327263 0
```

# Inputs

Collection

**FiQA**

Queries

IR system (Representation Model)


TF IDF

QREL

# Reading the corpus

```
1 dfcorpus = pd.DataFrame(corpus)
2 dfcorpus
```

| | _id | title | text |
|---|---|---|---|
| **0** | 3 | | I'm not saying I don't like the idea of on-the... |
| **1** | 31 | | So nothing preventing false ratings besides ad... |
| **2** | 56 | | You can never use a health FSA for individual ... |
| **3** | 59 | | Samsung created the LCD and other flat screen ... |
| **4** | 63 | | Here are the SEC requirements: The federal sec... |
| **...** | ... | ... | ... |
| **57633** | 599946 | | >Well, first off, the roads are more than j... |
| **57634** | 599953 | | Yes they do. There are billions and billions s... |
| **57635** | 599966 | | >It's biggly sad you don't understand human... |
| **57636** | 599975 | | "Did your CTO let a major group use ""admin/ad... |
| **57637** | 599987 | | Giving the government more control over the di... |

57638 rows × 3 columns

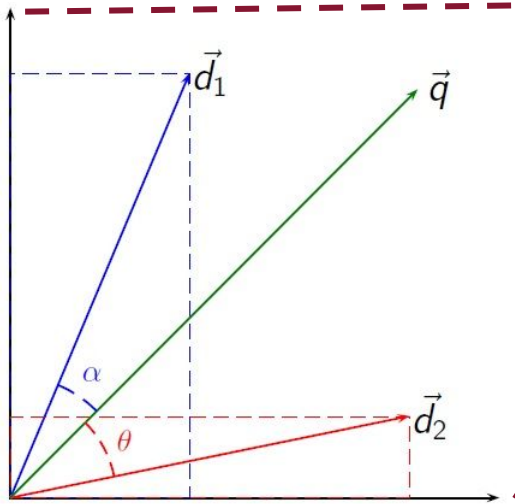
# Creating a vocabulary or a vector space

```
1 vectorizer = TfidfVectorizer(stop_words='english')
```

```
1 vectorizer.fit(dfcorpus['text'].values)
```

TfidfVectorizer

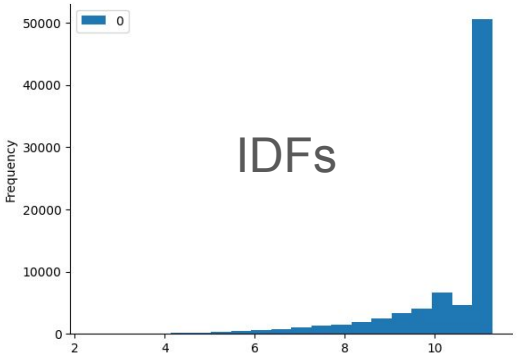
TfidfVectorizer(stop_words='english')



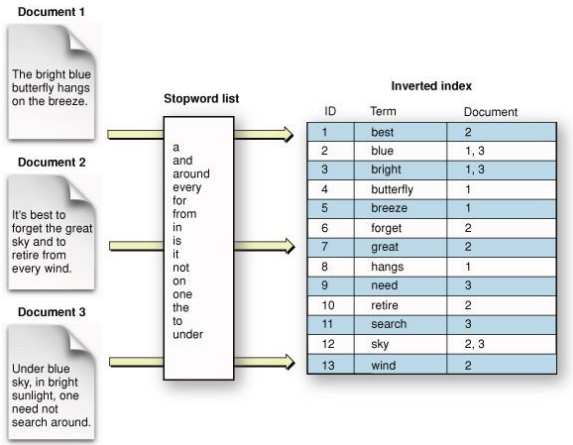

```
1 pd.DataFrame.from_dict(vectorizer.vocabulary_, orient='index')
```

| | 0 |
|---|---|
| saying | 61954 |
| don | 25494 |
| like | 43376 |
| idea | 37242 |
| job | 40622 |
| ... | ... |
| fraternal | 31740 |
| elks | 27200 |
| thelocal | 69650 |
| 20170829 | 2737 |
| denmarks | 23459 |

80004 rows × 1 columns



# Creating a corpus representation




```
1 X = csc_matrix(vectorizer.transform(dfcorpus['text'].values)).T
2 X.shape
```

```
(80004, 57638)
```

# Querying the corpus

```
similarities = vectorizer.transform([query]).dot(X)
dfsim = pd.DataFrame({'_id':list(dfcorpus['_id'].values),'sim':np.array(similarities.todense())[0], 'text':list(dfcorpus['text'].values)})
dfsim.sort_values('sim', ascending=False).head(1000)
```

| | _id | sim | text |
|---|---|---|---|
| **46858** | 486795 | 0.504381 | "> The true problem here is that, just like... |
| **27149** | 280368 | 0.338802 | Estos conceptos que son muy importantes saber,... |
| **23944** | 248167 | 0.330911 | France taxes capital / dividend gains accrued ... |
| **24159** | 250281 | 0.326717 | Al igual que la fidelidad del cliente, la "lea... |
| **22065** | 229229 | 0.318072 | La empresa ofrece los mejores servicios de vis... |
| ... | ... | ... | ... |
| **28474** | 294150 | 0.031995 | There is no right and wrong answer to this que... |
| **51768** | 538278 | 0.031984 | "A title such as ""5% Treasury Gilt 2020"" exp... |
| **29471** | 304376 | 0.031977 | You will not benefit at tax time like you did ... |
| **8799** | 90916 | 0.031910 | As my business partner likes to say, *don't ba... |
| **8378** | 86621 | 0.031815 | Residents of Canada must pay Canadian income t... |

1000 rows × 3 columns

```
query ="Quelle est la capital de la France?"
print(vectorizer.transform([query]))

```

```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 4 stored elements and shape (1, 80004)>
  Coords        Values
  (0, 16909)    0.23053358717572633
  (0, 28368)    0.4796207617957178
  (0, 31692)    0.3905704805875928
  (0, 42193)    0.7511809966525803
```

top 1000 documents!

# Evaluation our model - Anserini tools

```
! ./anserini-tools/eval/trec_eval.9.0.4/trec_eval -c -m ndcg_cut.10 -m recall anserini-tools/topics-and-qrels/qrels.beir-v1.0.0-fiqa.test.txt run_tfidfvectorizer.txt
```

```
recall_5                all     0.1547
recall_10               all     0.1985
recall_15               all     0.2425
recall_20               all     0.2654
recall_30               all     0.3080
recall_100              all     0.4515
recall_200              all     0.5331
recall_500              all     0.6286
recall_1000             all     0.7075
ndcg_cut_10             all     0.1553
```

ndcg@10=0.1553 and R@1000=0.7075

*Disclaimer: Performances are not excellent but everything is “pure” python, code is simple, indexing is online (usually offline), and it is quite fast*

# Evaluation our model - Anserini tools

```
! ./anserini-tools/eval/trec_eval.9.0.4/trec_eval -c -m ndcg_cut.10 -m recall anserini-tools/topics-and-qrels/qrels.beir-v1.0.0-fiqa.test.txt run_tfidfvectorizer.txt
```

```
recall_5                all     0.1547
recall_10               all     0.1985
recall_15               all     0.2425
recall_20               all     0.2654
recall_30               all     0.3080
recall_100              all     0.4515
recall_200              all     0.5331
recall_500              all     0.6286
recall_1000             all     0.7075
ndcg_cut_10             all     0.1553
```

| Model (→) | Lexical | Sparse | | |
|---|---|---|---|---|
| **Dataset (↓)** | **BM25** | **DeepCT** | **SPARTA** | **docT5query** |
| MS MARCO | 0.228 | 0.296‡ | 0.351‡ | 0.338‡ |
| TREC-COVID | 0.656 | 0.406 | 0.538 | 0.713 |
| BioASQ | 0.465 | 0.407 | 0.351 | 0.431 |
| NFCorpus | 0.325 | 0.283 | 0.301 | 0.328 |
| NQ | 0.329 | 0.188 | 0.398 | 0.399 |
| HotpotQA | 0.603 | 0.503 | 0.492 | 0.580 |
| FiQA-2018 | 0.236 | 0.191 | 0.198 | 0.291 |

ndcg@10=0.1553 and R@1000=0.7075

*Disclaimer: Performances are not excellent but everything is “pure” python, code is simple, indexing is online (usually offline), and it is quite fast*

# Evaluation our model - Anserini tools

```
1 ! ./anserini-tools/eval/trec_eval.9.0.4/trec_eval -c -m ndcg_cut.10 -m recall anserini-tools/topics-and-qrels/qrels.beir-v1.0.0-fiqa.test.txt run_tfidfvectorizer.txt
```

