Semantic Search:
New Solutions to Old Problems

Philipp Krenn, @xeraa

Agenda

- "Classic" search and its limitations
- How to use semantic search
- What is a model and how can you use it
- Where OpenAI's ChatGPT comes into play

Elasticsearch:
You Know, for Search


elasticsearch
Lucene

elastic

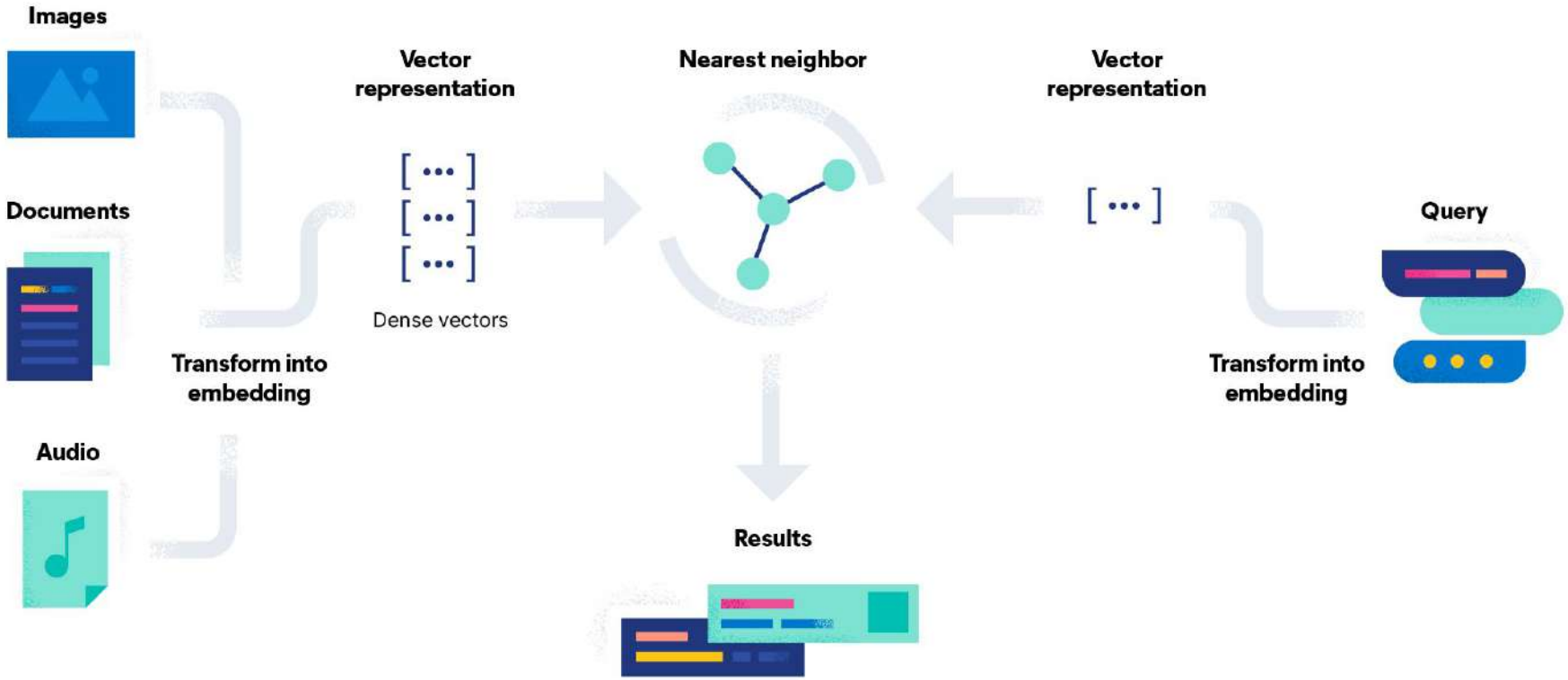
```
GET /_analyze
{
  "char_filter": [ "html_strip" ],
  "tokenizer": "standard",
  "filter": [ "lowercase", "stop", "snowball" ],
  "text": "These are <em>not</em> the droids
           you are looking for."
}
```

```
{ "tokens": [{
      "token": "droid",
      "start_offset": 27, "end_offset": 33,
      "type": "<ALPHANUM>",
      "position": 4
    },{
      "token": "you",
      "start_offset": 34, "end_offset": 37,
      "type": "<ALPHANUM>",
      "position": 5
    }, {
      "token": "look",
      "start_offset": 42, "end_offset": 49,
      "type": "<ALPHANUM>",
      "position": 7
    }
```

Semantic Search: Meaning, not literal matches

Elasticsearch:
You Know, for Vector Search

Vector Search



What is a Vector?

elastic