```
recall_5                all     0.1547
recall_10               all     0.1985
recall_15               all     0.2425
recall_20               all     0.2654
recall_30               all     0.3080
recall_100              all     0.4515
recall_200              all     0.5331
recall_500              all     0.6286
recall_1000             all     0.7075
ndcg_cut_10             all     0.1553
```

| Model (→) | Lexical | Sparse | | |
|---|---|---|---|---|
| **Dataset (↓)** | **BM25** | **DeepCT** | **SPARTA** | **docT5query** |
| MS MARCO | 0.228 | 0.296‡ | 0.351‡ | 0.338‡ |
| TREC-COVID | 0.656 | 0.406 | 0.538 | 0.713 |
| BioASQ | 0.465 | 0.407 | 0.351 | 0.431 |
| NFCorpus | 0.325 | 0.283 | 0.301 | 0.328 |
| NQ | 0.329 | 0.188 | 0.398 | 0.399 |
| HotpotQA | 0.603 | 0.503 | 0.492 | 0.580 |
| FiQA-2018 | 0.236 | 0.191 | 0.198 | 0.291 |

ndcg@10=0.1553 and R@1000=0.7075

https://tinyurl.com/RAGETALTP1

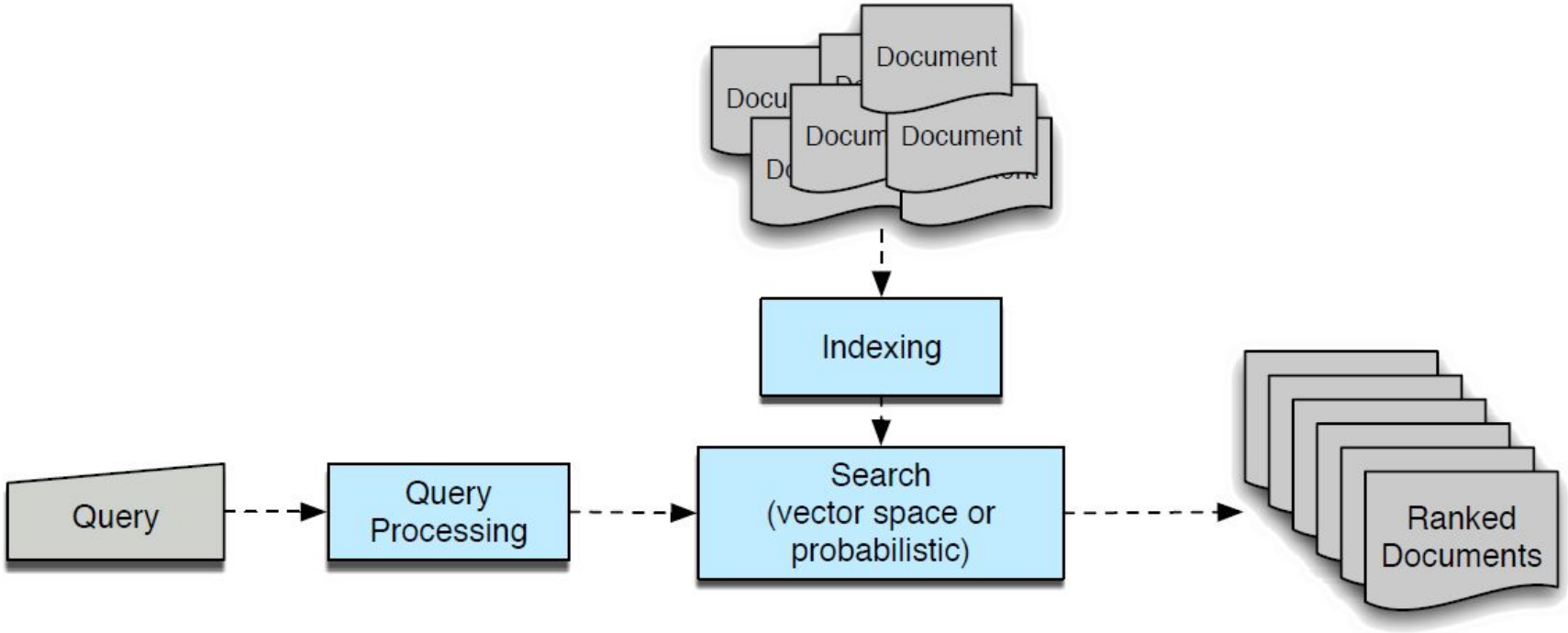
*Disclaimer: Performances are not excellent but everything is "pure" python, code is simple, indexing is online (usually offline), and it is quite fast*

# DL in IR

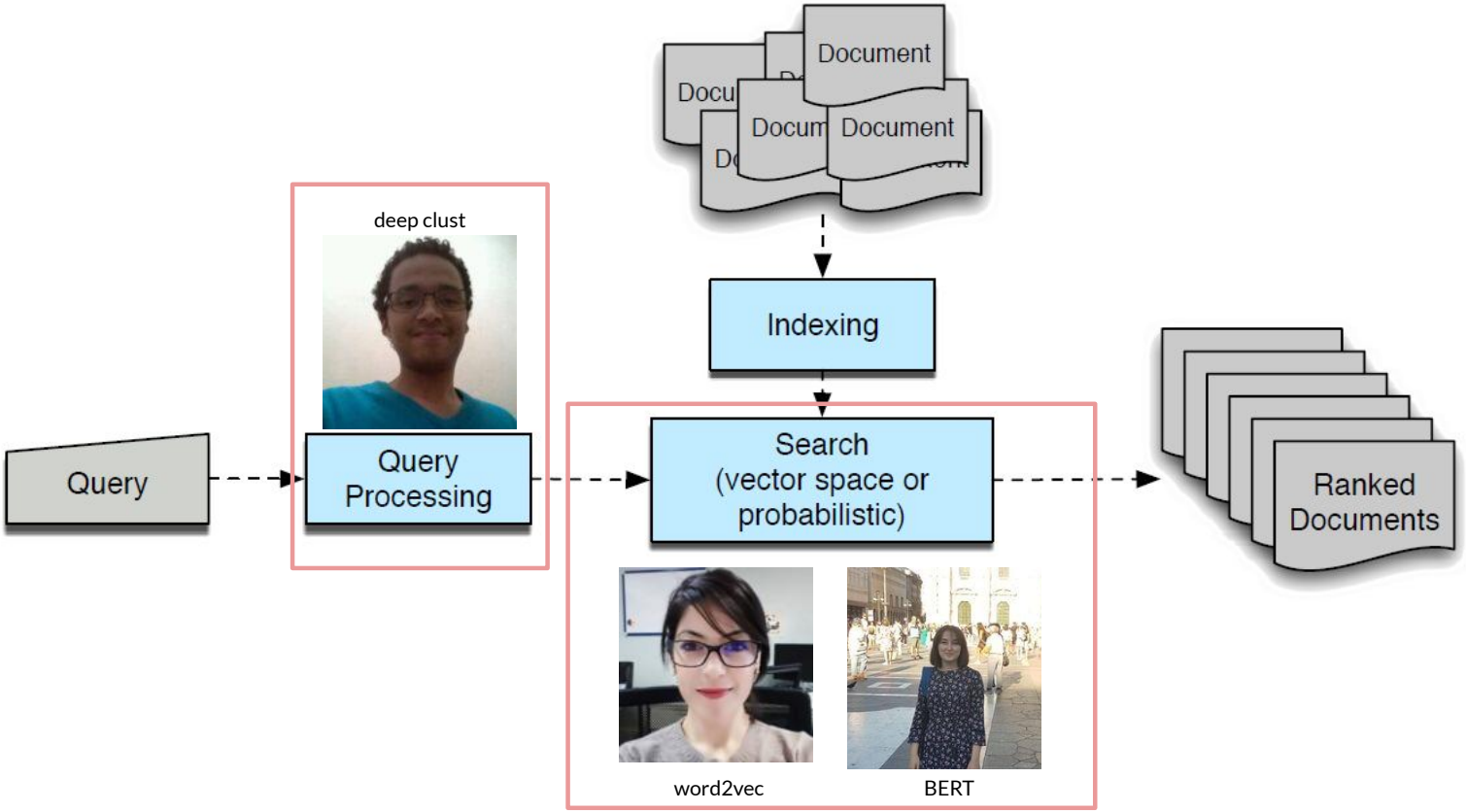
# DL in IR

A look at recent advances in IR…just a but more than 10 years back

# Where we can do DL in IR?



# Where we can do DL in IR?



# CONTEXT

- The bag of words (BoW) representation and exact matching
  - Used in most traditional models: BM25 (*Robertson et al., 1994*), LM (Metzler et al., 2004), Vectoriel (*Salton et al., 1975*)
  - Based on exact matching of words of both the input sequences

- **Limitations**

  - Vocabulary mismatch, and luck of semantics  e.g. *Combating* *Alien* *Smuggling*
  - Sparse representations

$$e(\text{“alien”}) = [0, ..., 1, 0, ...0]$$
$$dim = |V|$$

$D_1^+$

<DOC>
<FIRST>r j AM-HumanSmuggling 03-12 0659</FIRST>
<SECOND>AM-Human Smuggling,0682</SECOND>
<HEAD>
Reports Say Smugglers Sell Bangladeshis Into Prostitution, Servitude
</HEAD>
...
<TEXT>
Border guards rescued 88 men, women and children from traffickers who were trying to smuggle them out for
...
A senior police official, speaking on condition of anonymity, said hundreds of women had been smuggled across the border in recent years. ...
In some cases, people have paid smugglers who promised them an escape from Bangladesh's grinding poverty, according to newspapers. The smugglers, known...Bangladesh and the Indian state of West Bengal in an effort to stop the smuggling...women and children had been rescued from the smugglers and 15 men had been charged with human trafficking since Jan. 1.
...
The Dainik Bangla, a Bengali-language newspaper, said last October that police suspect some children have been smuggled out of Bangladesh for more macabre purposes...
</TEXT>
</DOC>

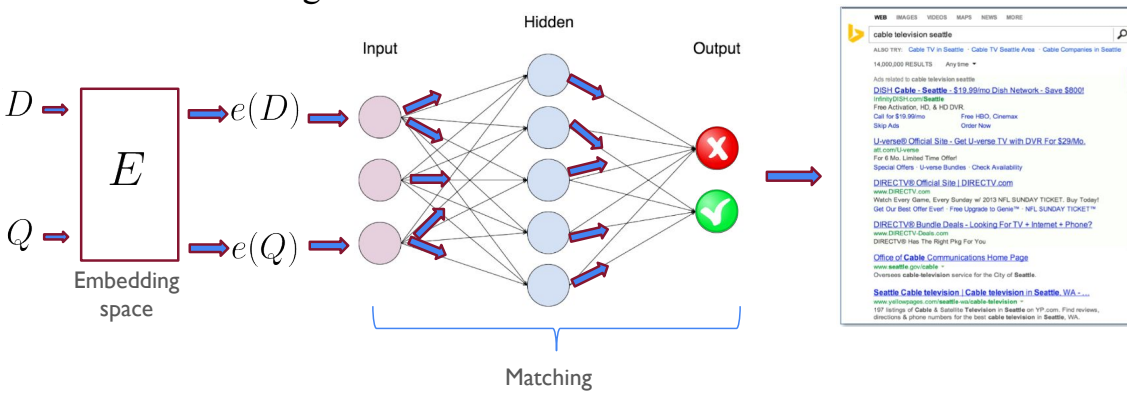
$D_2^-$

<DOC>
<FIRST>r a PM-AlienHouse 10-14 0479</FIRST>
<SECOND>PM-Alien House,0493</SECOND>
<HEAD>
House Functioned As Debtors Prison For Illegal Aliens
</HEAD>
<DATELINE>FULLERTON, Calif. (AP) </DATELINE>
<TEXT>
More than six dozen illegal aliens, including three tearful children, were found by authorities in a house that served as a debtors prison for immigrants who could not pay their smugglers... and Naturalization Service and 14 were suspected of being the smugglers, authorities said.
All the aliens were from Mexico and owed from $300 to ``whatever the market would bear'' to the smugglers, known as coyotes, Tom Gaines, an assistant district director for the INS, said Thursday.
...
Twelve of those arrested were arraigned Wednesday on conspiracy and illegal immigrant smuggling charges. Two are juveniles and will be deported, authorities said...ay as part of a four-month investigation into an alleged alien smuggling ring, Gaines said. ... The smugglers just had guards at every door. It was plain intimidation to keep them there.''.., illegal aliens who cannot pay their smugglers try to get word of their plight and need of money to relatives...
</TEXT>
</DOC>

# CONTEXT

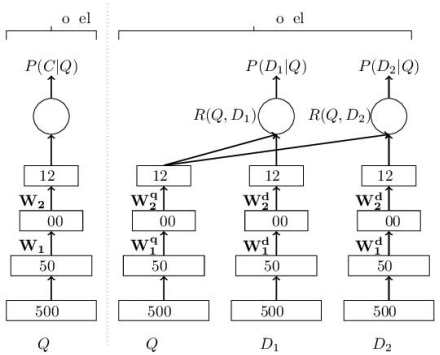
**Solution (before 2018):**

Use of neural networks for matching

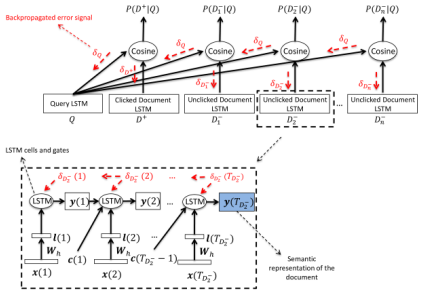



- Objective
  - Motivated by the advances in image processing using NN (*Lecun et al., 2015*)
  - Take advantage of the computational ability and non-linearity of NNs to capture latent characteristics related to text representation and matching.

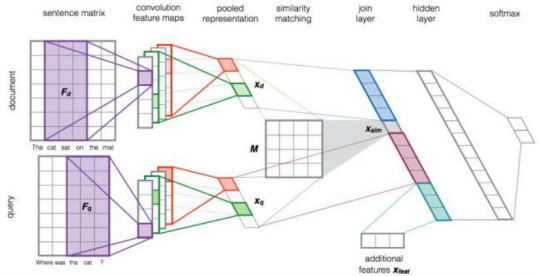
# DL evolution during 2013-2018 IR



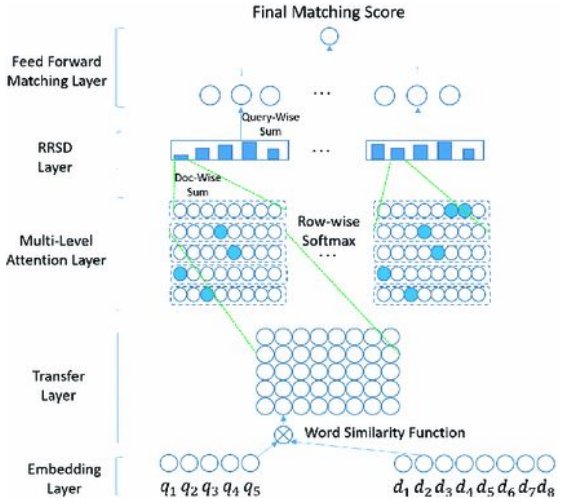

Huang et al. 2013




Palangi et al. 2015

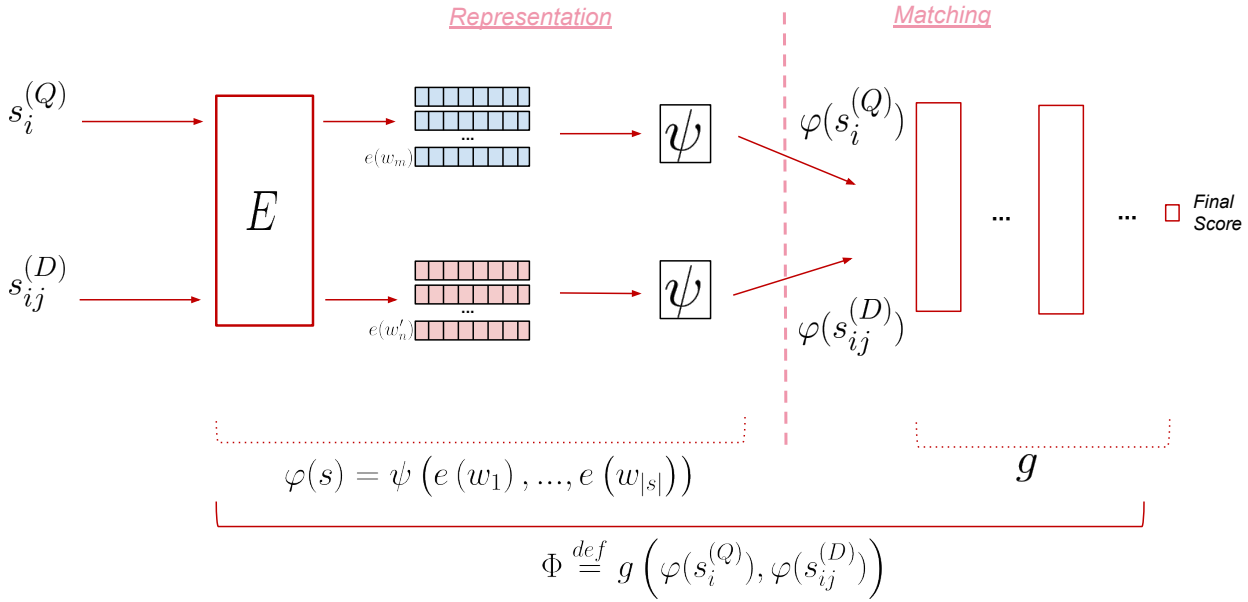



Severyn and Moschitti 2015




Sun and Wu 2018

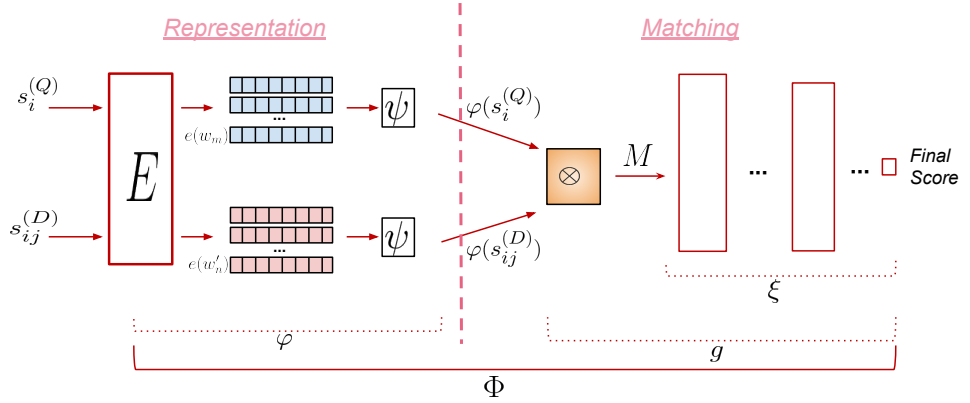
- Neural models for text matching applications
  - Sequences to match: $s_i^{(Q)} =< w_1 \dots w_m >$ and $s_{ij}^{(D)} =< w'_1 \dots w'_n >$

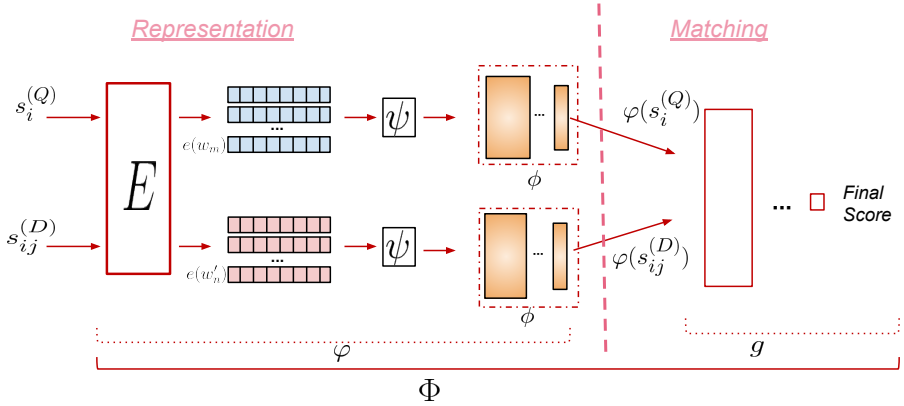


**Neural text matching models** (*Guo et al., 2016*)

**Interaction Focused**
- ARC-II (*Hu et al., 2014*)
- DRMM (*Guo et al., 2016*)
- DUET (*Mitra et al., 2017*)

**Representation Focused**
- DSSM (*Huang et al., 2013*)
- CLTR (*Severyn et al., 2015*)
- DRCN (*Kim et al., 2019*)

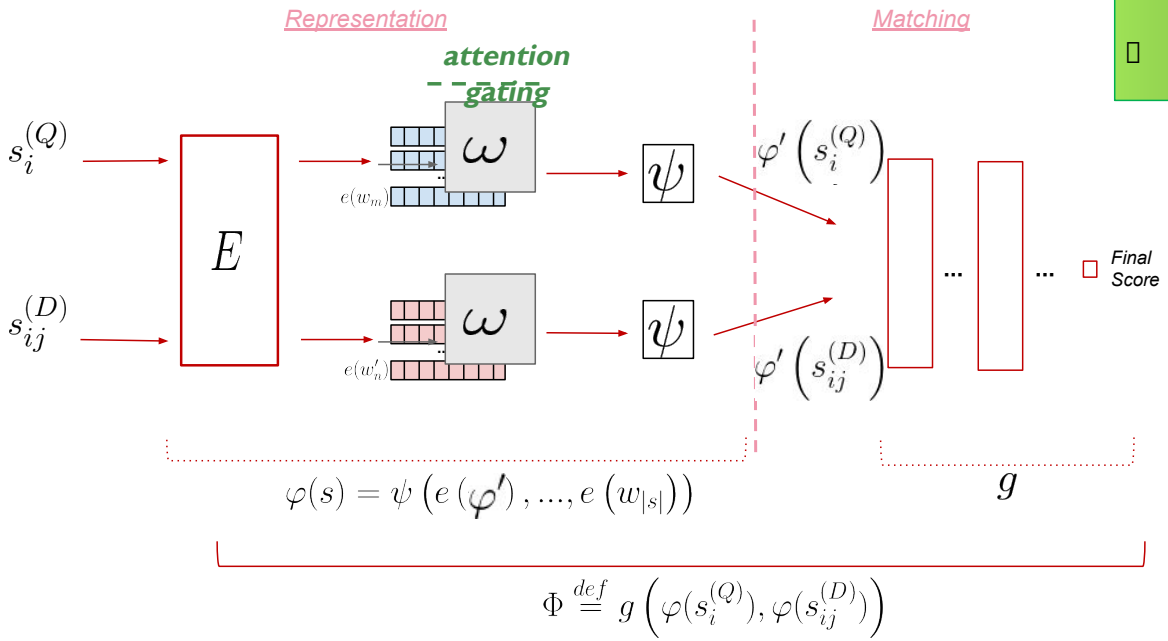


# CONTRIBUTIONS

## Asymmetry-sensitive neural models for text matching

- ***Asymmetry*** **sensitive architecture**




- Attention models (*Bahdanau et al., 2014*), (*Yang et al., 2016*);
- Determine the core information in a sequence.

Thiziri Belkacem, Jose G. Moreno, Taoufiq Dkaki, Mohand Boughanem. *Asymmetry Sensitive Architecture for Neural Text Matching.* European Conference on Information Retrieval (ECIR) 2019: 62-69

# CONTRIBUTIONS

## Asymmetry-sensitive neural models for text matching

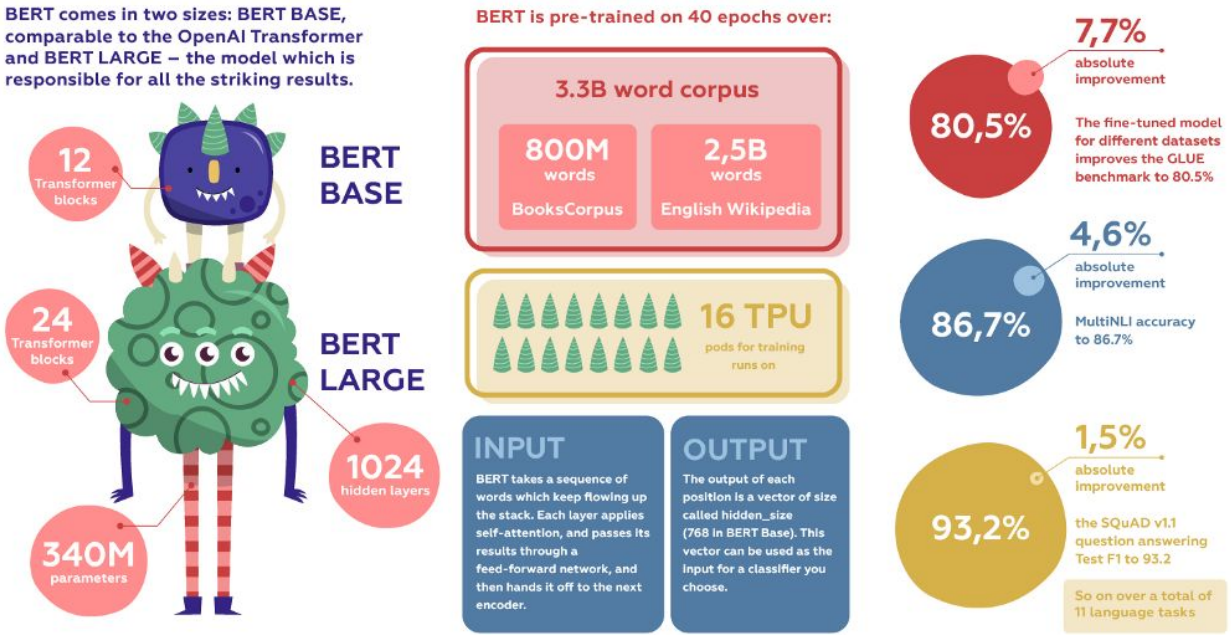
*Legend:*
□: extended with the attention layer
W (win) improved
L (loss) deteriorated
T (tied) unchainged
▲ significant improvement

- ***Results***
  - *What would be the results if the model is able to set its architecture according to the type of question?*
    - For a model $\Phi$ the performance w.r.t. the measure $M$ is the maximum possible:

$$M(\Phi_{oracle}) \geq max_M \left( M(\Phi), M(\Phi_{(Q)}), M(\Phi_{(A)}), M(\Phi_{(Q+A)}) \right)$$

| Version | Models | Performances | | | | | |
|---|---|---|---|---|---|---|---|
| | | *MAP* | *P@1* | *P@3* | *nDCG@1* | *nDCG@3* | *MRR* |
| *Original* | *CDSSM* | 0.5473 | 0.3671 | 0.2475 | 0.3671 | 0.5285 | 0.5586 |
| | *DUET* | 0.6113 | 0.4599 | 0.2714 | 0.4599 | 0.6016 | 0.6259 |
| | *MatchPyramid* | 0.6443 | 0.4726 | 0.2869 | 0.4726 | 0.6448 | 0.6529 |
| | *MV-LSTM* | 0.6046 | 0.4388 | 0.2813 | 0.4388 | 0.6101 | 0.6215 |
| *Oracle* | *CDSSM.*□ | 0.7180▲ | 0.5992▲ | 0.3136▲ | 0.5992▲ | 0.7220▲ | 0.7349▲ |
| | *DUET.*□ | 0.7354▲ | 0.6203▲ | 0.3136▲ | 0.6203▲ | 0.7350▲ | 0.7485▲ |
| | *MatchPyramid.*□ | 0.7726▲ | 0.6456▲ | 0.3375▲ | 0.6456▲ | 0.7810▲ | 0.7862▲ |
| | *MV-LSTM.*□ | 0.7569▲ | 0.6540▲ | 0.3263▲ | 0.6540▲ | 0.7677▲ | 0.7778▲ |

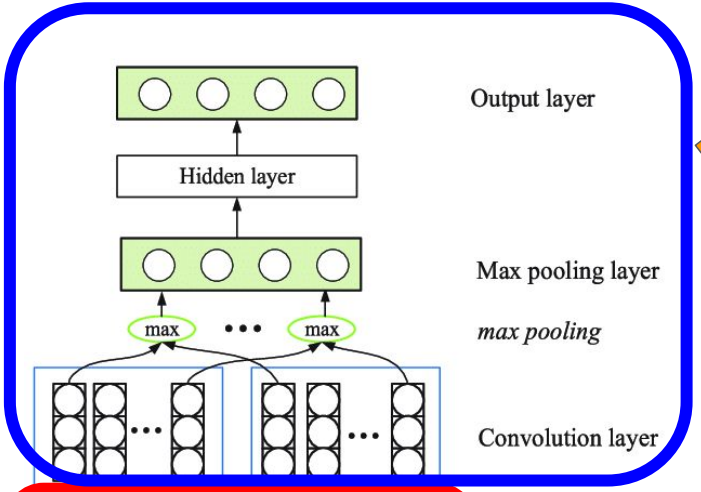
# BERT




one word, one vector

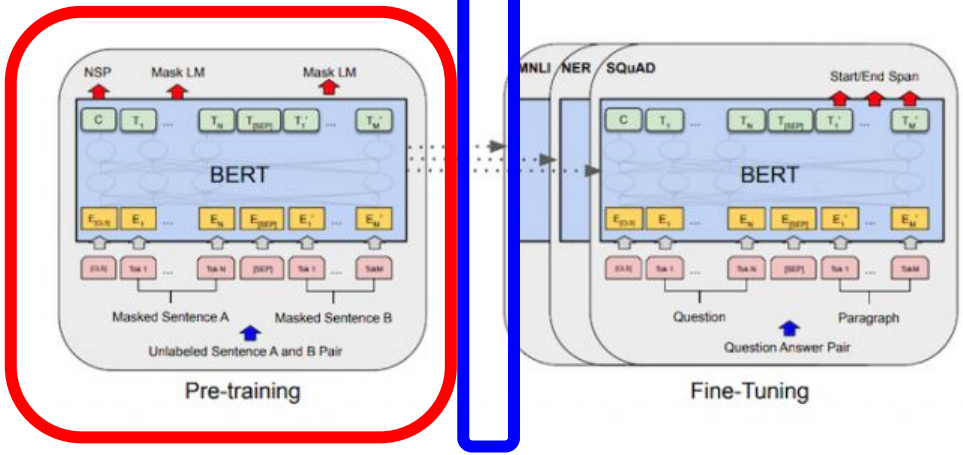
one word, multiple vectors

# One model to rule them all

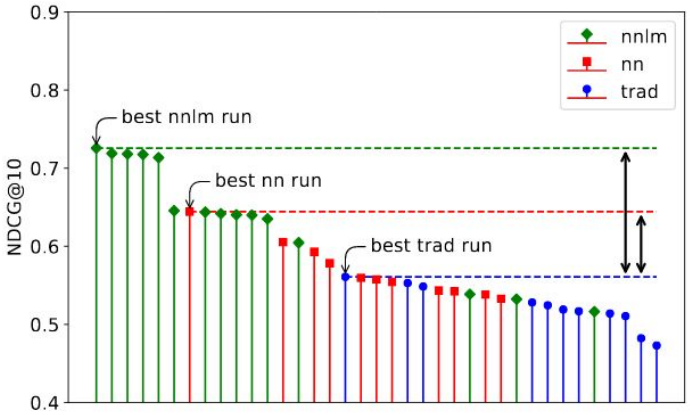



word2vec

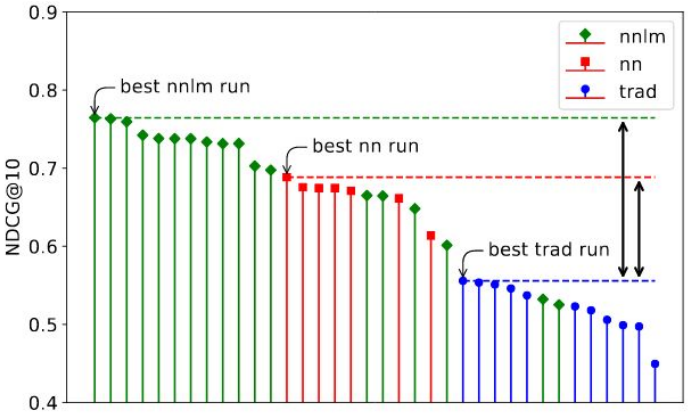
BERT



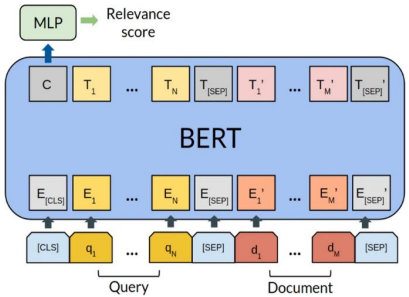
# TREC deep learning track 2019




(a) Document retrieval task



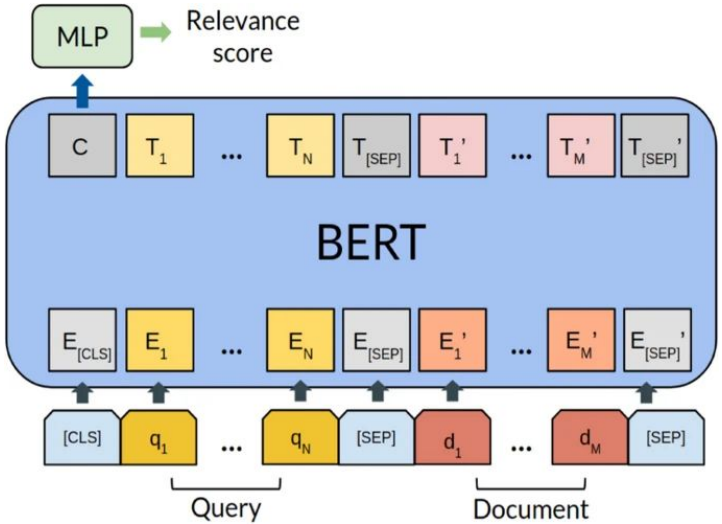

(b) Passage retrieval task




Cross encoders raise in IR!

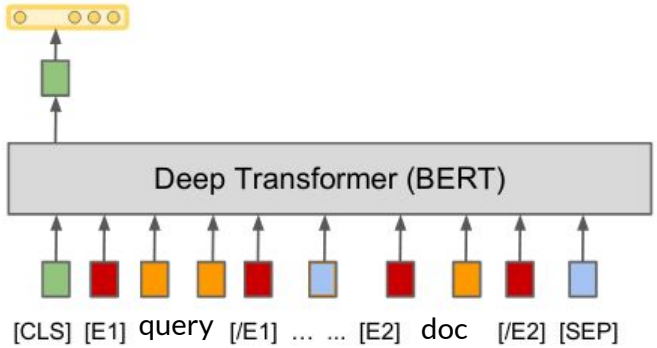
Voorhees, E. M., Craswell, N., Mitra, B., Campos, D., & Yilmaz, E. (2020). Overview of the TREC 2019 Deep Learning Track.

# MarkedBERT




$$\tilde{Q} = \{q_1, \ldots, [e_n] q_n [/e_n], \ldots, [e_m] q_m [/e_m], \ldots, q_{|Q|}\}$$
$$\tilde{D} = \{d_1, \ldots, [e_n] d_i [/e_n], \ldots, [e_n] d_j [/e_n], \ldots, [e_m] d_l [/e_m], \ldots, d_{|D|}\}$$

add information regarding query terms as in the asymmetric model. Focus differently in query terms!




Lila Boualili, Jose G. Moreno, Mohand Boughanem. *MarkedBERT: Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval.* SIGIR2020

# MarkedBERT

| Model | MRR@10 |
|---|---|
| BM25 [10] | 18.4 |
| Doc2query + BM25 [10] | 22.1 |
| Base model (Ours) | 30.2 |
| MarkedBERT | 32.8† |

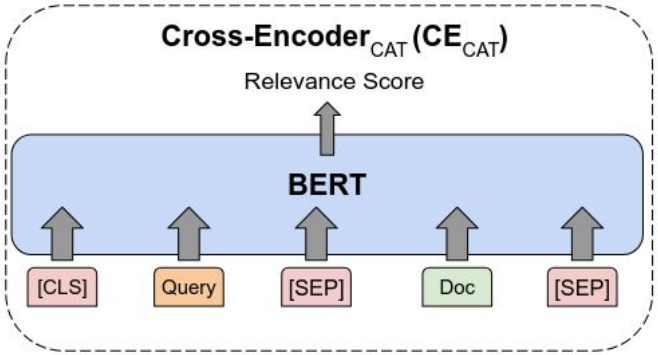
MRR@10 percentage of the Base model and Marked-BERT on the MS MARCO development set

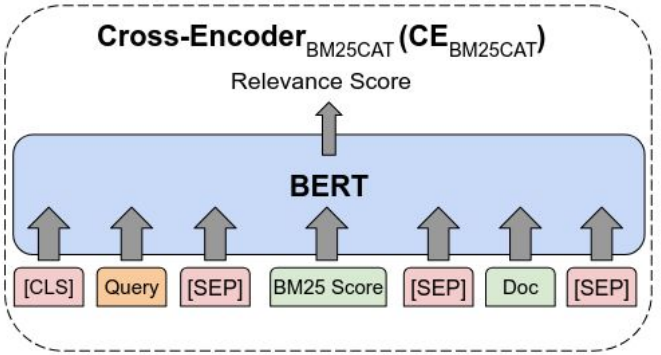
| QL | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| # | 273 | 621 | 1047 | 1156 | 1053 | 759 | 506 | 360 | 489 |
| D | 27.8 | 24.2 | 24.5 | 25.5 | 27.6 | 22.4 | 24.9 | 21.5 | 21.2 |
| B | 38.1 | 36.5 | 36.0 | 33.7 | 34.5 | 32.0 | 31.9 | 31.1 | 26.3 |
| M | 41.7 | 40.7 | 39.5 | 37.3 | 37.2 | 33.5 | 34.0 | 33.2 | 28.6 |
| d1 | 50.2 | 68.1 | 60.7 | 46.7 | 34.8 | 49.8 | 36.7 | 54.8 | 35.1 |
| d2 | 09.5 | 11.6 | 09.5 | 11.2 | 07.8 | 04.7 | 06.6 | 06.7 | 08.8 |

Average MRR@10 percentage per query length (QL). D: Doc2query + BM25, B: Base, M: MarkedBERT, #: Query count, d1, d2(%): improvement of MarkedBERT over Doc2query + BM25 and Base model respectively.

Lila Boualili, Jose G. Moreno, Mohand Boughanem. *MarkedBERT: Integrating Traditional IR Cues in Pre-trained Language Models for Passage Retrieval.* SIGIR2020

# MarkedBERT+BM25



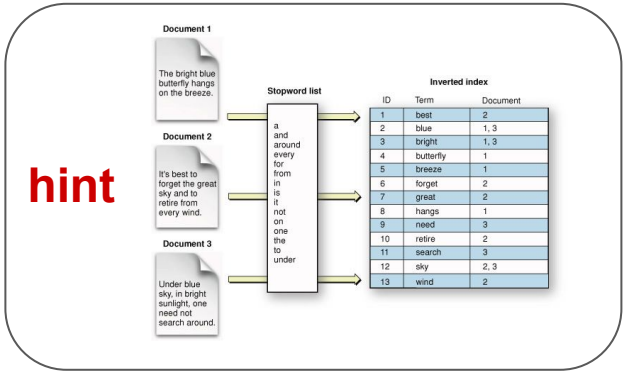




$$BM25(q,d) = \sum_{t \in q \cap d} rsj_t . \frac{tf_{t,d}}{tf_{t,d} + k_1\{(1-b) + b\frac{|d|}{l}\}}$$

| Model | TREC DL 20 | | TREC DL 19 | | MSMARCO DEV | | |
|---|---|---|---|---|---|---|---|
| | nDCG@10 | MAP | nDCG@10 | MAP | nDCG@10 | MAP | MRR@10 |
| BM25 | .480 | .286 | .506 | .377 | .234 | .195 | .187 |
| **Re-rankers** | | | | | | | |
| BERT-Base$_{CAT}$ | .689 | .447 | .713 | .441 | .399 | .346 | .342 |
| BERT-Base$_{BM25CAT}$ | .705† | .475† | .723† | .453† | .422† | .367† | .364† |
| BERT-Large$_{CAT}$ | .695 | .464 | .714 | .467 | .401 | .344 | .360 |
| BERT-Large$_{BM25CAT}$ | **.728**† | **.482**† | **.731**† | **.477**† | **.424**† | .367† | **.369**† |
| DistilBERT$_{CAT}$ | .670 | .442 | .679 | .440 | .383 | .310 | .325 |
| DistilBERT$_{BM25CAT}$ | .682† | .456† | .699† | .451† | .390† | .323† | .339† |
| MiniLM$_{CAT}$ | .681 | .448 | .704 | .452 | .419 | .363 | .360 |
| MiniLM$_{BM25CAT}$ | .710† | .473† | .711† | .463† | **.424**† | **.368**† | .367† |

Askari, A., Abolghasemi, A., Pasi, G., Kraaij, W., & Verberne, S. Injecting the BM25 score as text improves BERT-based re-rankers. ECIR2023

hint




# Cross-encoders bottleneck

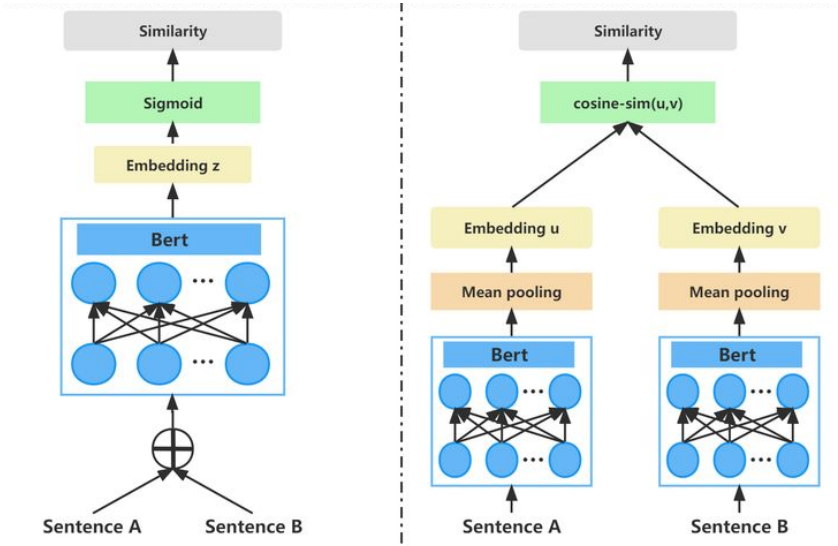
# Cross-encoders bottleneck

Querying is expensive!!! A new query demands a new BERT encoding of the full collection (millions)

→ Solution: use Bi-Encoders

**here start the raise of bi-encoders (such as sentenceBERT) in IR**

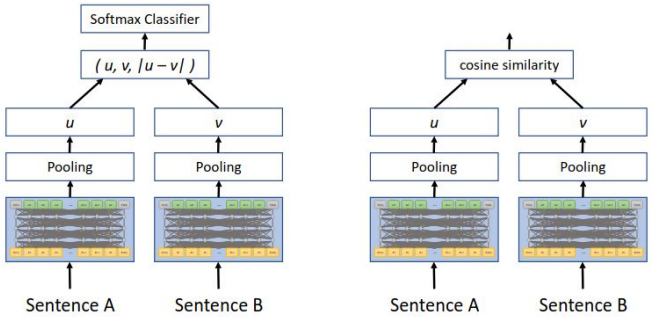
# Dense Passage Retrieval (DPR)




Source: https://link.springer.com/article/10.1007/s13042-024-02496-7

- Create the representations of documents
- Create the representation of the query
- Retrieve k documents vectors based on the query vector

# SentenceBERT -> Dense Passage Retrieval (DPR)




| Training | Model | NQ | TriviaQA | WQ | TREC | SQuAD |
|---|---|---|---|---|---|---|
| Single | BM25+BERT (Lee et al., 2019) | 26.5 | 47.1 | 17.7 | 21.3 | 33.2 |
| Single | ORQA (Lee et al., 2019) | 33.3 | 45.0 | 36.4 | 30.1 | 20.2 |
| Single | HardEM (Min et al., 2019a) | 28.1 | 50.9 | - | - | - |
| Single | GraphRetriever (Min et al., 2019b) | 34.5 | 56.0 | 36.4 | - | - |
| Single | PathRetriever (Asai et al., 2020) | 32.6 | - | - | - | **56.5** |
| Single | $\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020) | 39.2 | - | 40.2 | 46.8 | - |
| Single | $\text{REALM}_{\text{News}}$ (Guu et al., 2020) | 40.4 | - | 40.7 | 42.9 | - |
| Single | BM25 | 32.6 | 52.4 | 29.9 | 24.9 | 38.1 |
| | DPR | **41.5** | 56.8 | 34.6 | 25.9 | 29.8 |
| | BM25+DPR | 39.0 | 57.0 | 35.2 | 28.0 | 36.7 |
| Multi | DPR | **41.5** | 56.8 | **42.4** | 49.4 | 24.1 |
| | BM25+DPR | 38.8 | **57.9** | 41.1 | **50.6** | 35.8 |

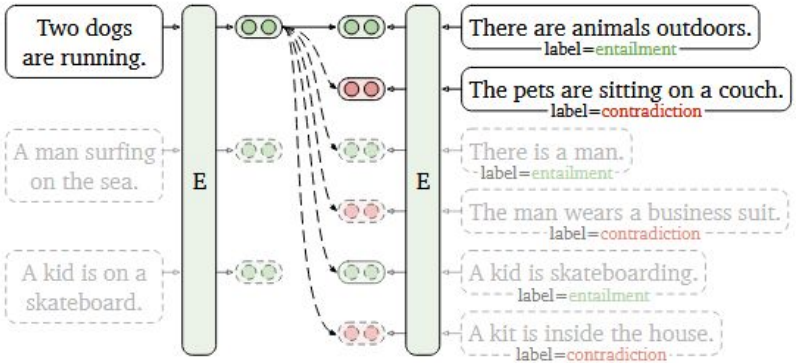
Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. EMNLP2019

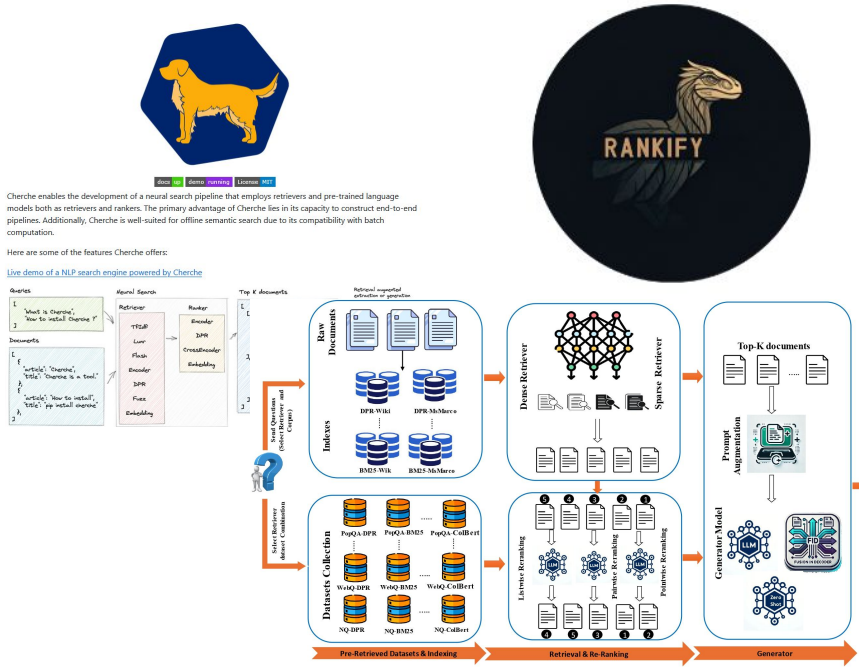
Karpukhin, V., Oguz, B., Min, S., Lewis, P. S., Wu, L., Edunov, S., ... & Yih, W. T. (2020, November). Dense Passage Retrieval for Open-Domain Question Answering. EMNLP2020

# SentenceTransformers - how to train? how to use?

How to create a Document-Query vector space?

Goal: Relevant pairs of questions-passages will have a smaller distance compared to the irrelevant ones.





Check: https://huggingface.co/blog/train-sentence-transformers

Gao, T., Yao, X., & Chen, D.. Simcse: Simple contrastive learning of sentence embeddings. EMNLP2021

Sourty, R., Moreno, J. G., Tamine, L., & Servant, F. P.. Cherche: A new tool to rapidly implement pipelines in information retrieval. SIGIR2022

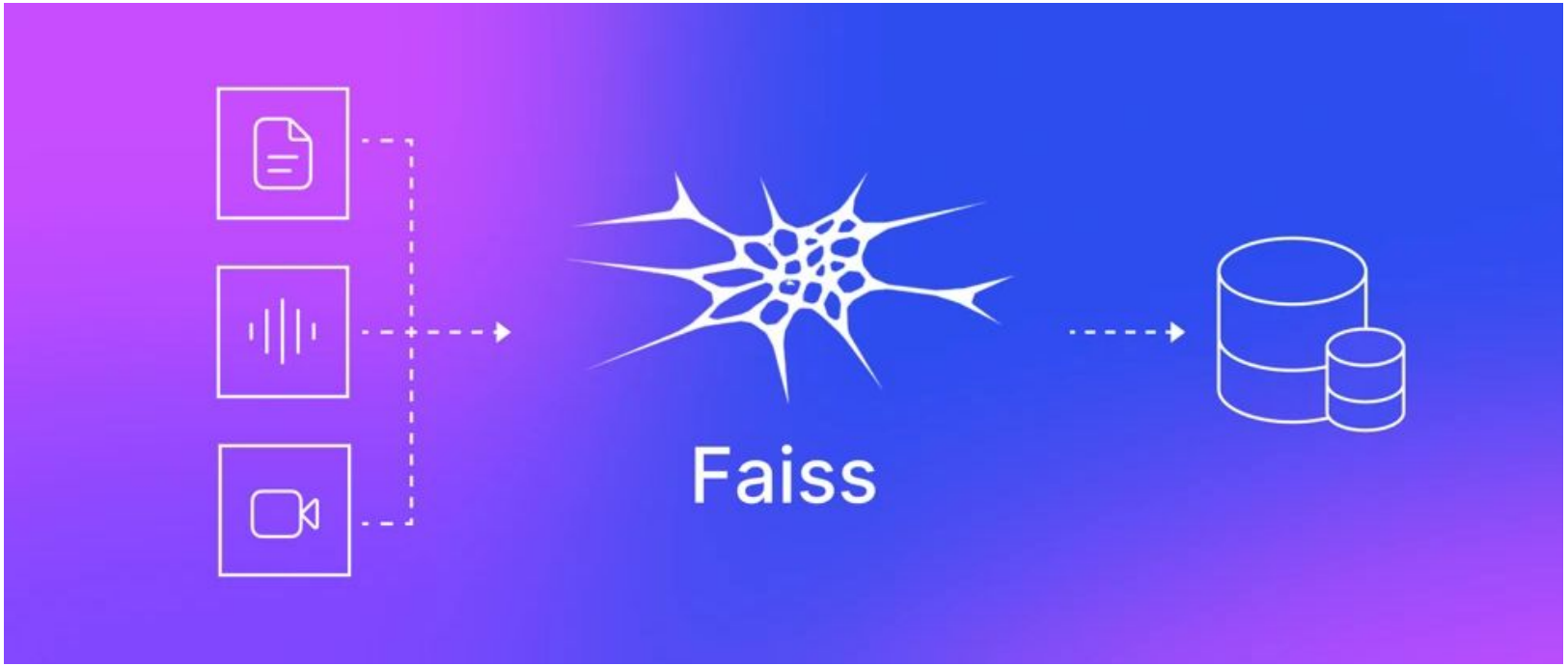
Abdallah, A., Piryani, B., Mozafari, J., Ali, M., & Jatowt, A. (2025). Rankify: A comprehensive python toolkit for retrieval, re-ranking, and retrieval-augmented generation. arXiv preprint arXiv:2502.02464.

but Bi-encoders are dense, there are ways to access the documents quickly and easily?

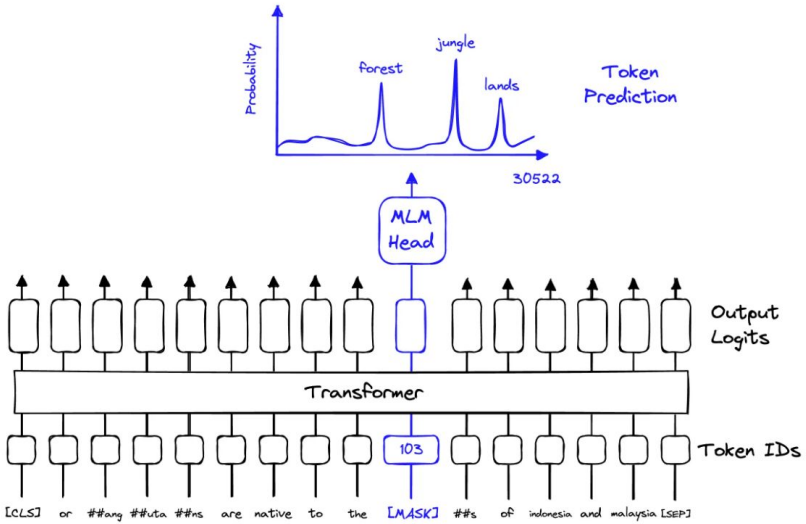
# but Bi-encoders are dense, there are ways to access the documents quickly and easily?

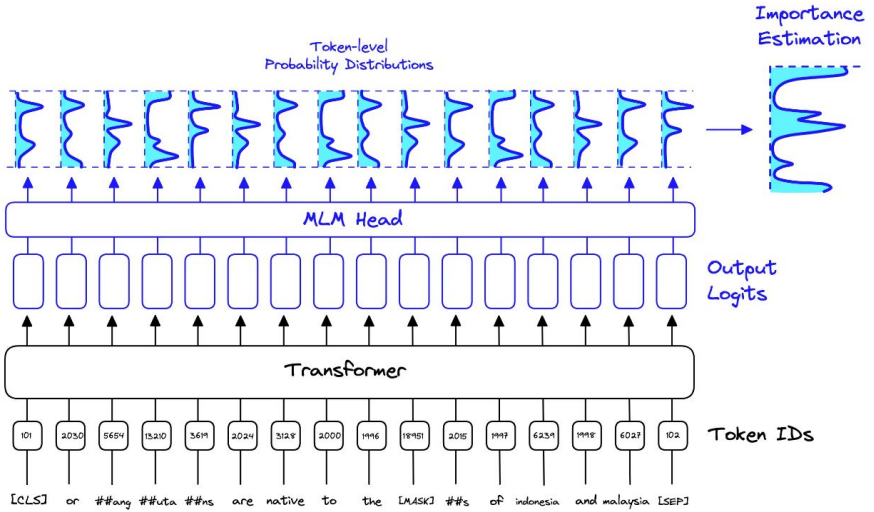
Yes! Use HNSW, PQ, IVF, etc…or Sparse IR Models
→ Solution: FAISS
here start the raise of Indexes for dense bi-encodes and the use of sparse bi-encoders (such as SPLADE) in IR

# Facebook AI Similarity Search



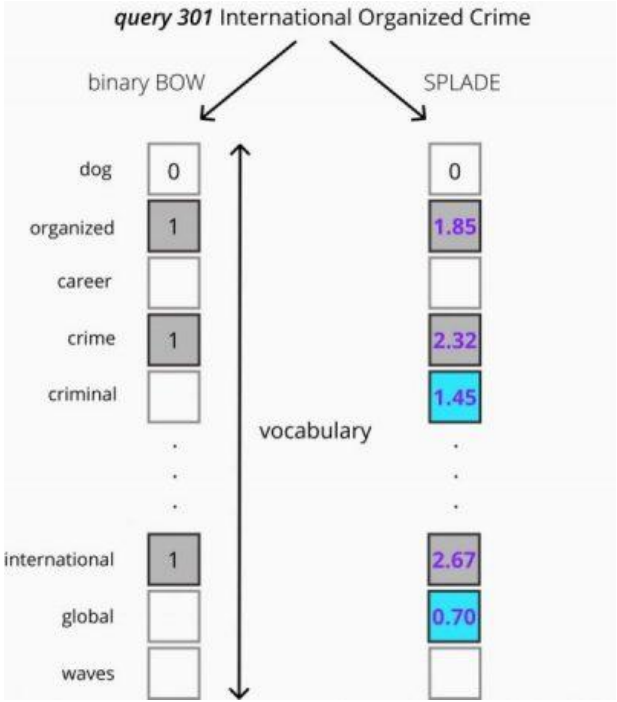
# SPLADE




J. Briggs, “SPLADE for Sparse Vector Search Explained —pinecone.io.” 2023.

# SPLADE - advantages

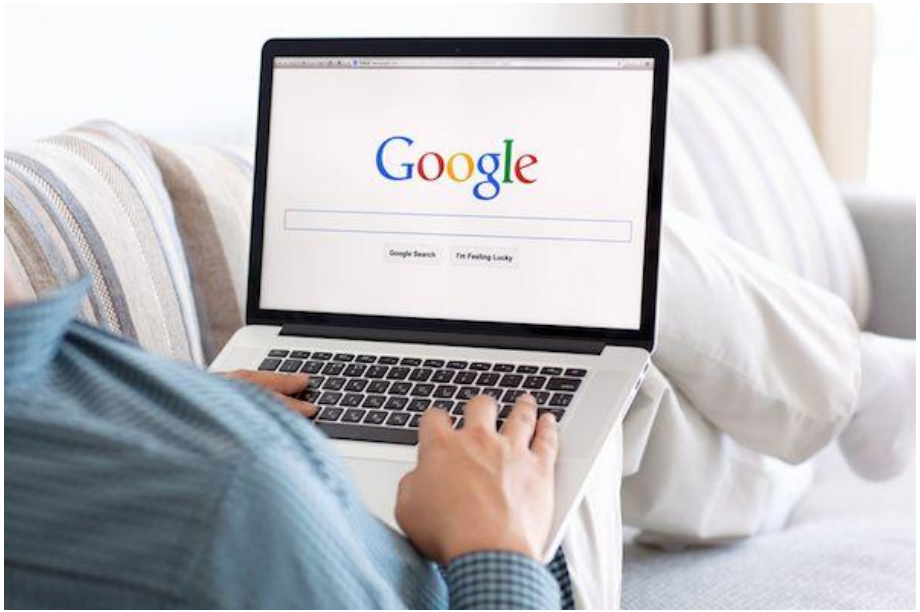


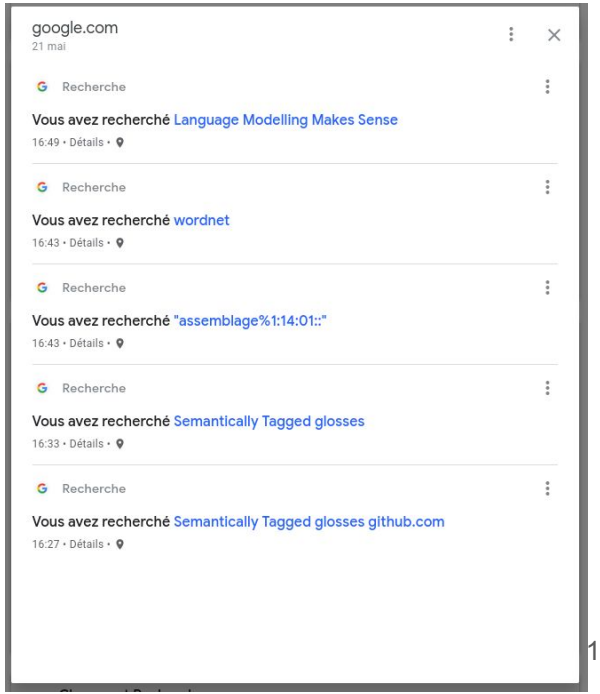

| Vector Type | Memory (GB) |
|---|---|
| Dense BERT Vector | 6.144 |
| OpenAI Embedding | 12.288 |
| Sparse Vector | 1.12 |

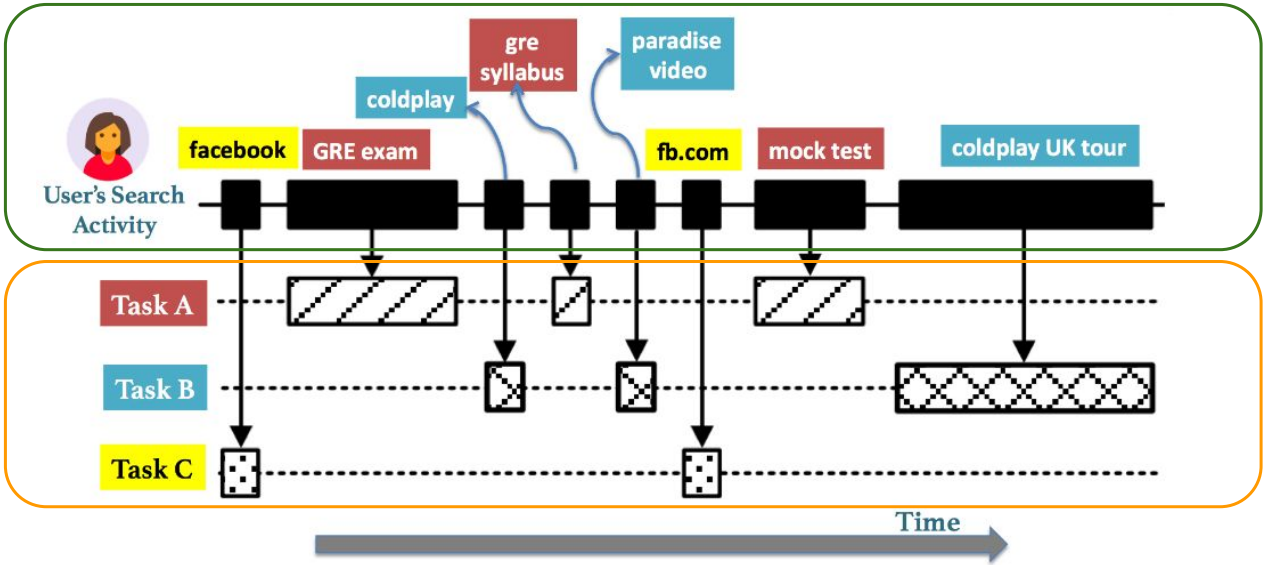
https://tinyurl.com/RAGETALTP2

before going further…let’s talk about queries

# Queries cannot be ignored





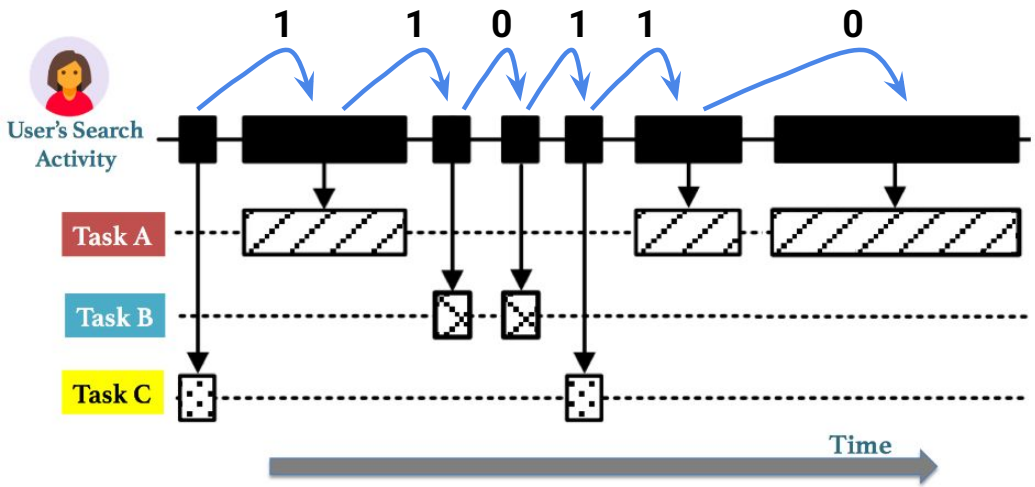

Mehrotra and Yilmaz, CIKM2017

**Known information**

**Models**

**Unknown patterns**

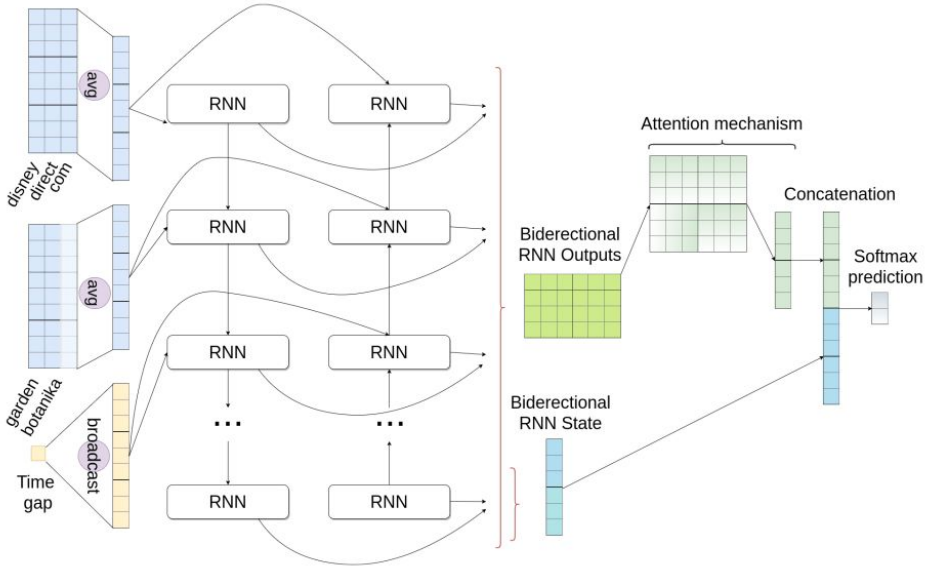
# Problem 1 - Finding task boundaries




**Input**:
Two query strings and metadatas

**Output**:
Boundary prediction

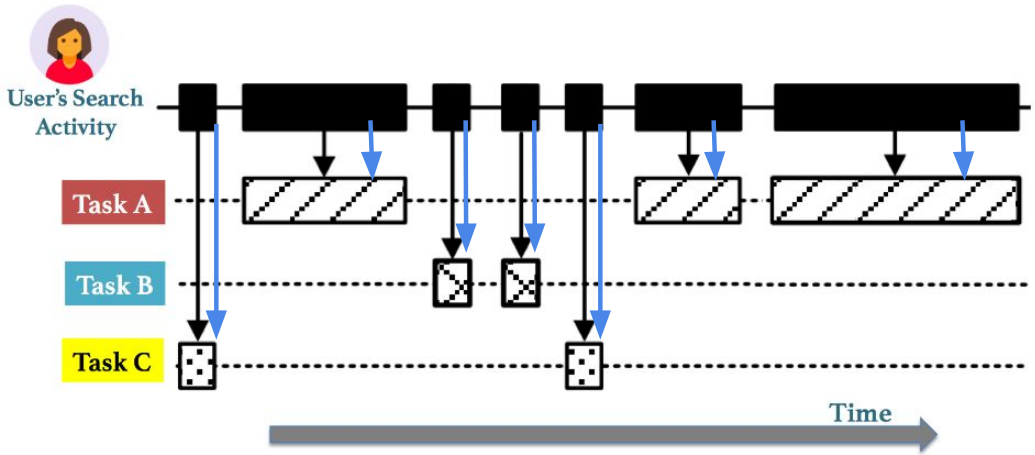
# Query segmentation




| Dataset | WSMC12 | | CSTE | |
|---|---|---|---|---|
| Segmentation model | Accuracy | F-score | Accuracy | F-score |
| Logistic regression | 0.733 | 0.238 | 0.634 | 0.322 |
| K-Nearest Neighbors | 0.865 | 0.701 | 0.708 | 0.460 |
| SVM, linear kernel | 0.735 | 0.022 | 0.669 | 0.232 |
| SVM, RBF kernel | 0.742 | 0.055 | 0.671 | 0.105 |
| Naive Bayes | 0.773 | 0.306 | 0.643 | 0.223 |
| QDA | 0.323 | 0.428 | 0.534 | 0.553 |
| Random Forest | 0.862 | 0.742 | 0.725 | 0.502 |
| AdaBoost | 0.862 | 0.739 | 0.721 | 0.504 |
| GPC | 0.905 | 0.811 | 0.720 | 0.417 |
| Decision Tree | 0.882 | 0.777 | 0.759 | 0.541 |
| HBSSM | 0.886 | 0.813 | 0.656 | 0.627 |
| BiRNN LSTM - time at AL | 0.921 | 0.861 | 0.784 | 0.651 |
| BiRNN GRU - time at AL | 0.927 | 0.867 | **0.789** | 0.648 |
| BiRNN LSTM | 0.931 | 0.875 | 0.788 | **0.663** |
| BiRNN GRU | **0.937** | **0.884** | 0.751 | 0.604 |

Luis Lugo, Jose G. Moreno, Gilles Hubert. *Segmenting Search Query Logs by Learning to Detect Search Task Boundaries.* SIGIR2020

# Problem 2 - Task mapping




**Input**:
Query string and metadata

**Output**:
Task class

**Algorithm 1** MGBC algorithm

**Input**: Query log $Q$ **Output**: Task labels $L$

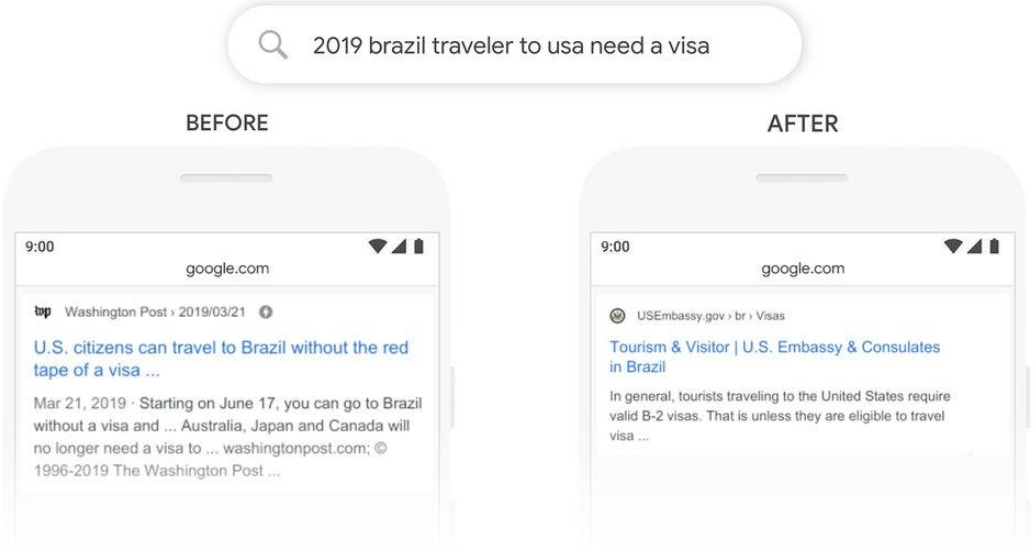
$V \leftarrow \{\}, E \leftarrow \{\}, G(V, E) \leftarrow (V, E)$
**for** all $q_i \in Q$ **do**
  $v_i \leftarrow multilingual\_vector(q_i)$
  $V \leftarrow V \cup \{v_i\}$
**end for**
**for** all $v_i, v_j \in V$ **do**
  $\mathbf{e}_k \leftarrow S_{ang}(v_i, v_j)$
  $E \leftarrow E \cup \{\mathbf{e}_k\}$
**end for**
**for** all $\mathbf{e}_k \in E$ **do**
  **if** $\mathbf{e}_k < \eta$ **then**
    $E \leftarrow E \setminus \{\mathbf{e}_k\}$
  **end if**
**end for**
**for** all $C_i \in G(V, E)$ **do**
  $task_i \leftarrow i$
  **for** all $v_k \in C_i$ **do**
    $L[v_k] \leftarrow task_i$
  **end for**
**end for**

| Clustering method | $\alpha$ | $\eta$ | $F_1$ | $F_{0.6}$ |
|---|---|---|---|---|
| QC-WCC | 0.8 | 0.4 | 0.471 | 0.428 |
| QRY-VEC word2vec | 0.6 | 0.5 | 0.473 | 0.441 |
| QRY-VEC tempo-lexical | 0.6 | 0.7 | 0.538 | 0.488 |
| MGBC | 0.4 | 0.3 | **0.624** | **0.695** |

| Dataset | Method | Accuracy | $F_1$ | $F_{0.6}$ | Query time |
|---|---|---|---|---|---|
| AOLQL | Trie | 0.693 | 0.543 | 0.543 | 0.029ms |
| | BM25 | **0.809** | **0.689** | **0.689** | 0.947s |
| | NGT | 0.751 | 0.608 | 0.607 | 0.308ms |
| TRECQT | Trie | 0.650 | 0.519 | 0.518 | 0.030ms |
| | BM25 | 0.791 | 0.688 | 0.688 | 2.532s |
| | NGT | **0.804** | **0.705** | **0.704** | 0.299ms |
| WIKIHQ | Trie | 0.471 | 0.310 | 0.311 | 0.032ms |
| | BM25 | 0.621 | 0.453 | 0.454 | 6.572m |
| | NGT | **0.648** | **0.481** | **0.481** | 0.368ms |

Luis Lugo, Jose G. Moreno, Gilles Hubert. *A Multilingual Approach for Unsupervised Search Task Identification.* SIGIR2020

# Large improvements with transformers-based models




We see **billions** of searches every day, and **15 percent** of those queries are ones **we haven't seen before**--

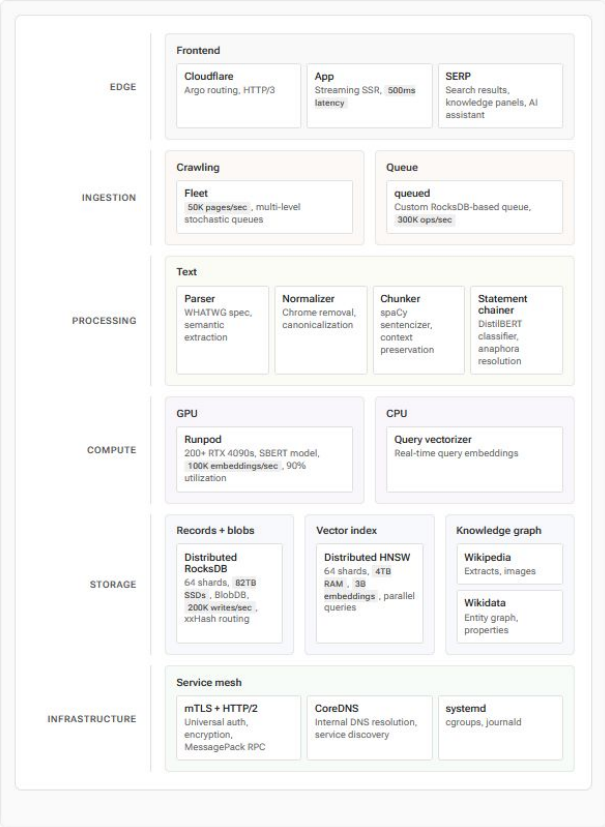
**...**

**BERT** will help Search better understand **one in 10 searches** in the U.S.

https://www.blog.google/products/search/search-language-understanding-bert/

but there are thousand of topics in IR and/or IR-related… in becomes hard to distinguish (and to know all them)

# Example of “required” topics to make a SE from scratch

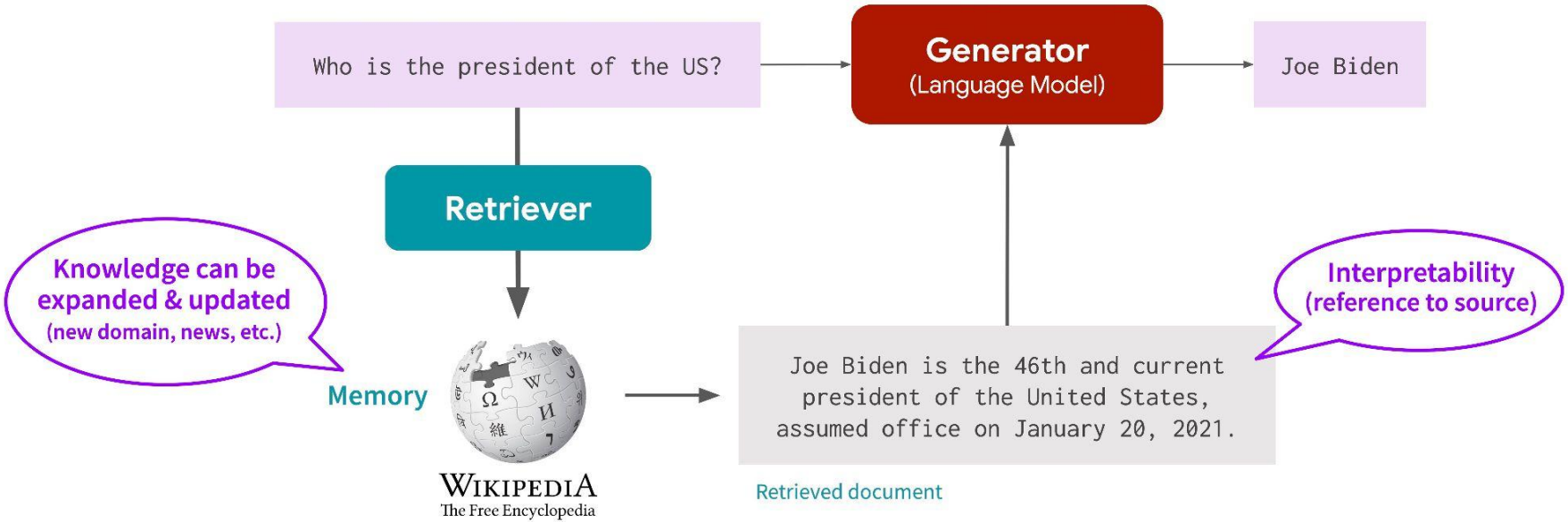


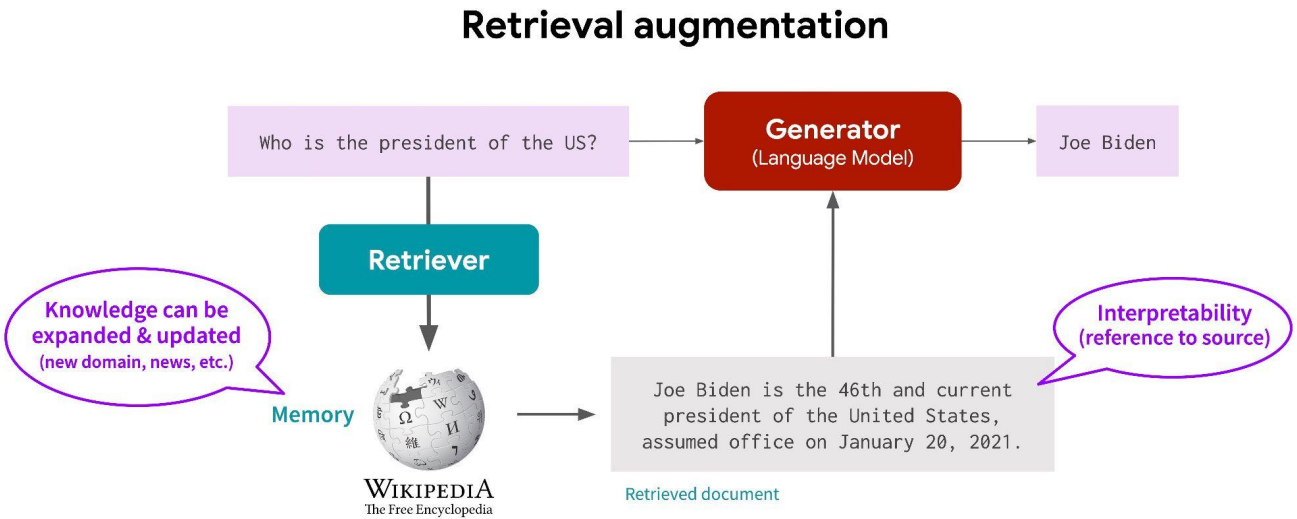

- Data crawling and processing
  - Queue administration
  - HTML storage and normalization
  - Chunking
- Semantic representation
  - Training a DPR
  - Online inference
  - Batch options are cheap nowadays
- Vector database
  - Low dimensions 768-3k
  - Billions inserts
  - Tools: CoreNN, FAISS,
  - Algorithms: HNSW, quantization,
- Knowledge graph
  - Wikipedia - wikidata

Source: https://blog.wilsonl.in/search-engine/

After this biased overview of the information retrieval field, let’s return to RAG…

# Retrieval augmentation

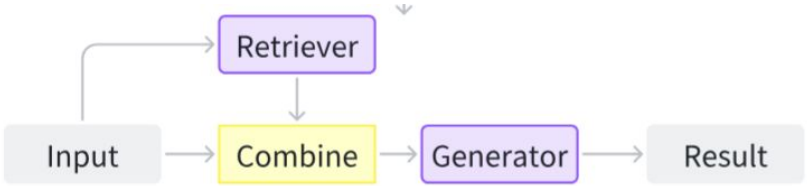




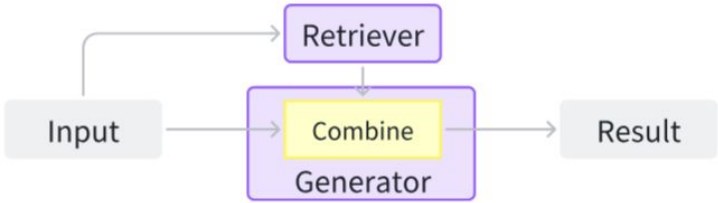

# What are the different ways a neural retriever can be integrated into a Language Model?

# Taxonomy of RAG foundations

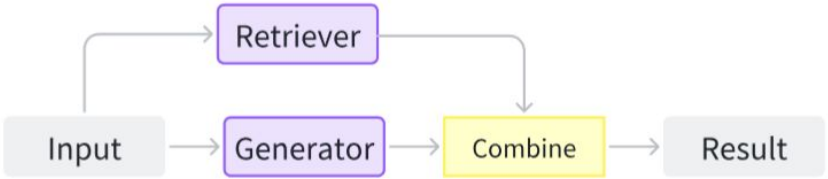
## Query-based RAG



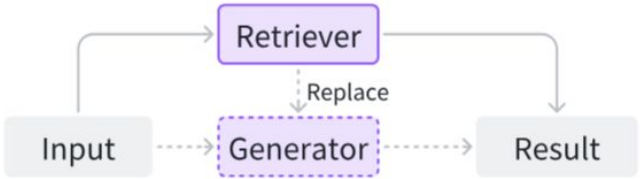

## Latent representation-based RAG




## Logit-based RAG




## Speculative RAG

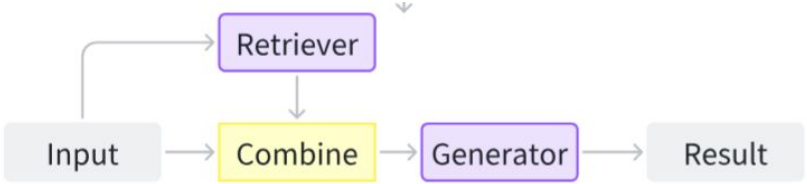



Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., ... & Cui, B. (2024). Retrieval-augmented generation for ai-generated content: A survey. arXiv preprint arXiv:2402.19473.

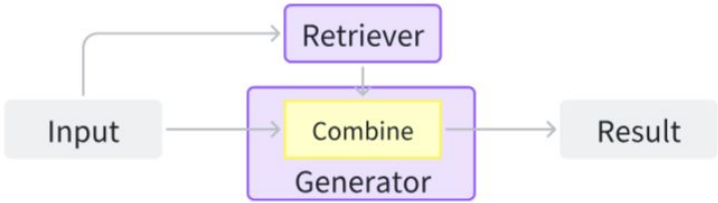
# Taxonomy of RAG foundations

**You already know this**

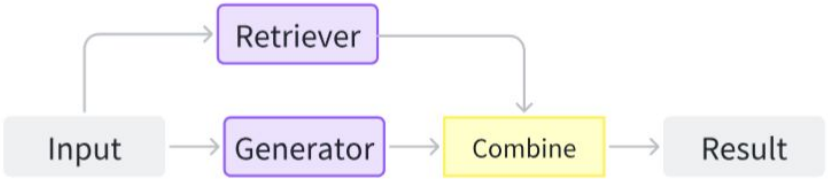
## Query-based RAG



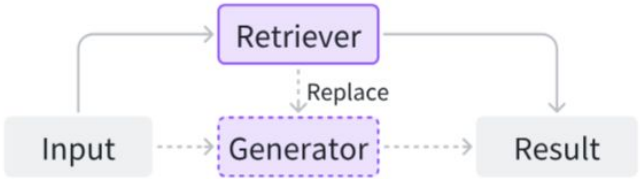

## Latent representation-based RAG




## Logit-based RAG



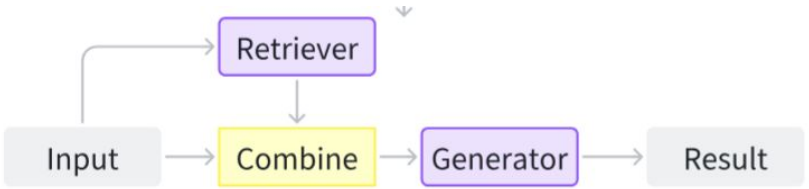

## Speculative RAG



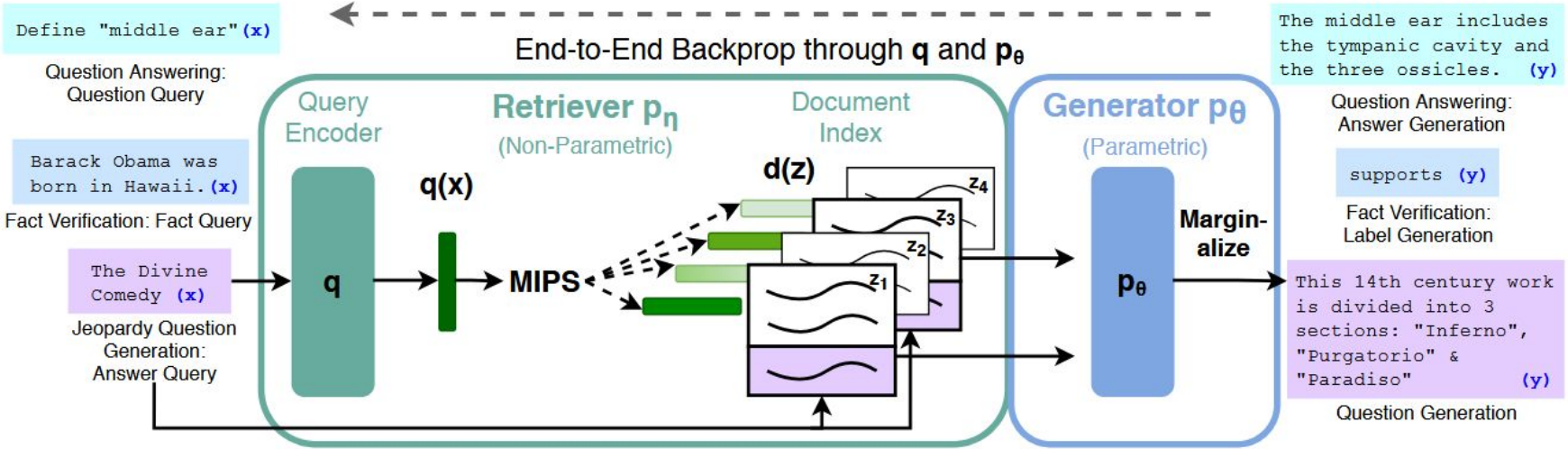

Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., ... & Cui, B. (2024). Retrieval-augmented generation for ai-generated content: A survey. arXiv preprint arXiv:2402.19473.

# Query-based RAG

- Query-based RAG integrates the user's query with insights from retrieved information
- It directly into the initial stage of the generators input
- This method is prevalent in RAG applications and is widely employed across various modalities

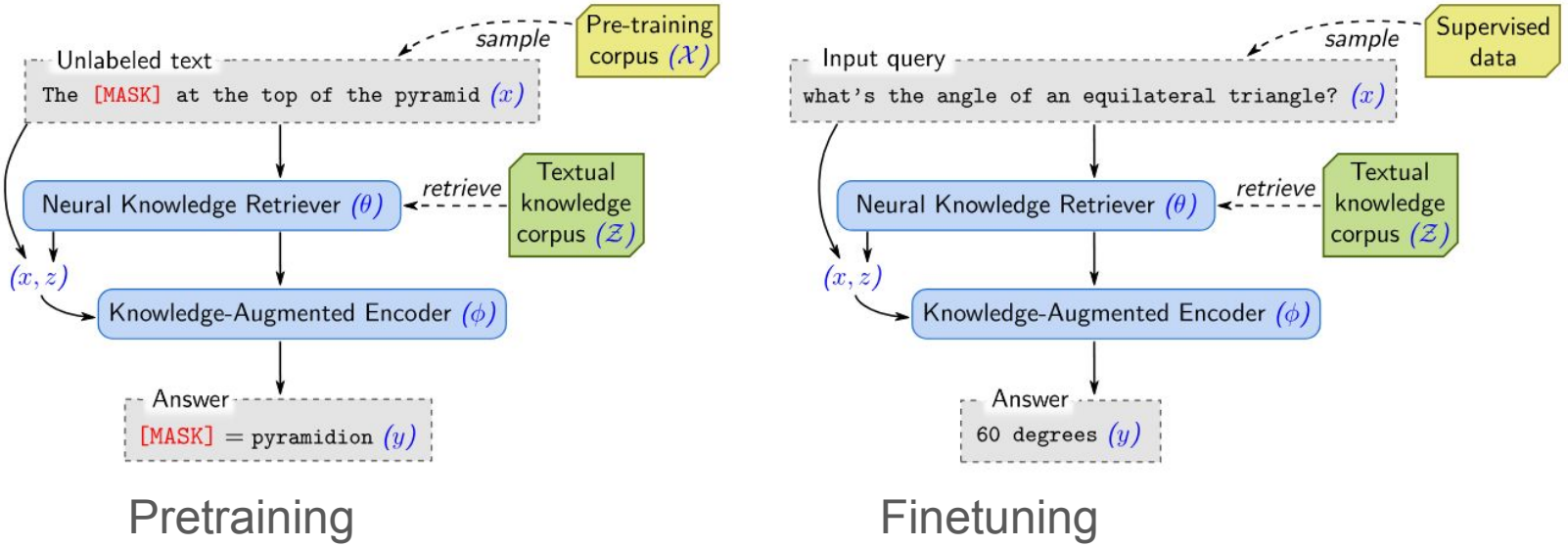


# RAG for knowledge-intensive NLP tasks



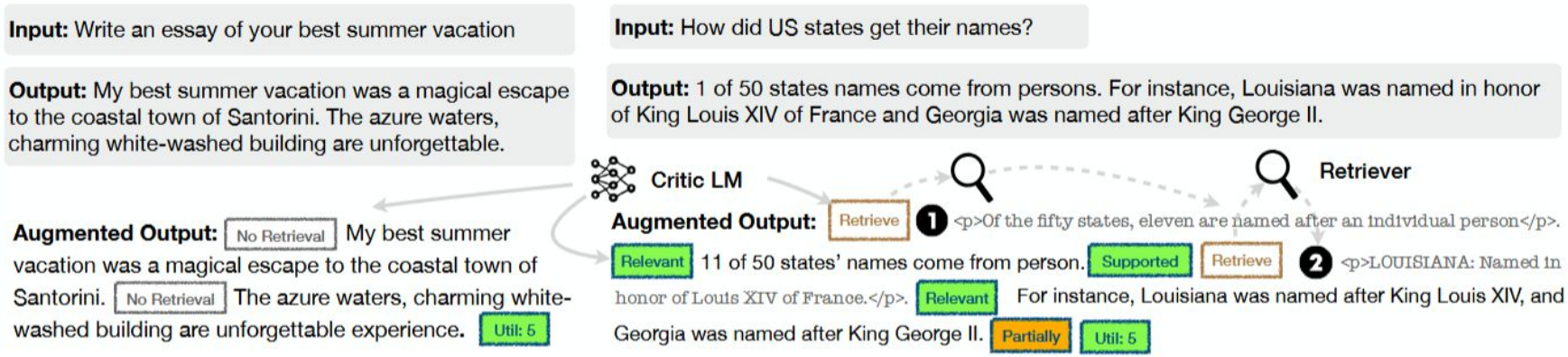

P. S. H. Lewis, E. Perez, A. Piktus et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in NeurIPS, 2020.

# REALM: Pre-training Retrieval Augmented LMs




Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020, November). Retrieval augmented language model pre-training. In International conference on machine learning (pp. 3929-3938). PMLR.

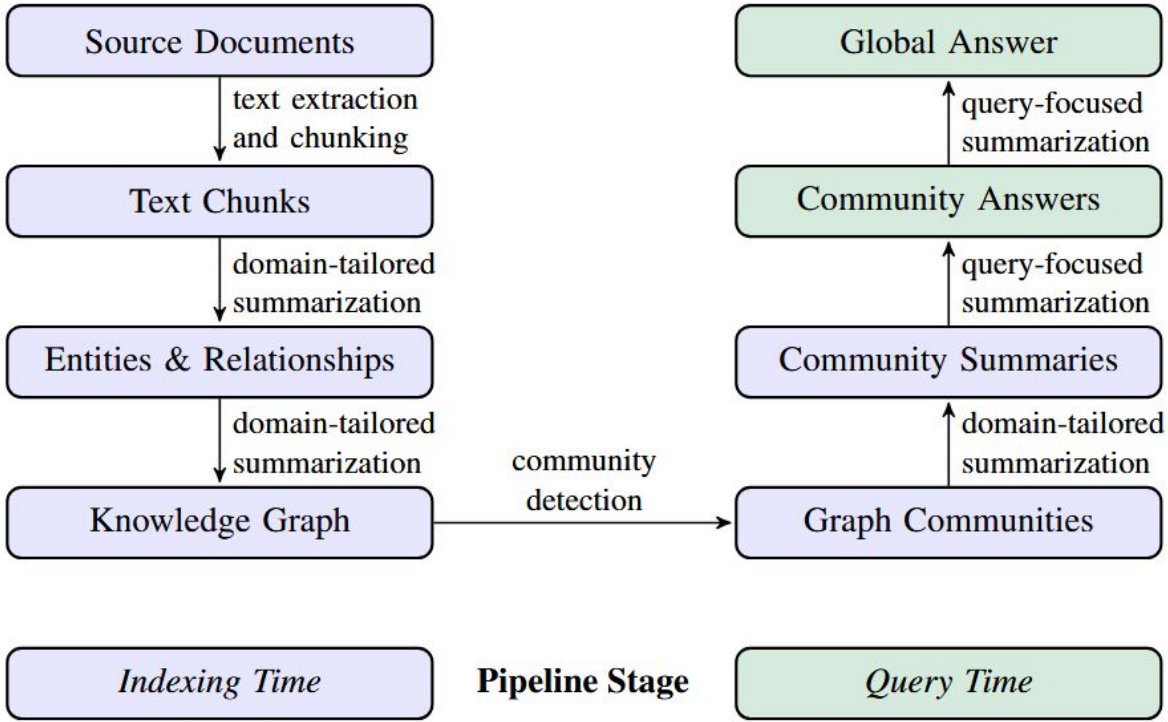
# Self-RAG




The left example doesn't involve retrieval, whereas the right one does, requiring the insertion of passages

A. Asai, Z. Wu, Y. Wang et al., "Self-rag: Learning to retrieve, generate, and critique through self-reflection," arxiv:2310.11511, 2023.

# GraphRAG




Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., ... & Larson, J. (2024). From local to global: A graph rag approach to query-focused summarization. arXiv preprint arXiv:2404.16130.

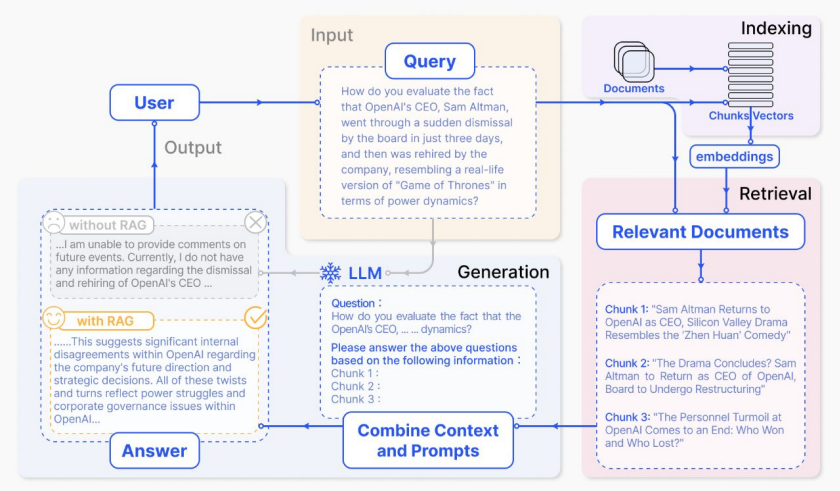
# A side question: is retrieval useful only for generation?

# Retrieval-Augmented Generative vs Extractive

What's the angle of an equilateral triangle?

LLM <- Retriever

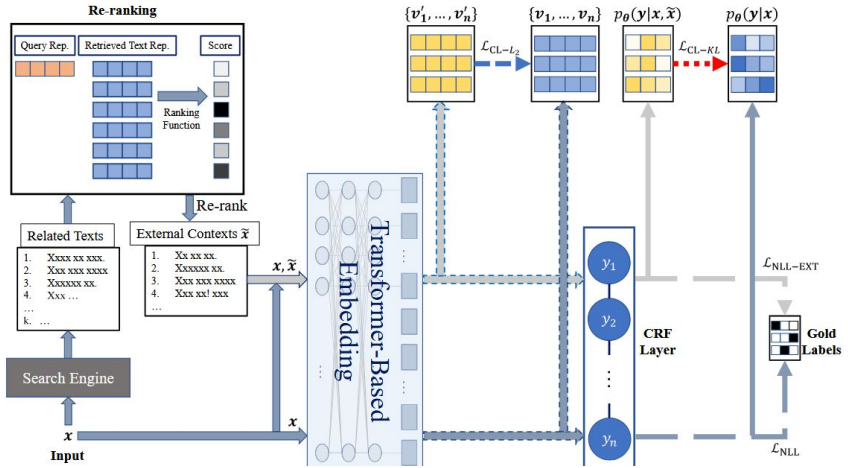
Answer: …generated answer…



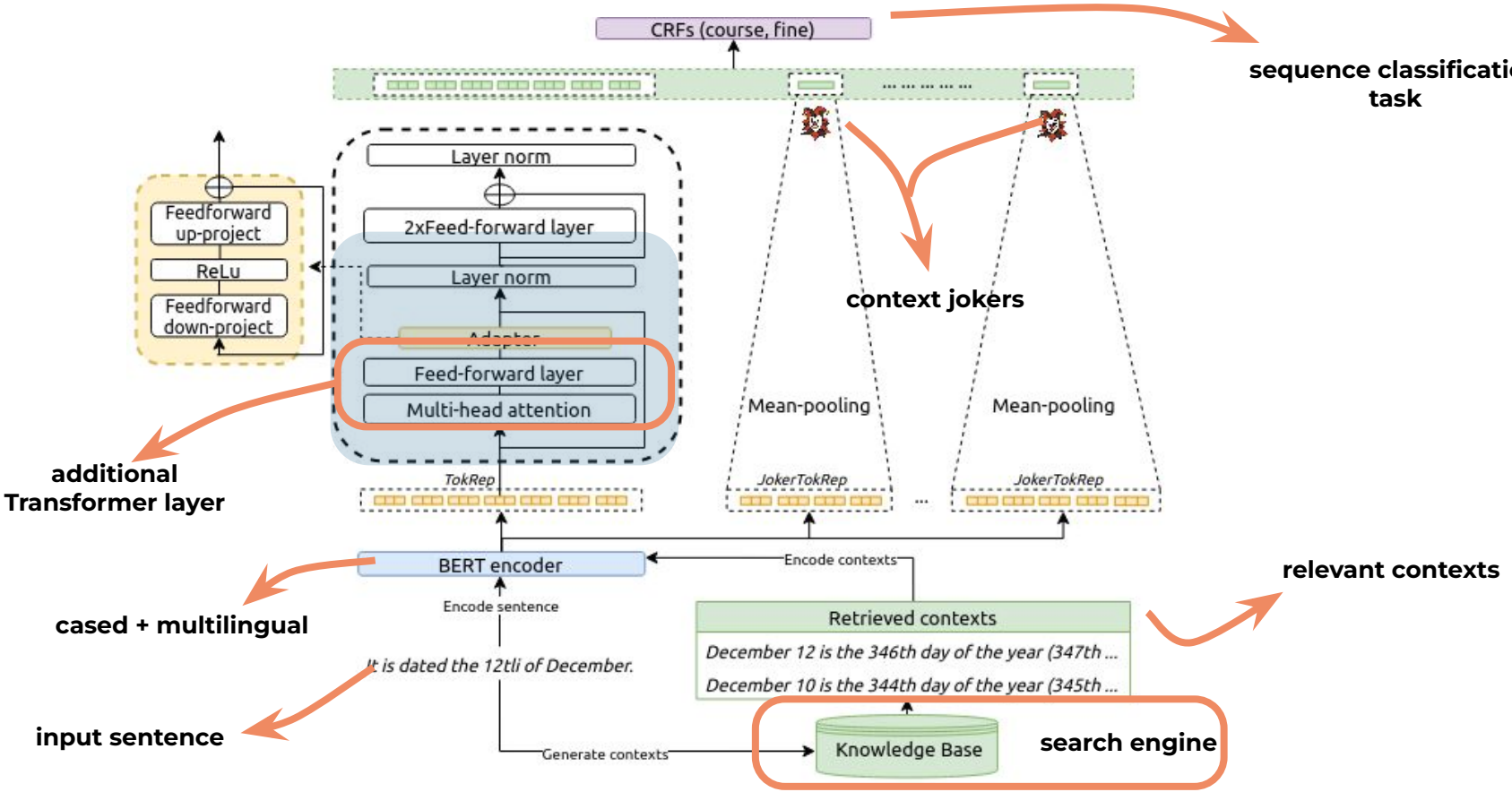

What's the angle of an equilateral triangle?

PLM <- Retriever

Answer: …extracted answer…



# NER Model Architecture






González-Gallardo, C. E., Tran, H. T. H., Hamdi, A., & Doucet, A. (2024, September). Leveraging open large language models for historical named entity recognition. ECIR2023

# Knowledge Base & temporal information integration

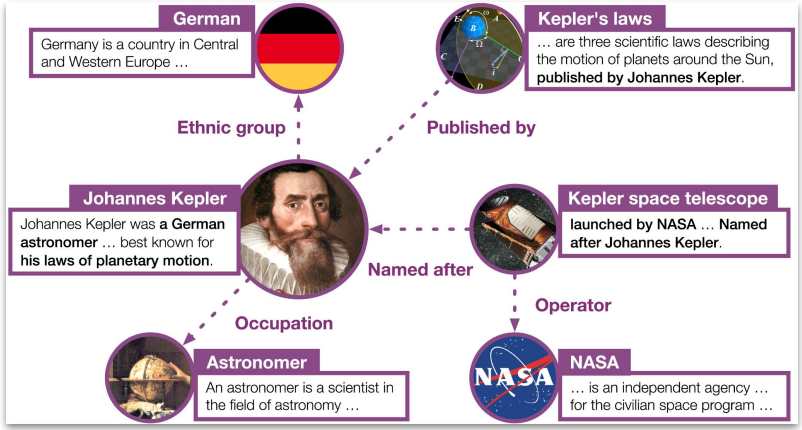
Wikidata5m [Wang X. et al., 2021]

- knowledge graph
- ~ 5M Wikidata entities in the general domain
- aligned to corresponding Wikipedia pages (1st paragraph)

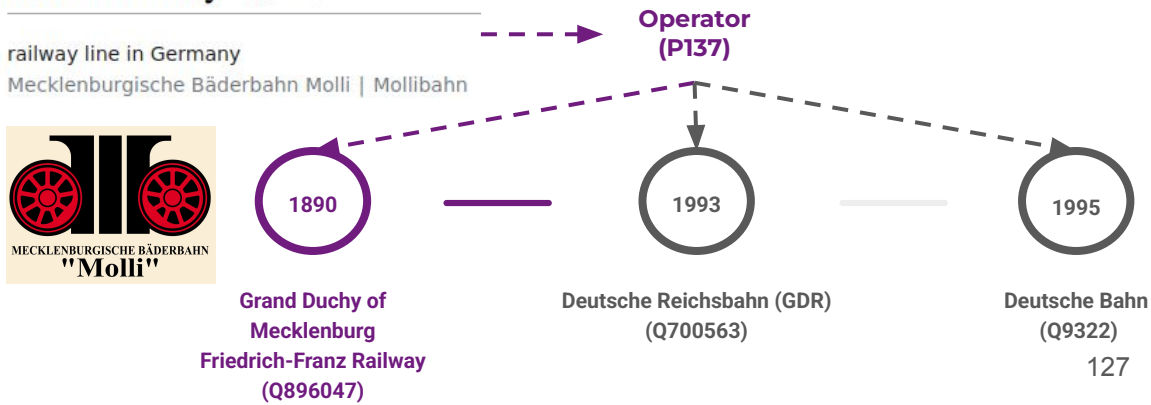
TKG [García-Durán A. et al., 2018]

- > 11k entities
- 150k time-related facts
- 508 - 2017 year scope
- multiple facts per entity → aggregation operator

**Wikidata5m**




**TKG**



# Configurations & evaluation

- `no-context`: model with no extra contexts
- `non-temporal`: *context jokers* integration with no time-related information
- `temporal-(10|25|50)`: *context jokers* integration with different year interval thresholds

- micro level precision (**P**), recall (**R**) & F-measure (**F1**)
- strict (***CS***) & fuzzy (***CF***) boundary matching

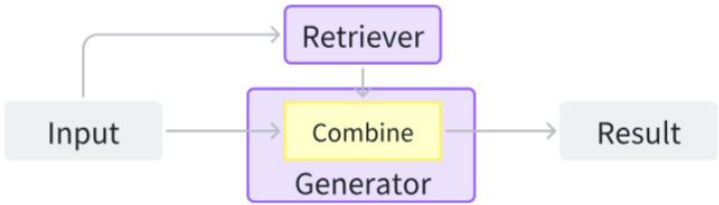
| | French | | | | | | German | | | | | | English | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | hipe-2020 | | | ajmc | | | hipe-2020 | | | ajmc | | | hipe-2020 | | | ajmc | | |
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| no-context | | | | | | | | | | | | | | | | | | |
| ***CS*** | 0.755 | 0.757 | 0.756 | 0.829 | 0.806 | 0.817 | 0.754 | 0.730 | 0.742 | 0.910 | 0.877 | 0.893 | 0.604 | 0.563 | 0.583 | 0.789 | 0.859 | 0.823 |
| ***CF*** | 0.857 | 0.859 | 0.858 | 0.883 | 0.858 | 0.870 | **0.853** | 0.826 | 0.839 | 0.935 | 0.901 | 0.917 | 0.778 | 0.726 | 0.751 | 0.855 | 0.931 | 0.891 |
| non-temporal | | | | | | | | | | | | | | | | | | |
| ***CS*** | 0.762 | **0.767** | **0.765** | 0.829 | 0.783 | 0.806 | 0.759 | **0.767** | **0.763** | **0.930** | 0.898 | 0.913 | 0.565 | 0.601 | 0.583 | 0.828 | 0.871 | 0.849 |
| ***CF*** | 0.862 | **0.869** | 0.866 | **0.906** | 0.856 | 0.880 | 0.847 | 0.856 | 0.852 | **0.949** | 0.916 | **0.932** | 0.741 | 0.788 | 0.764 | 0.885 | 0.931 | 0.908 |
| temporal-50 | | | | | | | | | | | | | | | | | | |
| ***CS*** | **0.765** | 0.765 | **0.765** | 0.839 | 0.822 | 0.830 | 0.748 | 0.756 | 0.752 | 0.921 | **0.911** | **0.916** | **0.643** | 0.617 | **0.630** | 0.855 | 0.882 | 0.868 |
| ***CF*** | **0.867** | 0.867 | **0.867** | 0.901 | 0.883 | 0.892 | 0.833 | 0.842 | 0.838 | 0.937 | **0.927** | **0.932** | **0.794** | 0.762 | 0.777 | 0.916 | **0.945** | 0.931 |
| temporal-25 | | | | | | | | | | | | | | | | | | |
| ***CS*** | 0.759 | 0.756 | 0.757 | **0.848** | **0.839** | **0.844** | 0.757 | 0.743 | 0.750 | 0.925 | 0.903 | 0.914 | 0.621 | 0.630 | 0.625 | 0.833 | 0.876 | 0.854 |
| ***CF*** | 0.863 | 0.859 | 0.861 | 0.902 | **0.892** | **0.897** | 0.852 | 0.835 | 0.843 | 0.938 | 0.916 | 0.927 | 0.787 | 0.800 | **0.793** | 0.893 | 0.940 | 0.916 |
| temporal-10 | | | | | | | | | | | | | | | | | | |
| ***CS*** | 0.762 | 0.764 | 0.763 | **0.848** | **0.839** | **0.844** | **0.760** | 0.765 | 0.762 | 0.917 | 0.898 | 0.907 | 0.605 | **0.646** | 0.625 | **0.866** | **0.888** | **0.877** |
| ***CF*** | 0.863 | 0.866 | 0.865 | 0.902 | **0.892** | **0.897** | 0.852 | **0.857** | **0.854** | 0.936 | 0.916 | 0.926 | 0.760 | **0.811** | 0.784 | **0.922** | **0.945** | **0.933** |

**Table 2.** Results on French, German and English, for the `hipe-2020` and `ajmc` datasets.

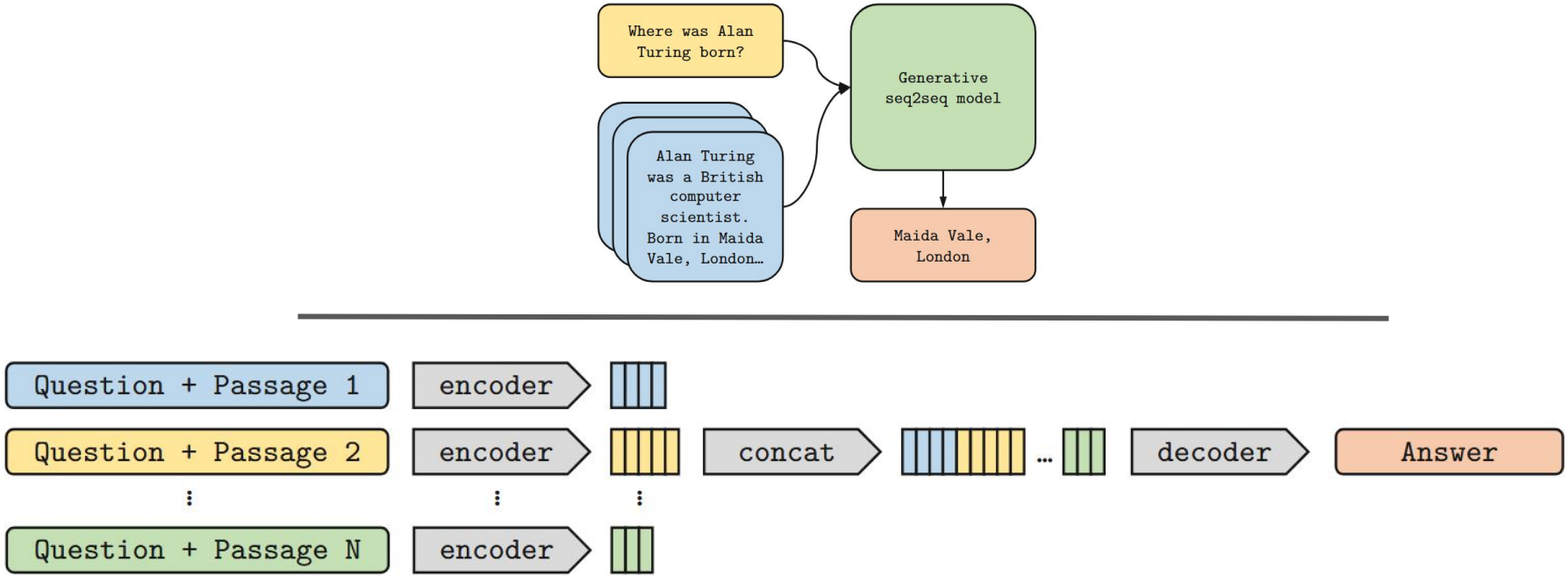
González-Gallardo, C. E., Tran, H. T. H., Hamdi, A., & Doucet, A. (2024, September). Leveraging open large language models for historical named entity recognition. ECIR2025

# Latent representation-based RAG

- In the latent representation-based RAG framework, retrieved objects are integrated as latent representations
- This integration enhances the model's understanding and improves the quality of the generated content
- It is also used in multimodality as it combines retriever and generator states, but required extra training to align latent spaces that integrate retrieved information

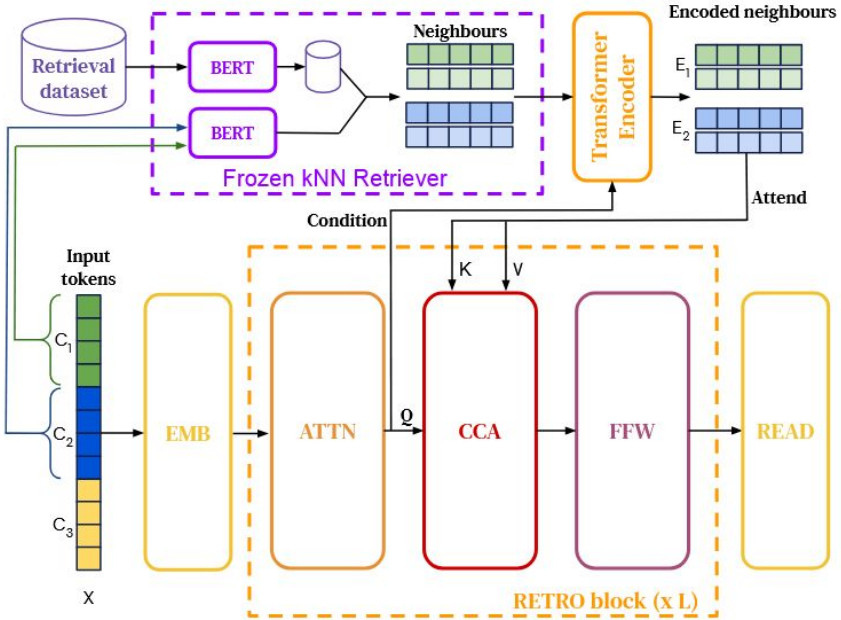


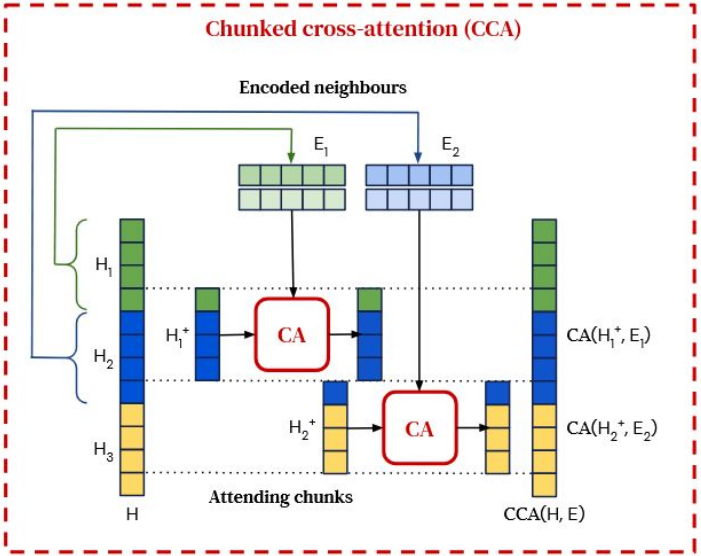
# Fusion-in-Decoder




G. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” in EACL, 2021.

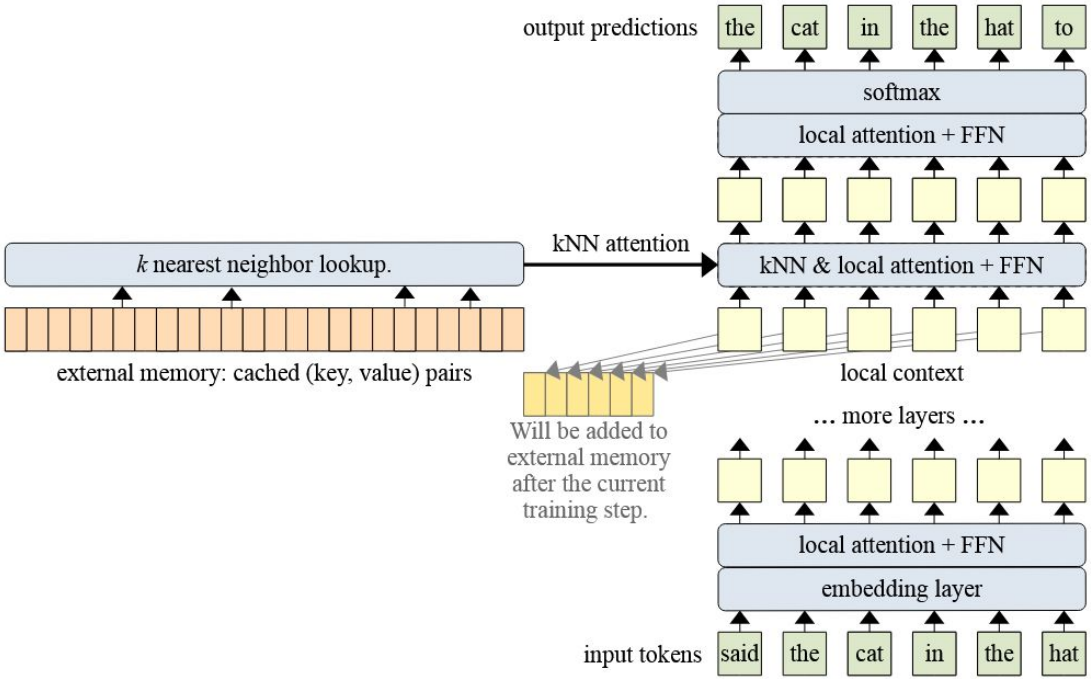
# RETRO




S. Borgeaud, A. Mensch et al., “Improving language models by retrieving from trillions of tokens,” in ICML, 2022.

# Memorizing transformers



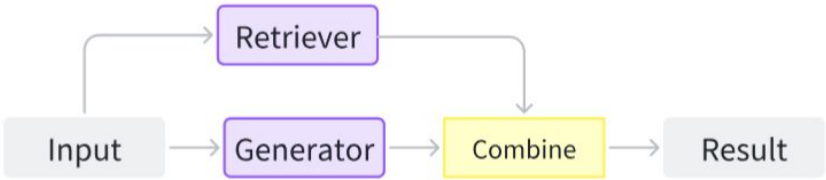

Y. Wu, M. N. Rabe, D. Hutchins, and C. Szegedy, “Memorizing transformers,” in ICLR, 2022.

# Logit-based RAG

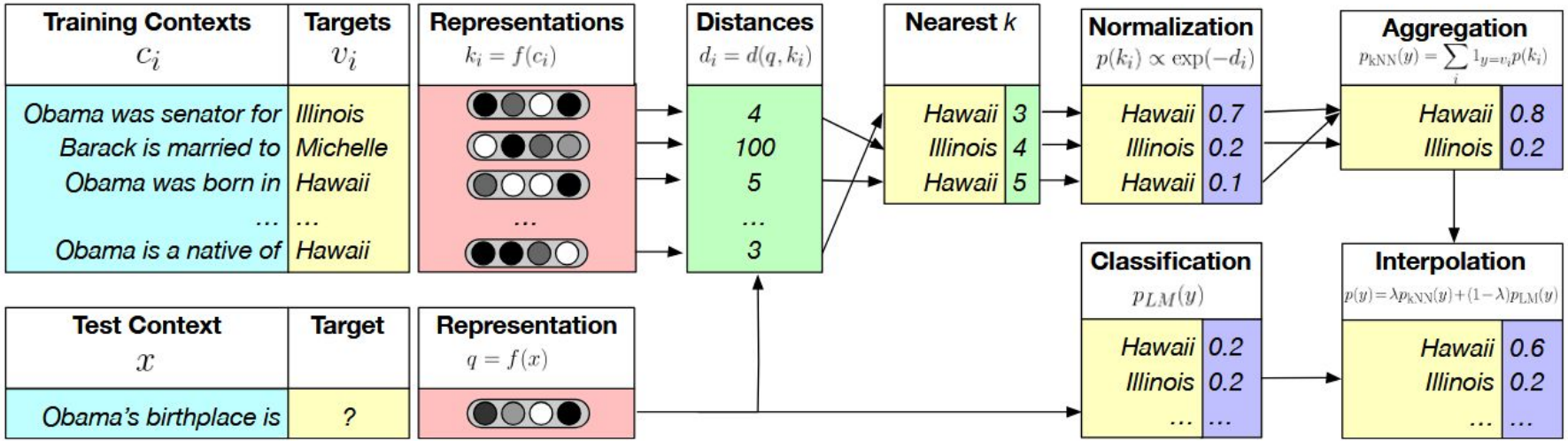
Generative models integrate retrieval information through logits during the decoding process

Logits are combined through simple summation or models to compute the probabilities for step-wise generation

Beyond text, other modalities, such as code and image, also leverage logit-based RAG

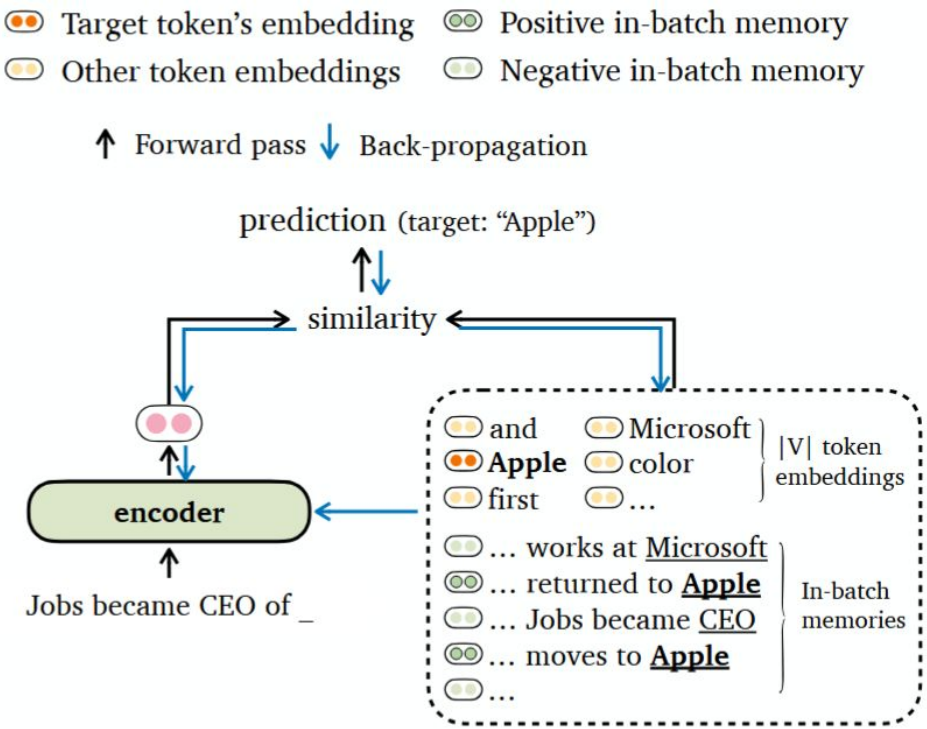


# kNN-LM



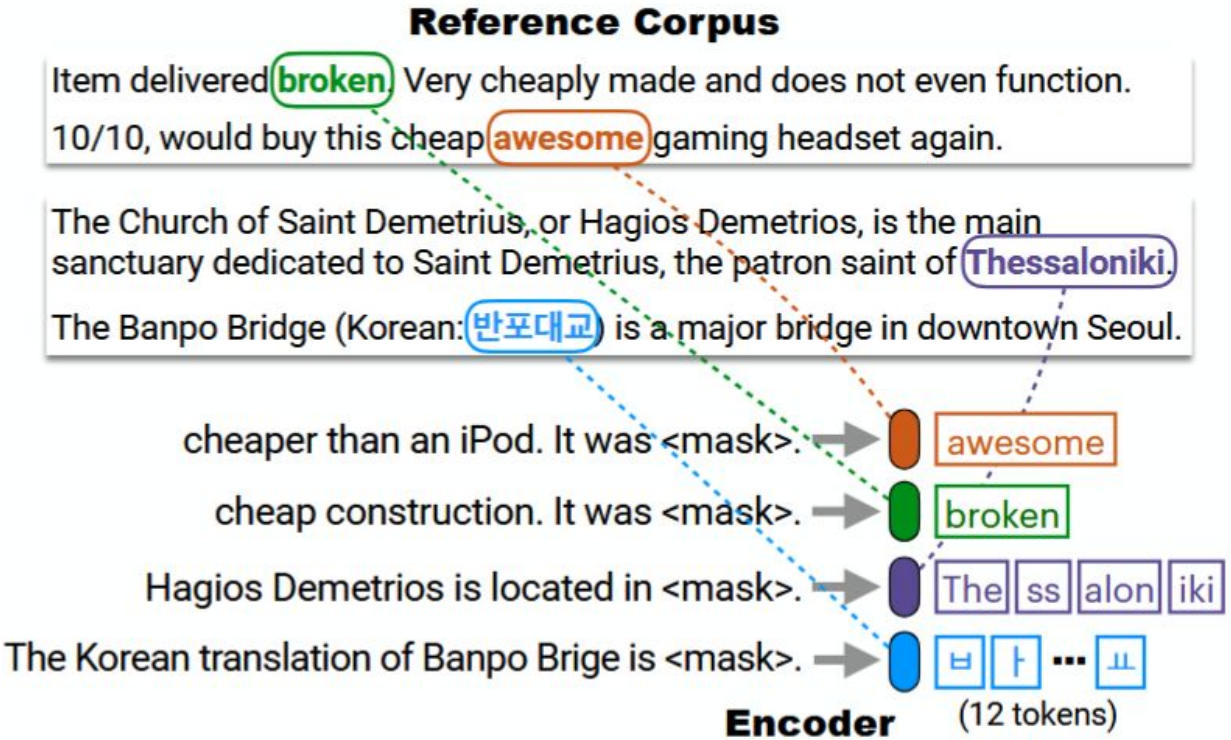

U. Khandelwal, O. Levy, D. Jurafsky et al., "Generalization through memorization: Nearest neighbor language models," in ICLR, 2020

# TRIME




Z. Zhong, T. Lei, and D. Chen, "Training language models with memory augmentation," in EMNLP, 2022.

# NPM



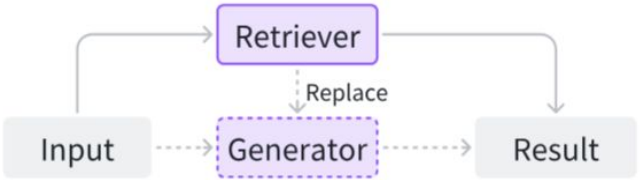

S. Min, W. Shi, M. Lewis et al., "Nonparametric masked language modeling," in ACL Findings, 2023

# Speculative RAG

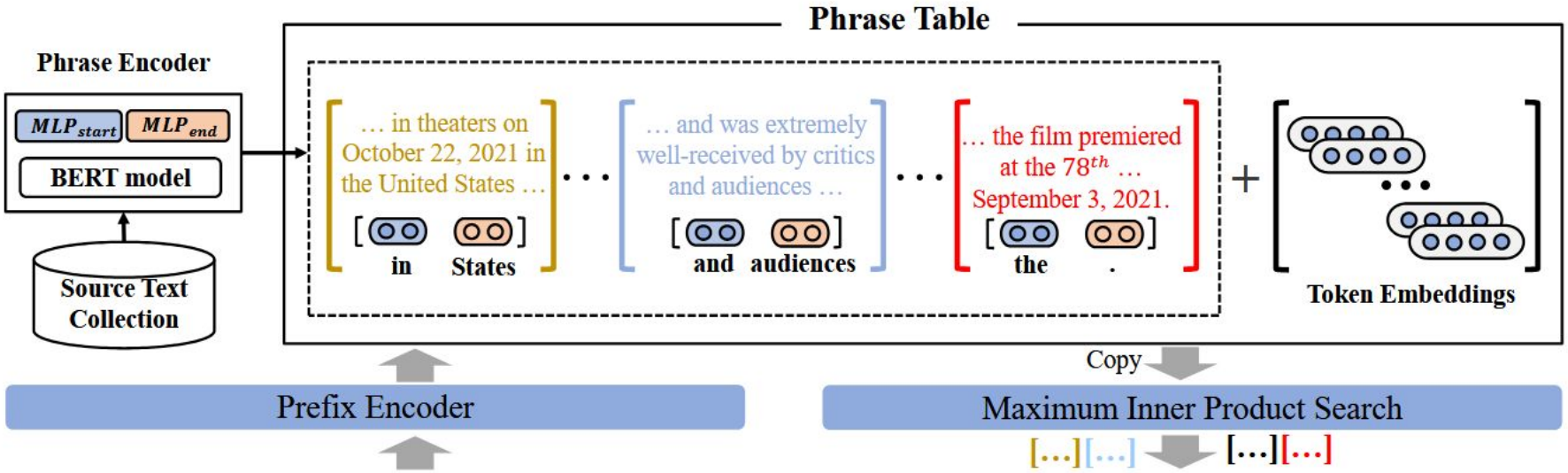
Speculative RAG aims to leverage retrieval over pure generation, with the goal of saving resources and speeding up response times

It decouples the generator and the retriever, enabling the direct use of pre-trained models as components

In this paradigm, we can explore various strategies to make the to effectively use the retrieved content



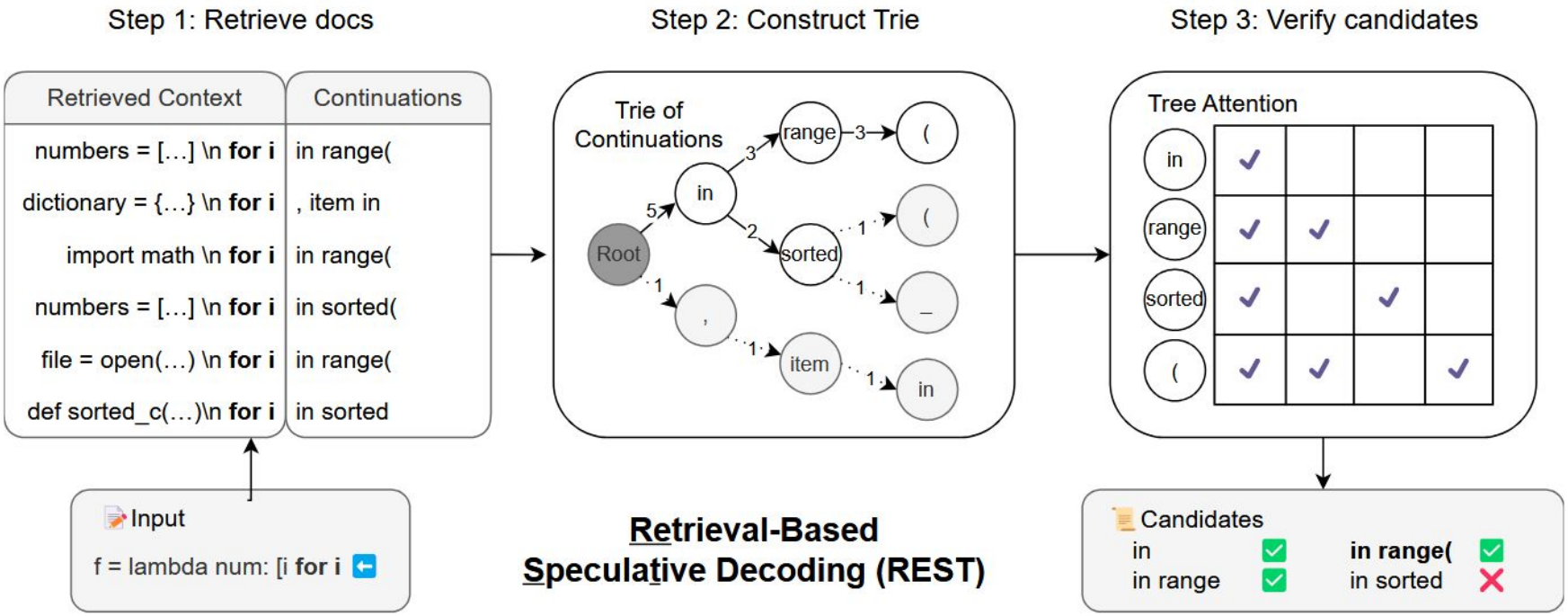
# COG




***The Dune film was released*** [in theaters on October 22, 2021 in the United States] [and was extremely well-received by critics and audiences] **[Before] [that] [,]** [the film premiered at the $78^{th}$ International Film Festival on September 3, 2021.]

T. Lan, D. Cai, Y. Wang et al., "Copy is all you need," in ICLR, 2023.

# REST




Z. He, Z. Zhong, T. Cai et al., "REST: retrieval-based speculative decoding," arxiv:2311.08252, 2023

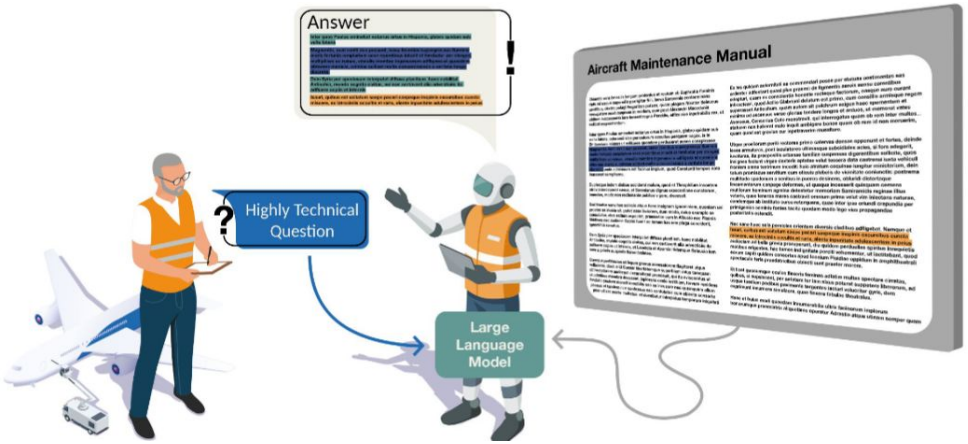
# SEBRAG on Aircraft maintenance data (technical domain)

Operators in aircraft maintenance rely on a large number of certified manuals to ensure safety and effectiveness

Reliable QA systems can help speed up access to information

In aircraft maintenance, all procedures come from certified manuals. Any deviation, no matter how small, is a potential safety risk.

We need to guarantee that answers are extracted exclusively from these trusted documents




Signé, Q., Boughanem, M., Moreno, J. G., & Belkacem, T. (2025, July). A Substring Extraction-Based RAG Method for Minimising Hallucinations in Aircraft Maintenance Question Answering. ICTIR2025

# SEBRAG

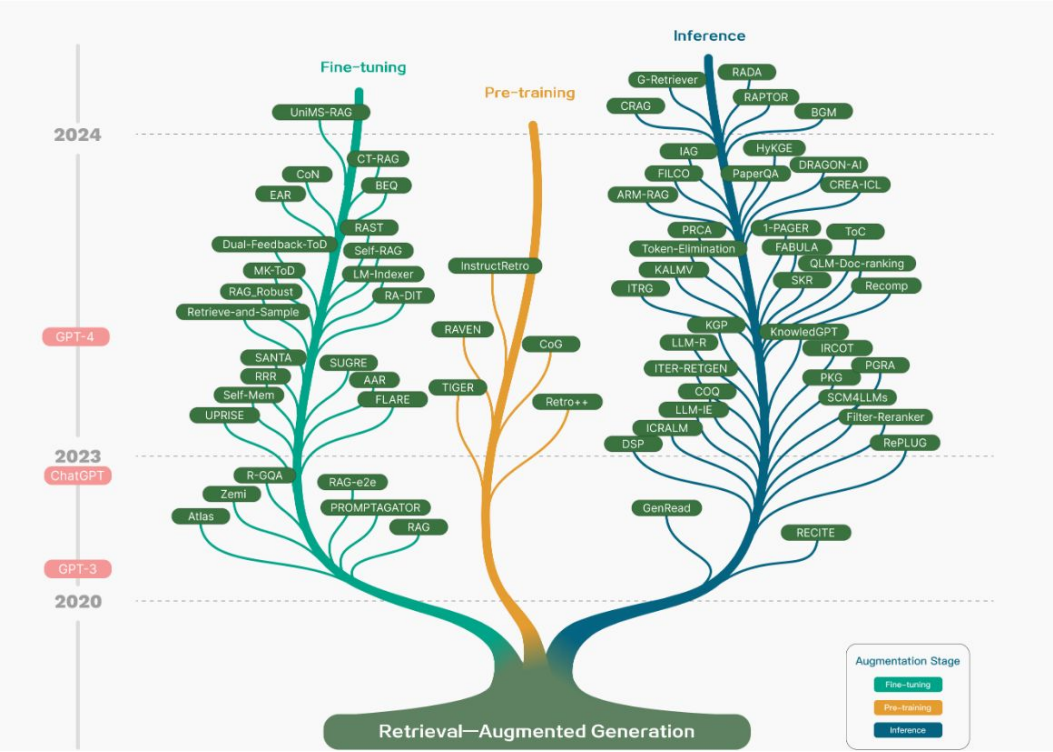
| Question | Context | Answer |
| --- | --- | --- |
| *What is this aircraft used for?* | <pre>{<br>  "doc1": {<br>  "content":<br>  "The aircraft WT9 Dynamic LSA / Club is a single engine, two seat (arranged side by side), cantilever low wing aircraft with a cruciform tail. [...] The aircraft is intended for sporting, recreation and tourist flying and is approved for VFR day operation only.",<br>  "numbering": {<br>  1: "The", 2: "aircraft", ..., 149: "sporting,",<br>  150: "recreation", 151: "and", 152: "tourist",<br>  153: "flying", ..., 161: "only."<br>  }<br>  }<br>}</pre> | <pre>{<br>  "name": "substring",<br>  "parameters": {<br>  "$id_{\text{document}}$": "doc1",<br>  "$pos_{\text{first word}}$": 149,<br>  "$pos_{\text{last word}}$": 153<br>  }<br>}</pre> |

Substring Extraction Based RAG (SEBRAG)

Signé, Q., Boughanem, M., Moreno, J. G., & Belkacem, T. (2025, July). A Substring Extraction-Based RAG Method for Minimising Hallucinations in Aircraft Maintenance Question Answering. ICTIR2025

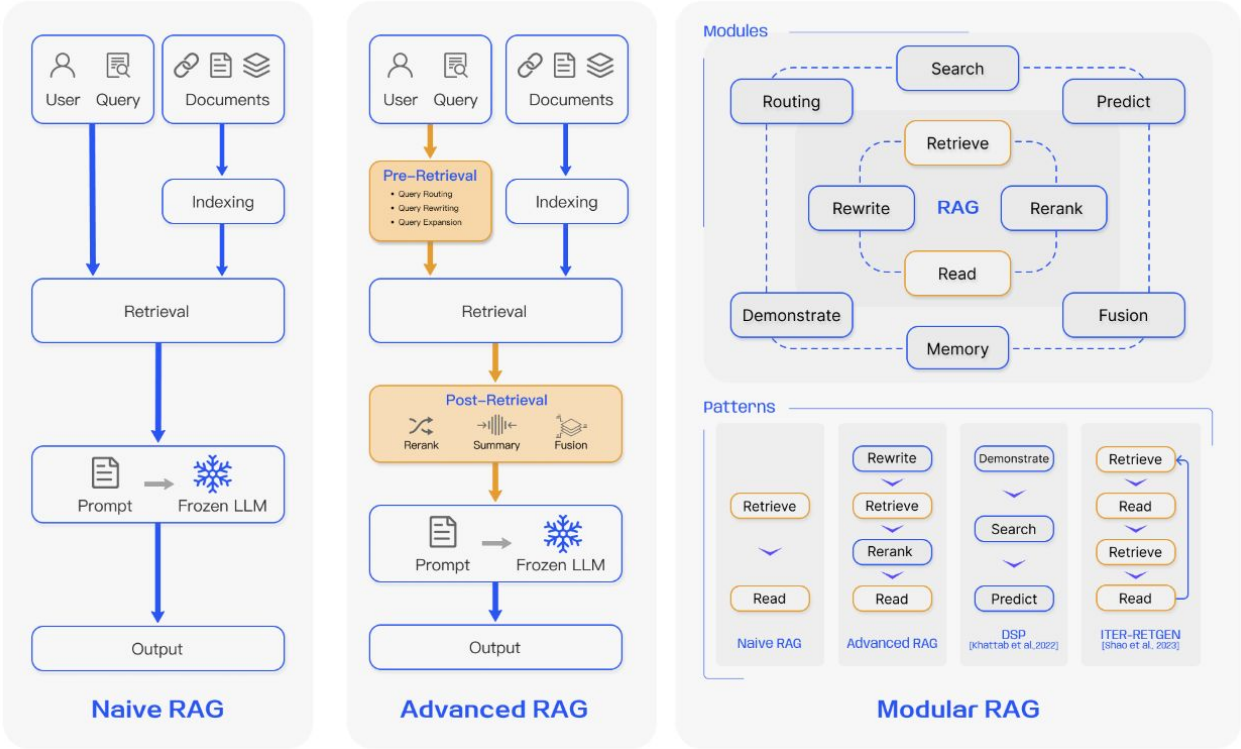
but there are more complex RAG, isn’t it?

# Yes! RAG is not only naive-RAG




Source: https://arxiv.org/pdf/2312.10997

# Naive vs Advanced RAG and beyond




Source: https://arxiv.org/pdf/2312.10997

# What additional benefits does augmentation provide?

# Additional benefits of Augmented LMs

**Modularity**

We can change external memory and update the model's knowledge on test time.

**Attribution**

We can trace back the information (documents)

that the generated answer is based on

**Parameter efficiency**

We can leverage external memory to reduce the number of implicit parameters of the LM without compromising performance.

# Attribution

Required for:

- **Explaining** the response process
- **Verifying** information in responses
- Increasing **trust** in AI

An ideal attribution system:

- Provide **correct** attributions
- Give attributions that accurately **reflect** the AI's thought process
- No (or **low**) **latency**

Existing attribution methods

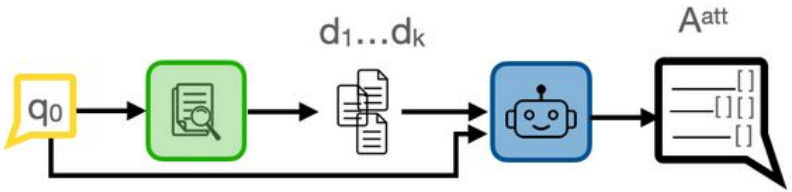
- Let AI generate attributions -> Black box, justifies rather than accurately reflects the model “thinking”

- Use another AI to perform “post-hoc” attribution -> Additional cost, latency during execution

➔In addition to poor performance, current existing methods have significant problems!

# Existing attribution methods

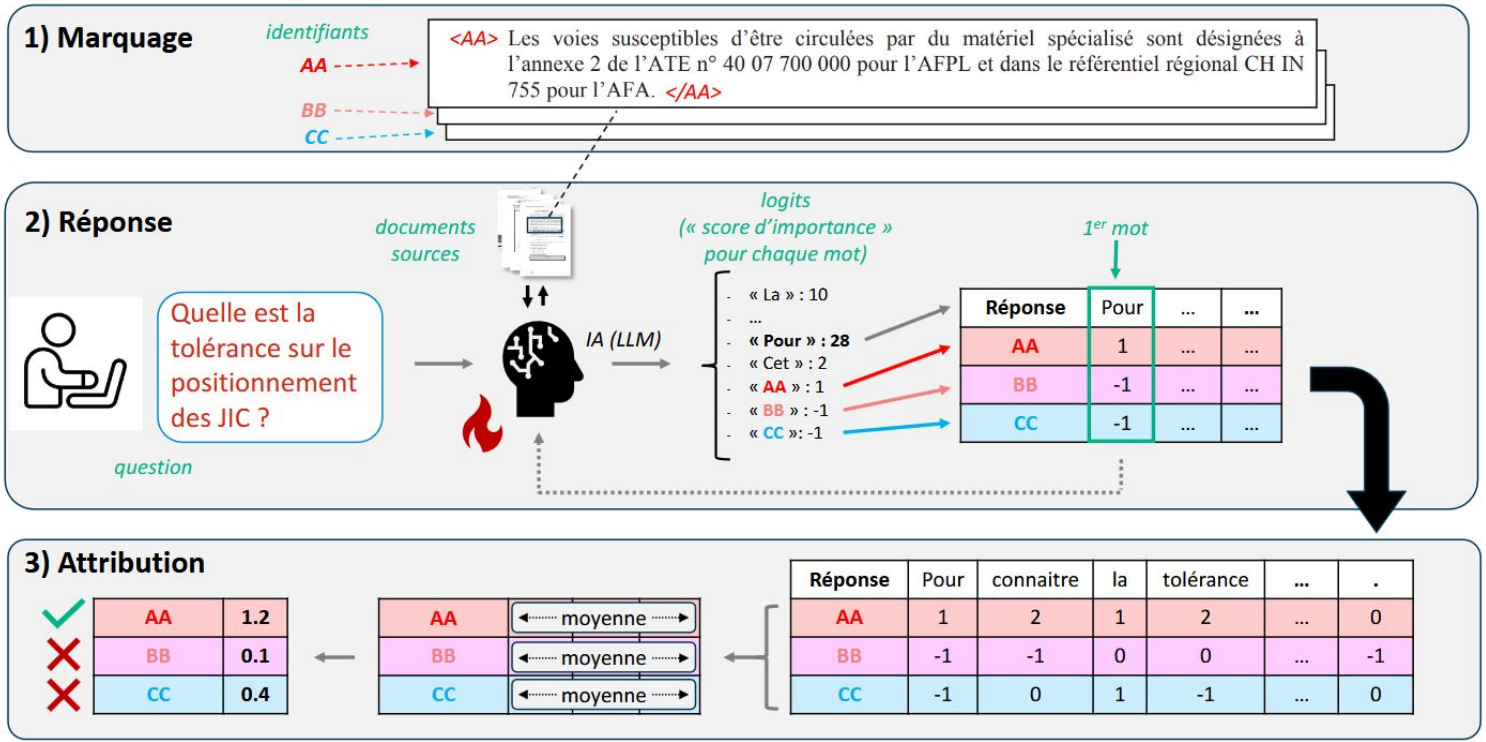
- Retrieve-Then-Generate (RTG)
- Let AI generate attributions
- Black box, justifies rather than accurately reflects the model “thinking”




- Generate-Then-Retrieve (GTR)
- Use another AI to perform “post-hoc” attribution
- Additional cost, latency during execution


Djeddal, H., Erbacher, P., Toukal, R., Soulier, L., Pinel-Sauvagnat, K., Katrenko, S., & Tamine, L. (2024, October). An evaluation framework for attributed information retrieval using large language models. CIKM2024

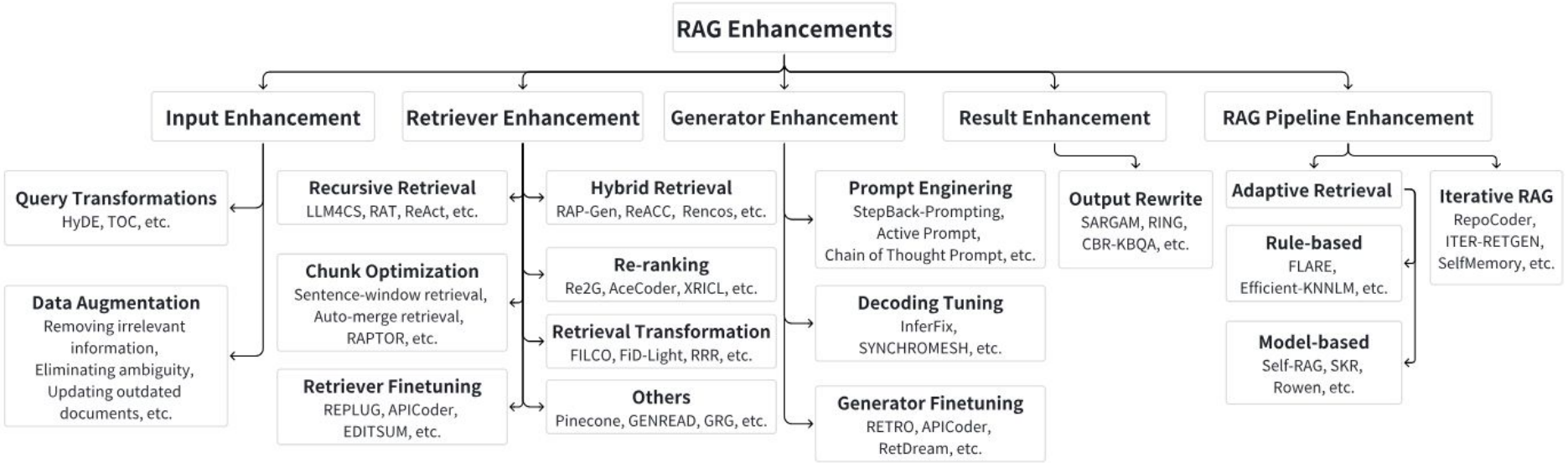
# Jointly Generating and Attributing Answers






Albarede, L., Moreno, J., Tamine, L., & Lefeuvre, L. (2025). Jointly Generating and Attributing Answers using Logits of Document-Identifier Tokens. arXiv preprint arXiv:2508.08942.

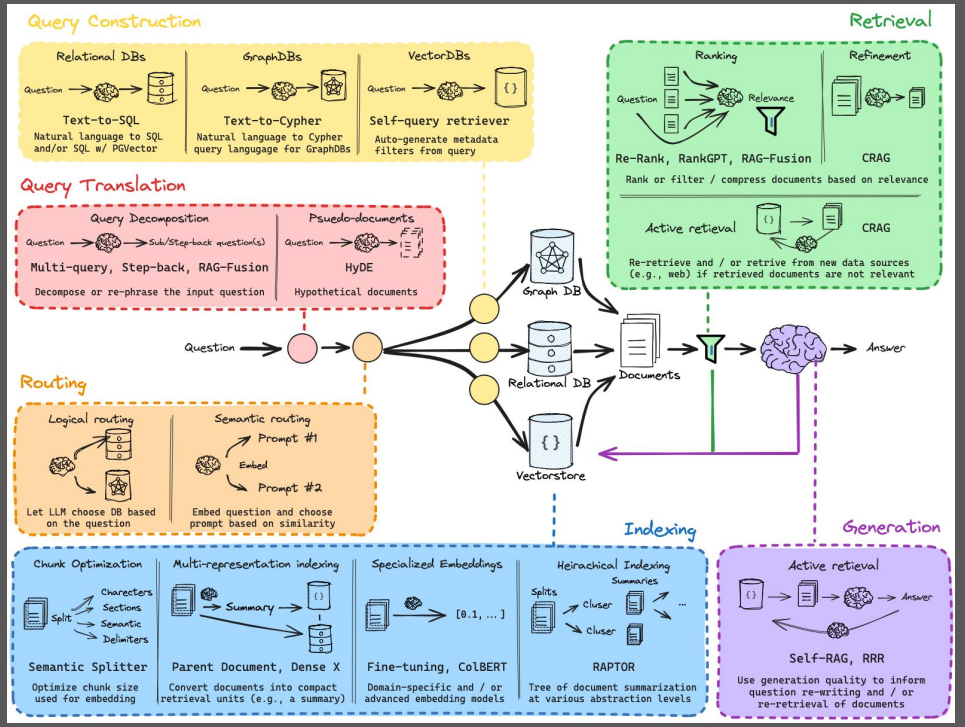
# What and how enhance our RAG?




Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., ... & Cui, B. (2024). Retrieval-augmented generation for ai-generated content: A survey. arXiv preprint arXiv:2402.19473.

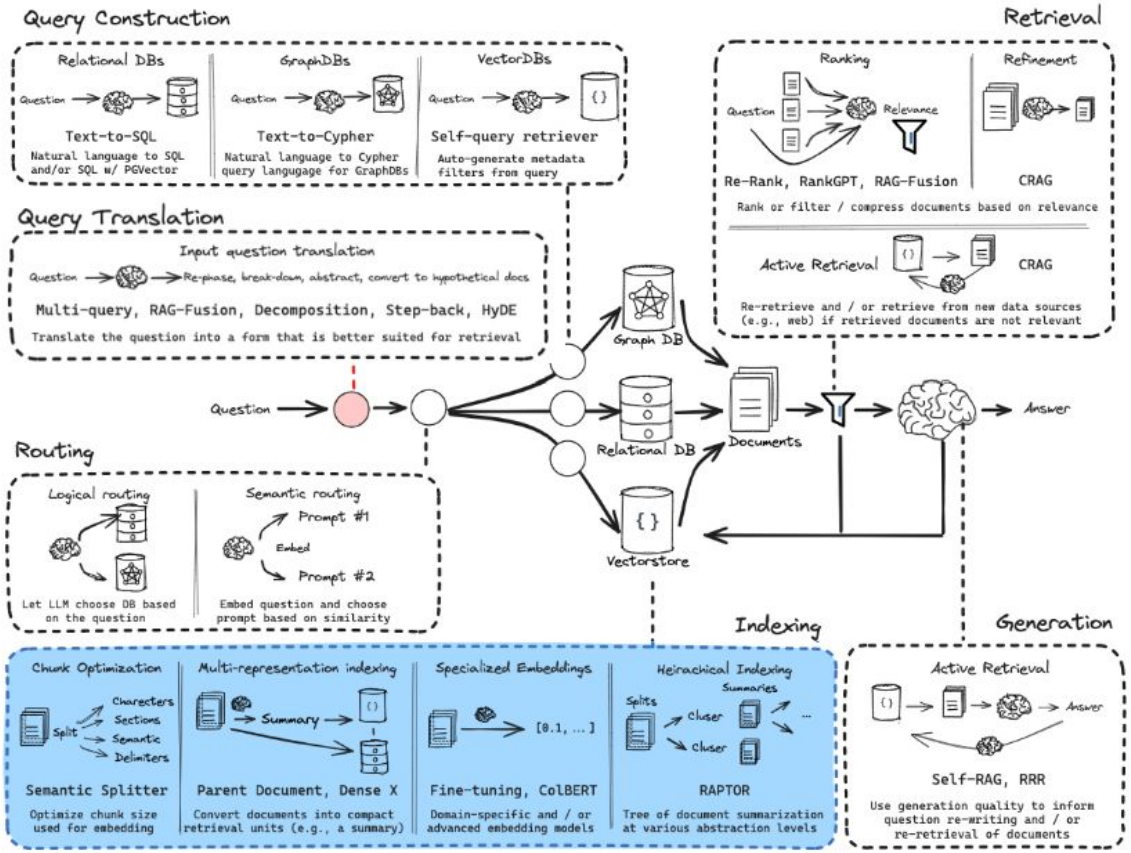
# Revisiting advanced concepts in RAG with Langchain

- Indexing
- Retrieval
- Query Construction
- Query Translation
- Routing
- Generation

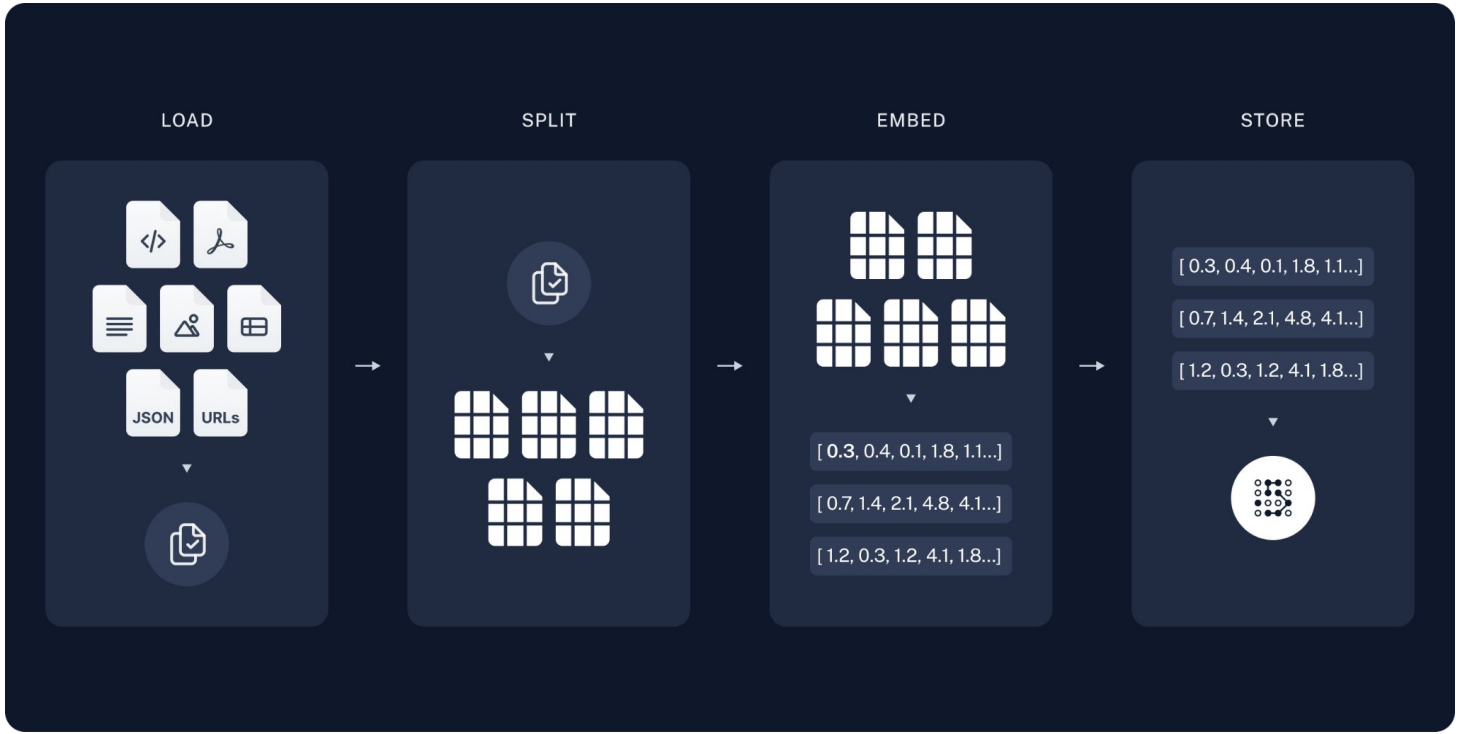



Main source for following slides: https://python.langchain.com/docs/tutorials/rag/

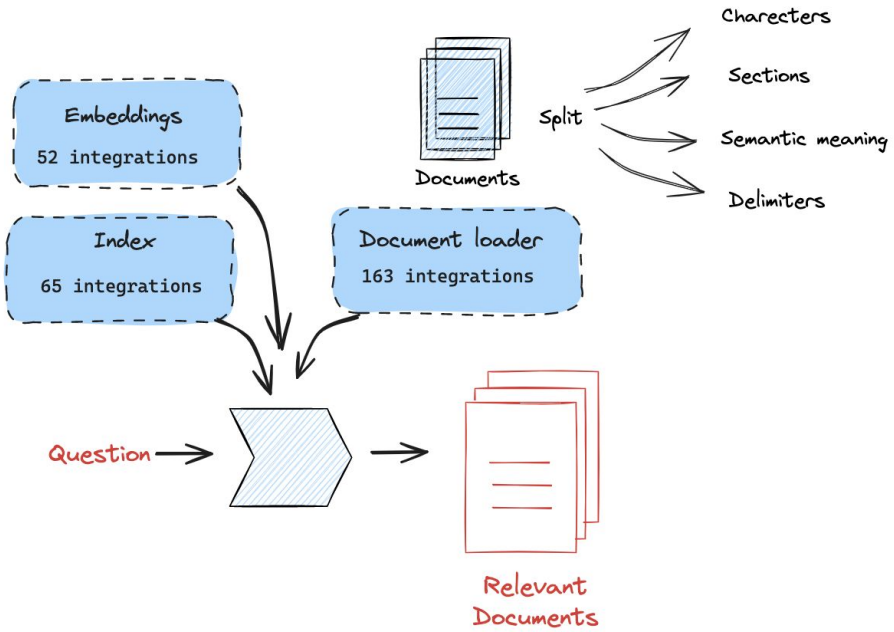
# Indexing


Query Construction
Relational DBs
Question
Text-to-SQL
Natural language to SQL and/or SQL w/ PGVector
GraphDBs
Question
Text-to-Cypher
Natural language to Cypher query langugage for GraphDBs
VectorDBs
Question
{}
Self-query retriever
Auto-generate metadata filters from query
Retrieval
Ranking
Question
Relevance
Re-Rank, RankGPT, RAG-Fusion
Refinement
CRAG
Rank or filter / compress documents based on relevance
Active Retrieval
{}
CRAG
Re-retrieve and / or retrieve from new data sources (e.g., web) if retrieved documents are not relevant
Query Translation
Input question translation
Question
Re-phase, break-down, abstract, convert to hypothetical docs
Multi-query, RAG-Fusion, Decomposition, Step-back, HyDE
Translate the question into a form that is better suited for retrieval
Graph DB
Question
Relational DB
Documents
Answer
Vectorstore
Routing
Logical routing
Let LLM choose DB based on the question
Semantic routing
Prompt #1
Embed
Prompt #2
Embed question and choose prompt based on similarity
Indexing
Generation
Chunk Optimization
Split
Characters
Sections
Semantic
Delimiters
Semantic Splitter
Optimize chunk size used for embedding
Multi-representation indexing
Summary
{}
Parent Document, Dense X
Convert documents into compact retrieval units (e.g., a summary)
Specialized Embeddings
[0.1, ...]
Fine-tuning, ColBERT
Domain-specific and / or advanced embedding models
Heirachical Indexing
Splits
Summaries
Cluser
Cluser
...
RAPTOR
Tree of document summarization at various abstraction levels
Active Retrieval
{}
Answer
Self-RAG, RRR
Use generation quality to inform question re-writing and / or re-retrieval of documents

# Indexing

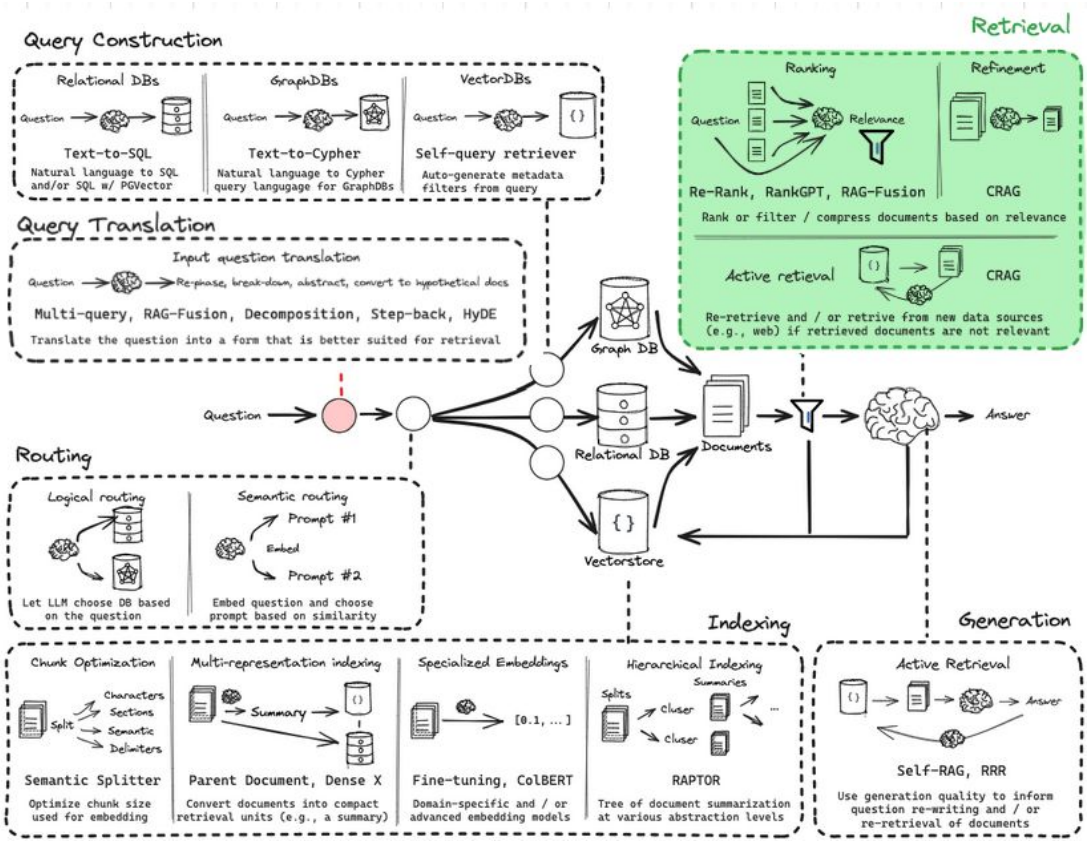

LOAD
SPLIT
EMBED
STORE
JSON
URLs
[ 0.3, 0.4, 0.1, 1.8, 1.1...]
[ 0.7, 1.4, 2.1, 4.8, 4.1...]
[ 1.2, 0.3, 1.2, 4.1, 1.8...]
[ 0.3, 0.4, 0.1, 1.8, 1.1...]
[ 0.7, 1.4, 2.1, 4.8, 4.1...]
[ 1.2, 0.3, 1.2, 4.1, 1.8...]

# LangChain has many integrations to support this



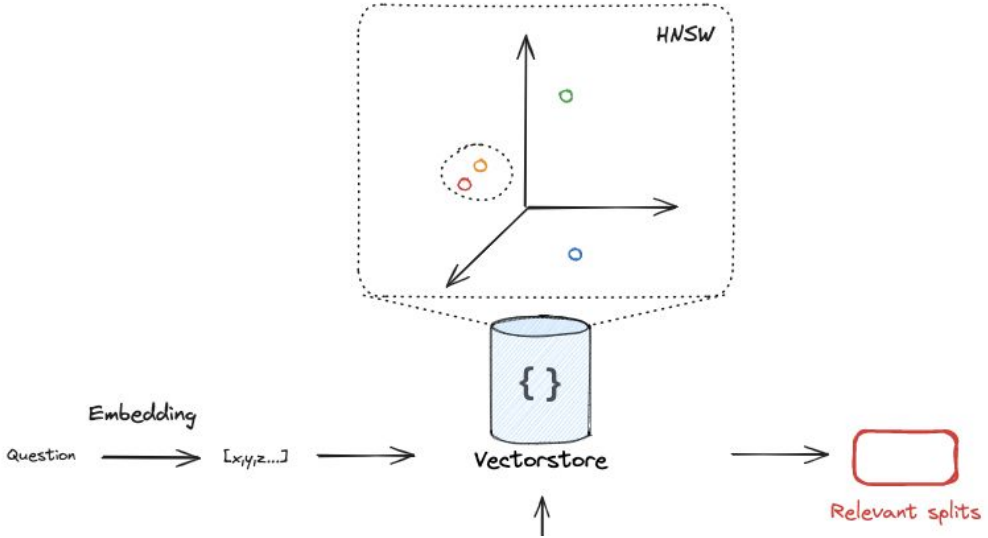

https://integrations.langchain.com/
https://vdbs.superlinked.com/

# Retrival


Query Construction
Relational DBs
Question
Text-to-SQL
Natural language to SQL and/or SQL w/ PGVector
GraphDBs
Question
Text-to-Cypher
Natural language to Cypher query langugage for GraphDBs
VectorDBs
Question
{}
Self-query retriever
Auto-generate metadata filters from query
Query Translation
Input question translation
Question
Re-phase, break-down, abstract, convert to hypothetical docs
Multi-query, RAG-Fusion, Decomposition, Step-back, HyDE
Translate the question into a form that is better suited for retrieval
Routing
Logical routing
Let LLM choose DB based on the question
Semantic routing
Prompt #1
Embed
Prompt #2
Embed question and choose prompt based on similarity
Retrieval
Ranking
Question
Relevance
Re-Rank, RankGPT, RAG-Fusion
Refinement
CRAG
Rank or filter / compress documents based on relevance
Active retieval
{}
CRAG
Re-retrieve and / or retrive from new data sources (e.g., web) if retrieved documents are not relevant
Graph DB
Question
Relational DB
Documents
Answer
{}
Vectorstore
Indexing
Chunk Optimization
Split
Characters
Sections
Semantic
Delimiters
Semantic Splitter
Optimize chunk size used for embedding
Multi-representation indexing
Summary
{}
Parent Document, Dense X
Convert documents into compact retrieval units (e.g., a summary)
Specialized Embeddings
[0.1, ...]
Fine-tuning, ColBERT
Domain-specific and / or advanced embedding models
Hierarchical Indexing
Summaries
Splits
cluser
cluser
...
RAPTOR
Tree of document summarization at various abstraction levels
Generation
Active Retrieval
{}
Answer
Self-RAG, RRR
Use generation quality to inform question re-writing and / or re-retrieval of documents

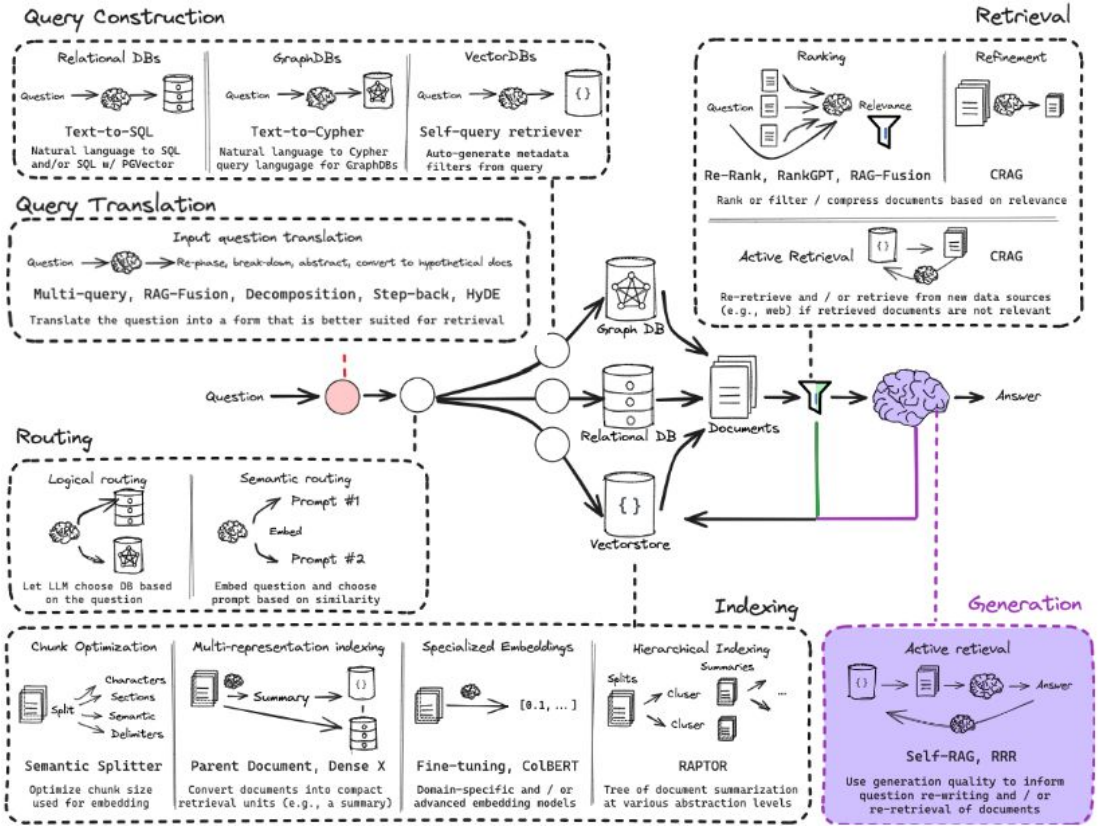
# Vectorstores add documents to context

Chroma

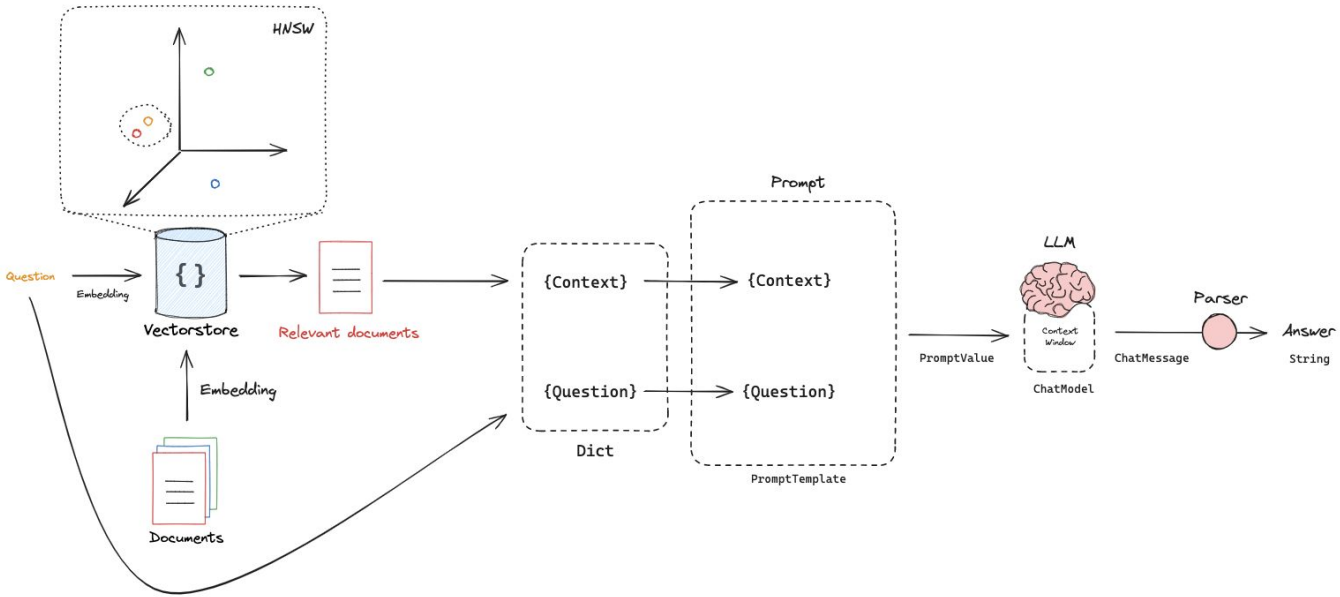



https://www.pinecone.io/learn/series/faiss/hnsw/

# Generation


Query Construction
Relational DBs
Question
Text-to-SQL
Natural language to SQL and/or SQL w/ PGVector
GraphDBs
Question
Text-to-Cypher
Natural language to Cypher query langugage for GraphDBs
VectorDBs
Question
{ }
Self-query retriever
Auto-generate metadata filters from query
Query Translation
Input question translation
Question
Re-phase, break-down, abstract, convert to hypothetical docs
Multi-query, RAG-Fusion, Decomposition, Step-back, HyDE
Translate the question into a form that is better suited for retrieval
Routing
Logical routing
Let LLM choose DB based on the question
Semantic routing
Prompt #1
Embed
Prompt #2
Embed question and choose prompt based on similarity
Question
Graph DB
Relational DB
{ }
Vectorstore
Documents
Answer
Retrieval
Ranking
Question
Relevance
Re-Rank, RankGPT, RAG-Fusion
Refinement
CRAG
Rank or filter / compress documents based on relevance
Active Retrieval
{ }
CRAG
Re-retrieve and / or retrieve from new data sources (e.g., web) if retrieved documents are not relevant
Indexing
Chunk Optimization
Split
Characters
Sections
Semantic
Delimiters
Semantic Splitter
Optimize chunk size used for embedding
Multi-representation indexing
{ }
Summary
Parent Document, Dense X
Convert documents into compact retrieval units (e.g., a summary)
Specialized Embeddings
[0.1, ... ]
Fine-tuning, ColBERT
Domain-specific and / or advanced embedding models
Hierarchical Indexing
Summaries
Splits
cluser
cluser
RAPTOR
Tree of document summarization at various abstraction levels
Generation
Active retieval
{ }
Answer
Self-RAG, RRR
Use generation quality to inform question re-writing and / or re-retrieval of documents

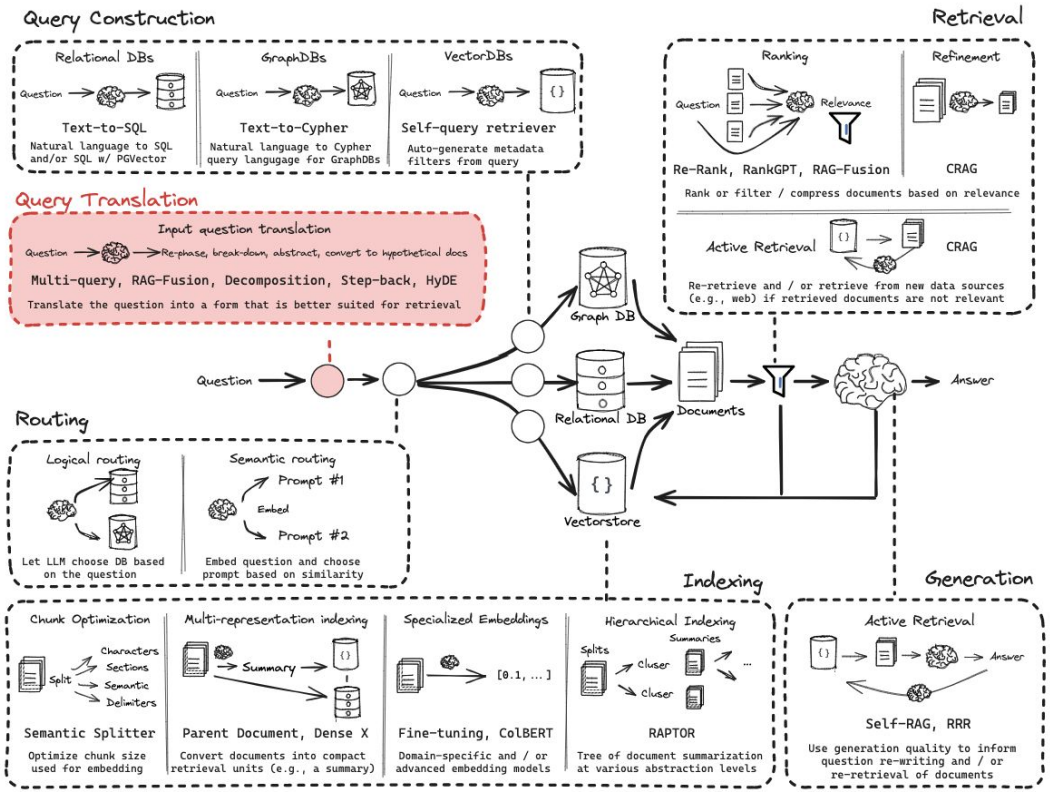
# Connecting retrieval with LLMs via prompt



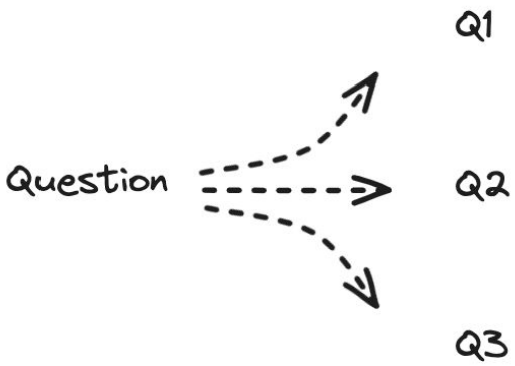

https://python.langchain.com/docs/expression_language/get_started

https://smith.langchain.com/hub/rlm/rag-prompt?organizationId=1fa8b1f4-fcb9-4072-9aa9-983e35ad61b8

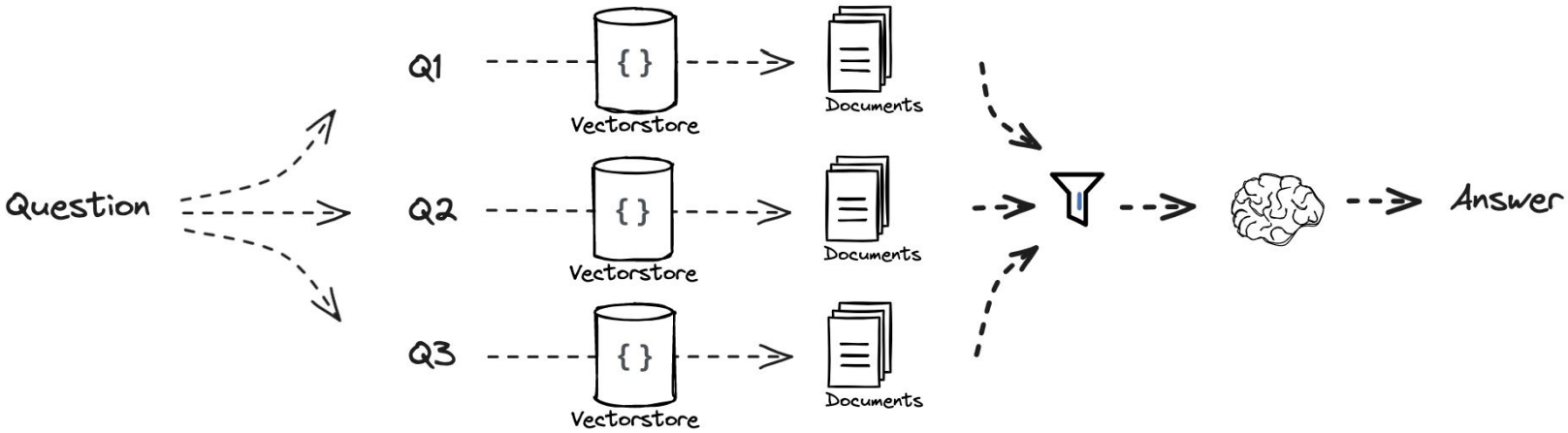
# Query Translation



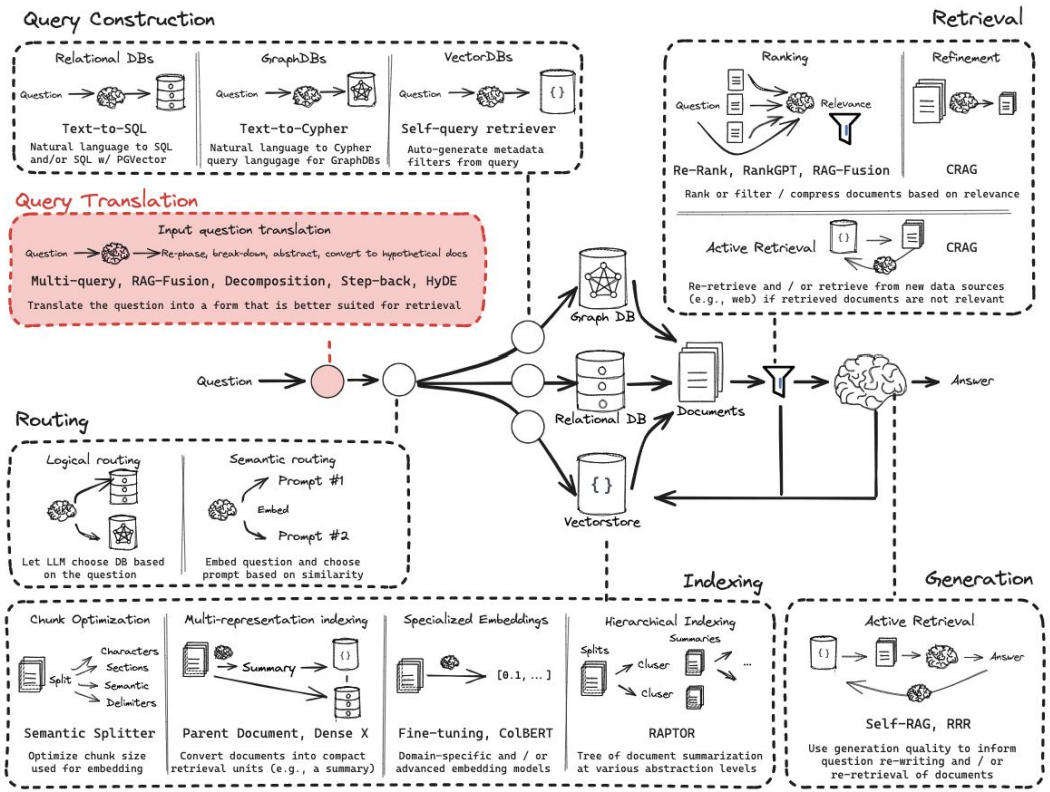
# Transform a question into multiple perspectives



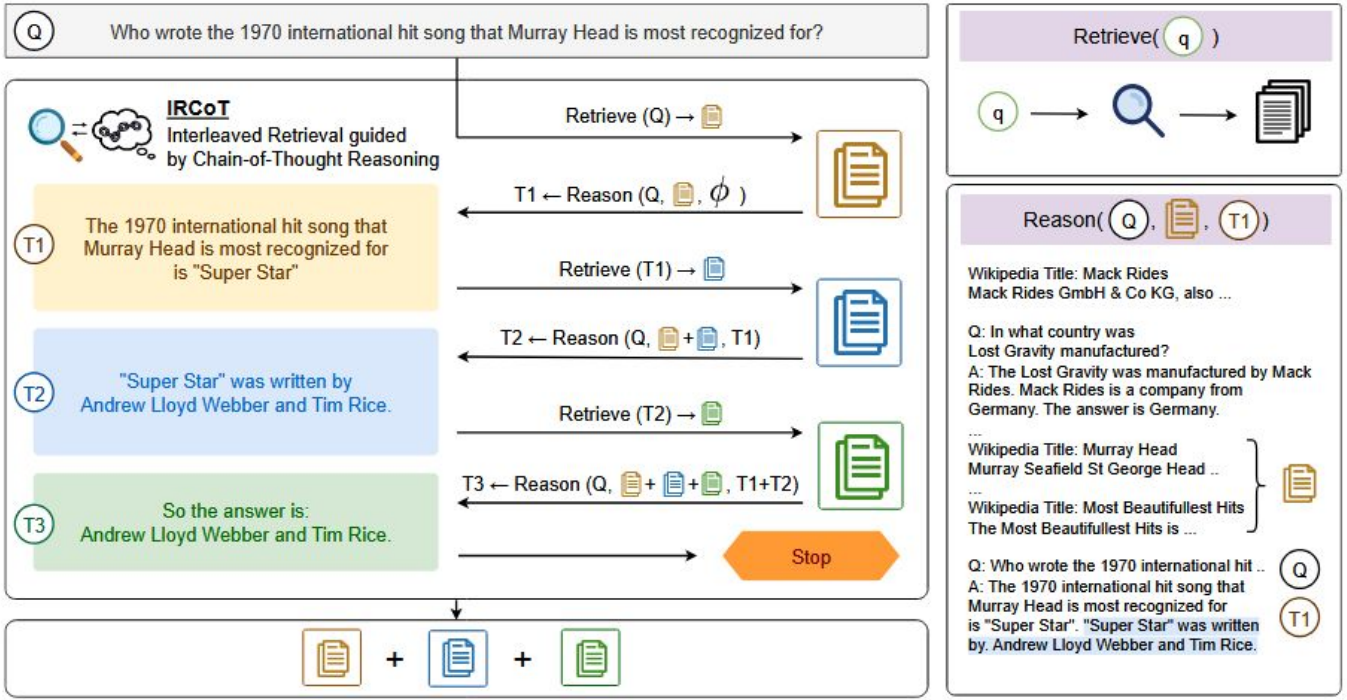
# Improve search, parallelized retrieval, and produce consolidated ranking



# Query Translation



# Intervaled Retrieval guided by Chain-of-Thought (IR-CoT)




https://arxiv.org/pdf/2212.10509.pdf

# Take a step back

| Knowledge QA Final-Answer Prompt |
| --- |
| You are an expert of world knowledge. I am going to ask you a question. Your response should be comprehensive and not contradicted with the following context if they are relevant. Otherwise, ignore them if they are not relevant.<br><br><Passage from original retrieval augmentation><br><Passage from step-back retrieval augmentation><br><br>Original Question: <Original Question><br>Answer: |

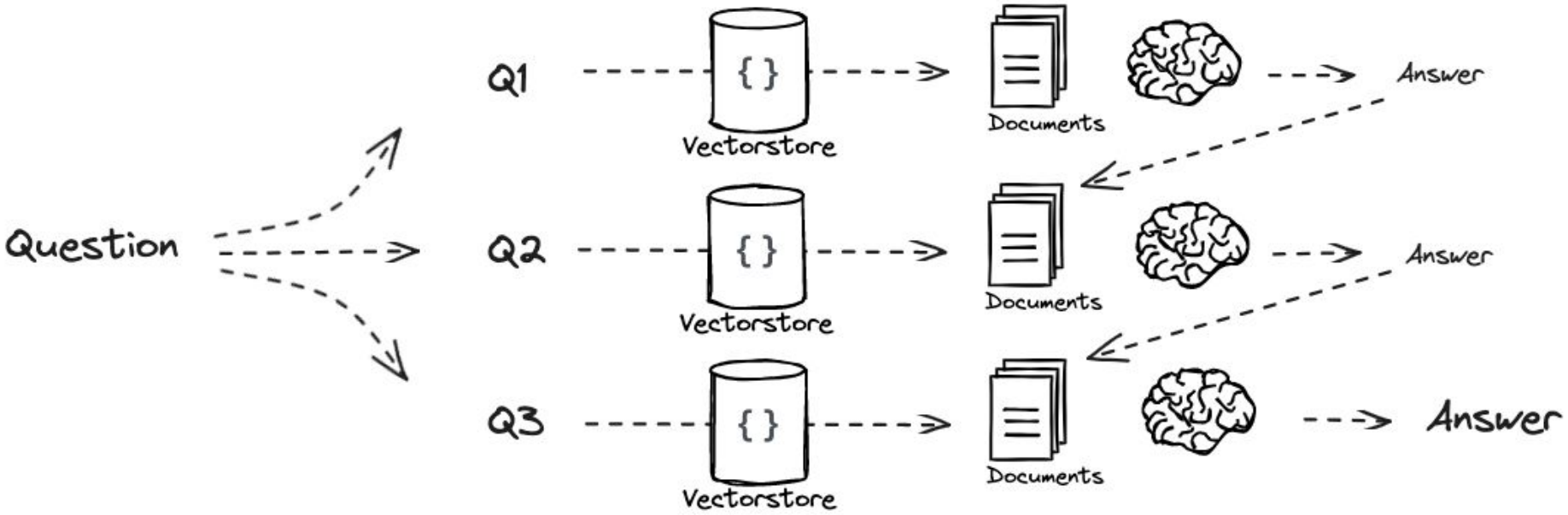
Table 12: Prompt of querying the model for final answer with additional contexts from original and step-back retrieval augmentations in TimeQA and SituatedQA

| dataset | Original Question | Step-back Question |
| --- | --- | --- |
| MuSiQue | at year saw the creation of the region where the county of Hertfordshire is located? | which region is the county of Hertfordshire located? |
| MuSiQue | Jan Šindel's was born in what country? | what is Jan Šindel's personal history? |
| MuSiQue | When was the abolishment of the studio that distributed The Game? | which studio distributed The Game? |
| MuSiQue | What city is the person who broadened the doctrine of philosophy of language from? | who broadened the doctrine of philosophy of language |

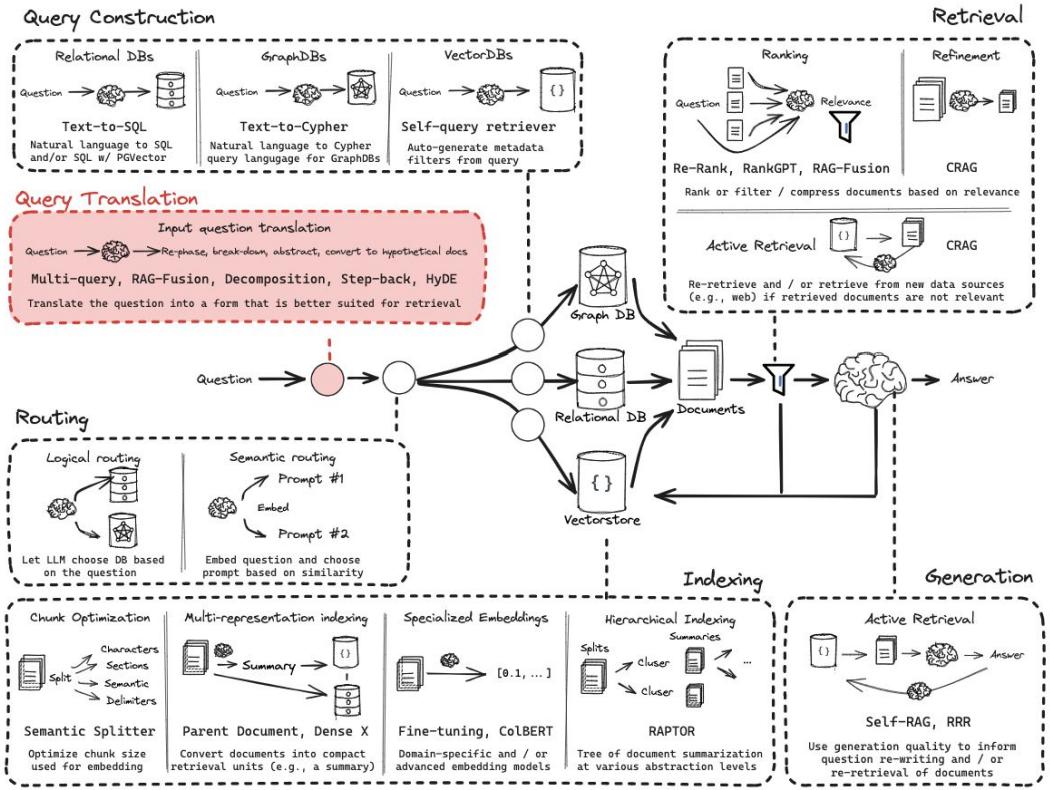


https://arxiv.org/pdf/2310.06117.pdf

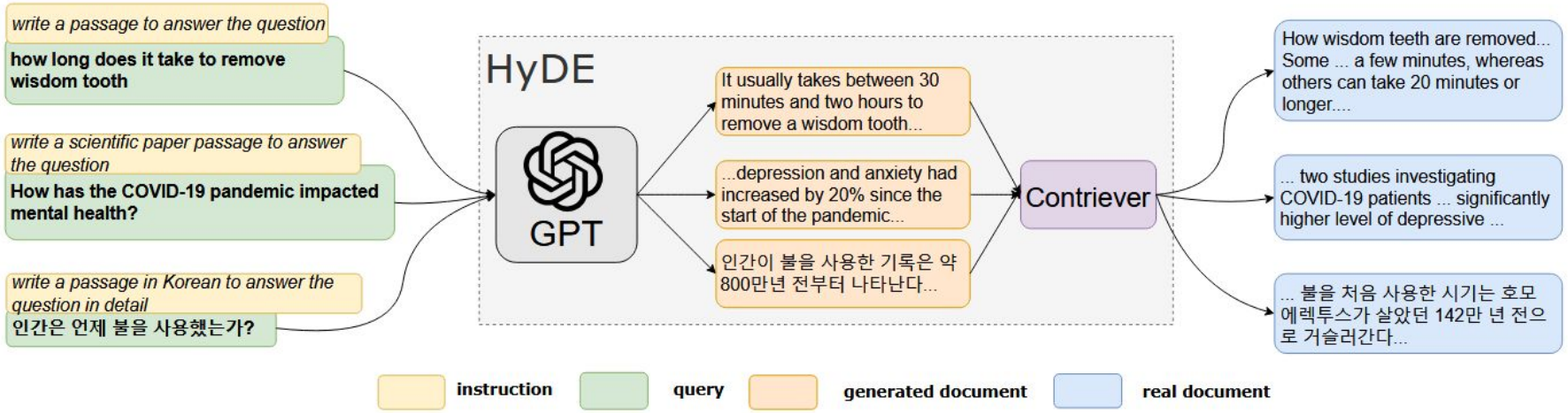
# Combine ideas: Dynamically retrieve to aid in solving the subproblems



# Query Translation

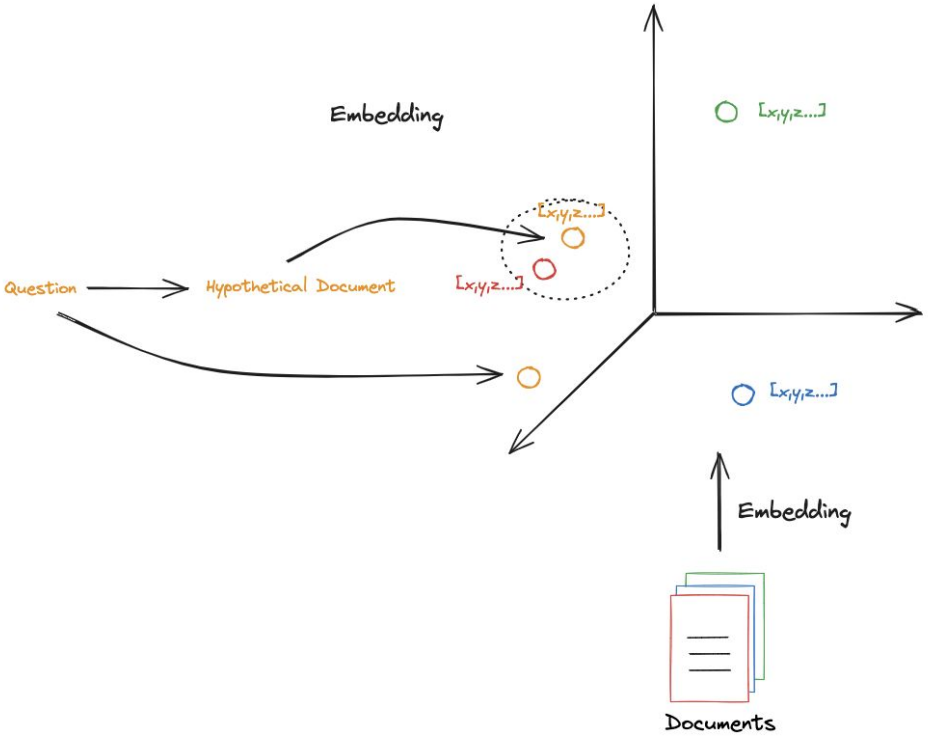


# Hypothetical Document Embeddings (HyDE)

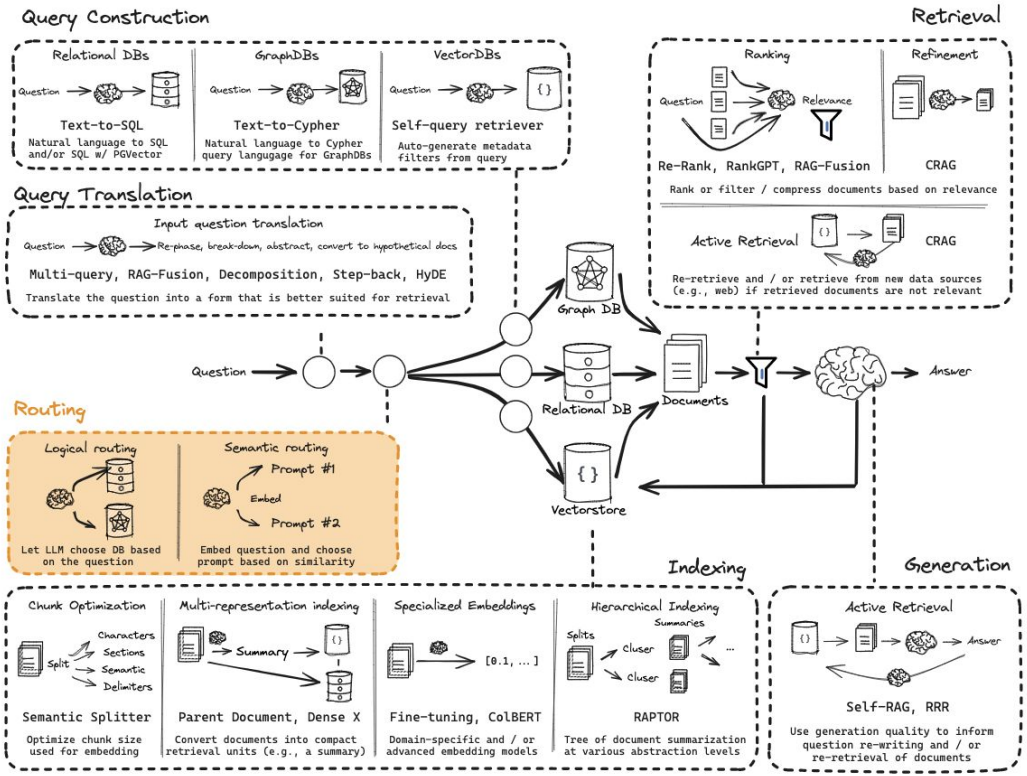



https://arxiv.org/pdf/2212.10496.pdf

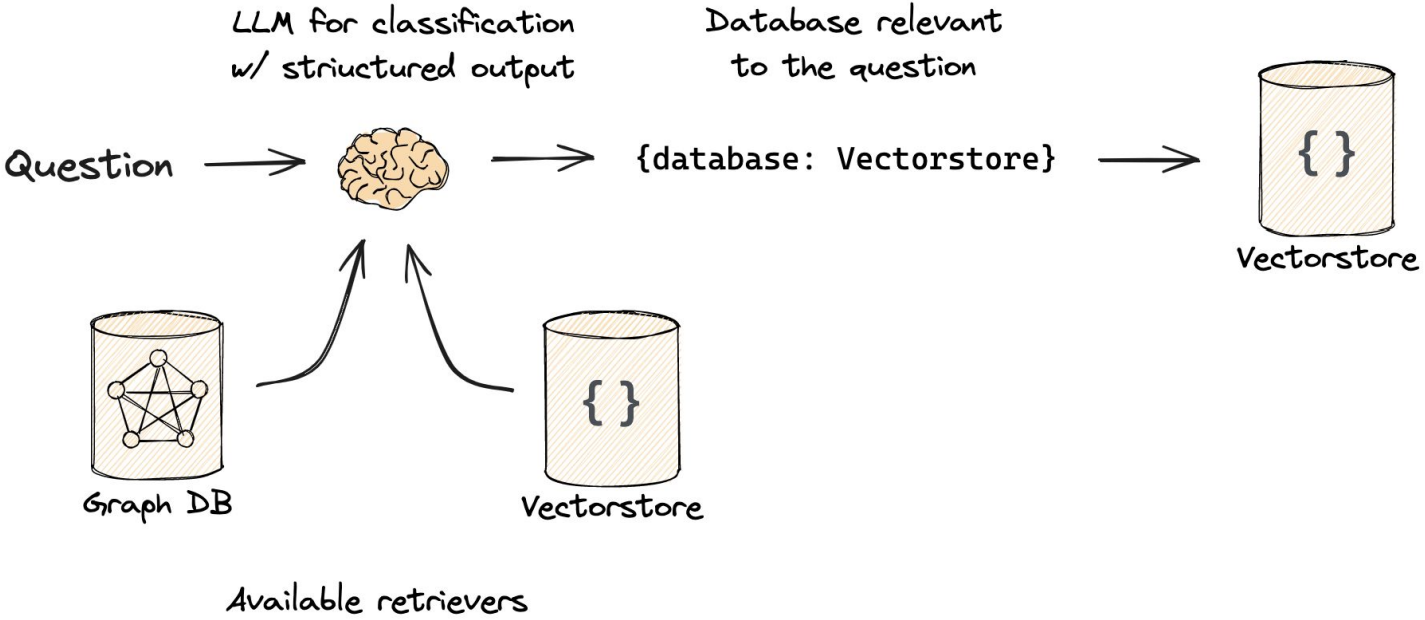
# HyDE Intuition



# Routing

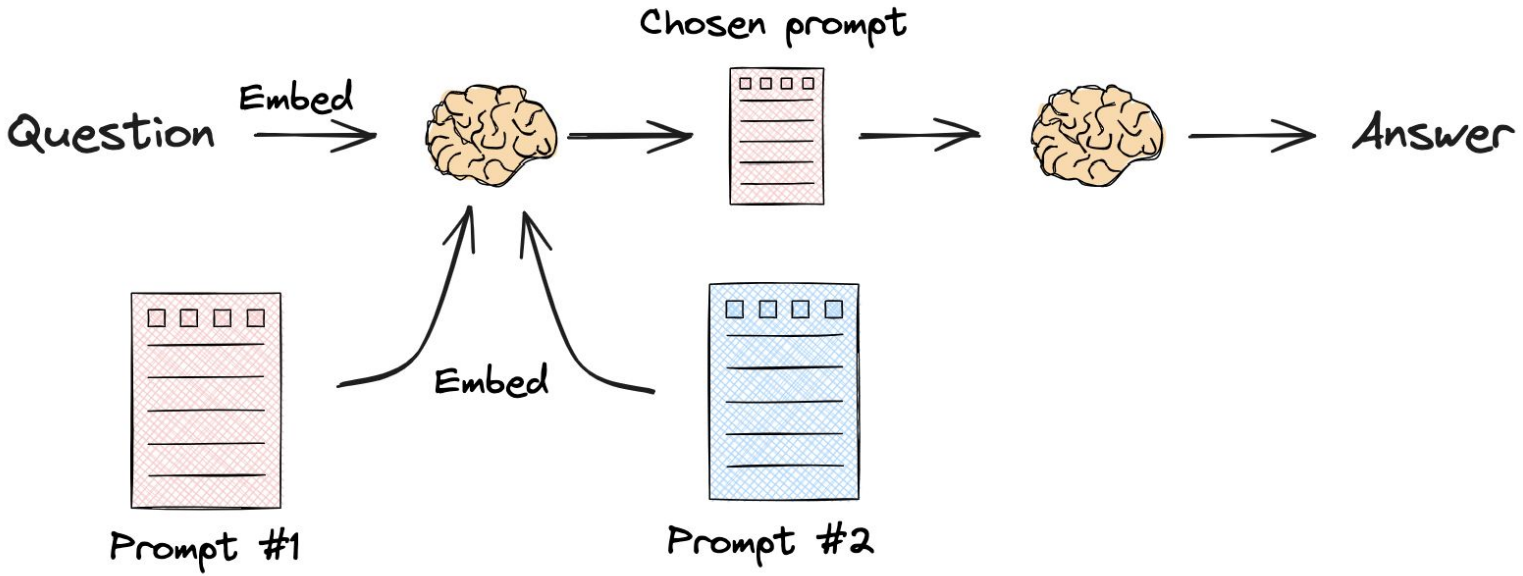

Query Construction
Relational DBs
Question
Text-to-SQL
Natural language to SQL and/or SQL w/ PGVector
GraphDBs
Question
Text-to-Cypher
Natural language to Cypher query language for GraphDBs
VectorDBs
Question
{ }
Self-query retriever
Auto-generate metadata filters from query
Query Translation
Input question translation
Question
Re-phase, break-down, abstract, convert to hypothetical docs
Multi-query, RAG-Fusion, Decomposition, Step-back, HyDE
Translate the question into a form that is better suited for retrieval
Retrieval
Ranking
Question
Relevance
Re-Rank, RankGPT, RAG-Fusion
Refinement
CRAG
Rank or filter / compress documents based on relevance
Active Retrieval
{ }
CRAG
Re-retrieve and / or retrieve from new data sources (e.g., web) if retrieved documents are not relevant
Graph DB
Question
Relational DB
Documents
Answer
Vectorstore
{ }
Routing
Logical routing
Let LLM choose DB based on the question
Semantic routing
Prompt #1
Embed
Prompt #2
Embed question and choose prompt based on similarity
Indexing
Chunk Optimization
Split
Characters
Sections
Semantic
Delimiters
Semantic Splitter
Optimize chunk size used for embedding
Multi-representation indexing
Summary
{ }
Parent Document, Dense X
Convert documents into compact retrieval units (e.g., a summary)
Specialized Embeddings
[0.1, ... ]
Fine-tuning, ColBERT
Domain-specific and / or advanced embedding models
Hierarchical Indexing
Summaries
Splits
cluser
cluser
...
RAPTOR
Tree of document summarization at various abstraction levels
Generation
Active Retrieval
{ }
Answer
Self-RAG, RRR
Use generation quality to inform question re-writing and / or re-retrieval of documents

# Logical routing




https://python.langchain.com/docs/use_cases/query_analysis/techniques/routing

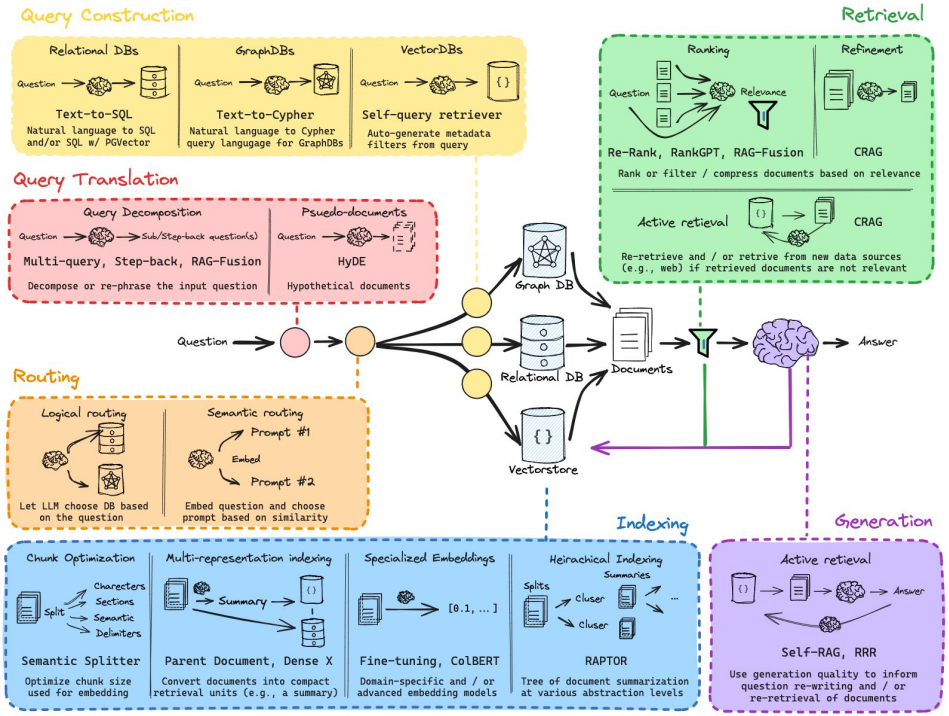
# Semantic routing




https://python.langchain.com/docs/expression_language/cookbook/embedding_router

# Fiqa example with Langchain (Dense+HNSW+Gemma3)

https://tinyurl.com/RAGETALTP3



# LLM only vs naïve RAG

**Question:Where should I park my rainy-day / emergency fund?**

w/o RAG — I don't know where to park your rainy-day/emergency fund.

w/ RAG — You should park your emergency fund somewhere that is ""safe"", meaning something that is not subject to market fluctuations. Your emergency fund is something you need to be able to count on when times are tough!

**Question:Tax considerations for selling a property below appraised value to family?**

w/o RAG — Tax considerations for selling a property below appraised value to family are complex and depend on factors such as the property's condition, the family's income, and the applicable tax laws.

w/ RAG — Yes, this is legal. The state of Maryland has a transfer/recordation tax of 1.5% for each, the buyer and seller. Would this be computed on the appraised or sale value?

# LLM only vs naïve RAG

**Question:Can the Delta be used to calculate the option premium given a certain target?**

w/o RAG — The Delta can be used to calculate the option premium given a certain target.

w/ RAG — The Delta is the sensitivity to the underlying stock price, and the theta is the sensitivity to time passing.

**Question:Basic Algorithmic Trading Strategy**

w/o RAG — The basic algorithmic trading strategy involves understanding the market dynamics and employing various techniques to profit from price fluctuations.

w/ RAG — Algorithmic trading involves using algorithms to analyze market data and make trading decisions. The key to success is understanding the underlying principles and developing a strategy that is both efficient and adaptable to changing market conditions.

# LLM only vs naïve RAG

**Question:What does a high operating margin but a small but positive ROE imply about a company?**

w/o RAG — A high operating margin but a small but positive ROE suggests the company is efficient in generating profits, which can be a positive factor for investors.

w/ RAG — A high operating margin but a small but positive ROE implies that the company's stock is undervalued relative to its earnings, potentially due to its stock being bid up in relation to its earnings and may not be a good thing.

La phrase de la fin… et MERCI POUR VOTRE ATTENTION !!!

Il faut tourner son **moteur de recherche sept fois** dans une boucle **avant de générer une réponse**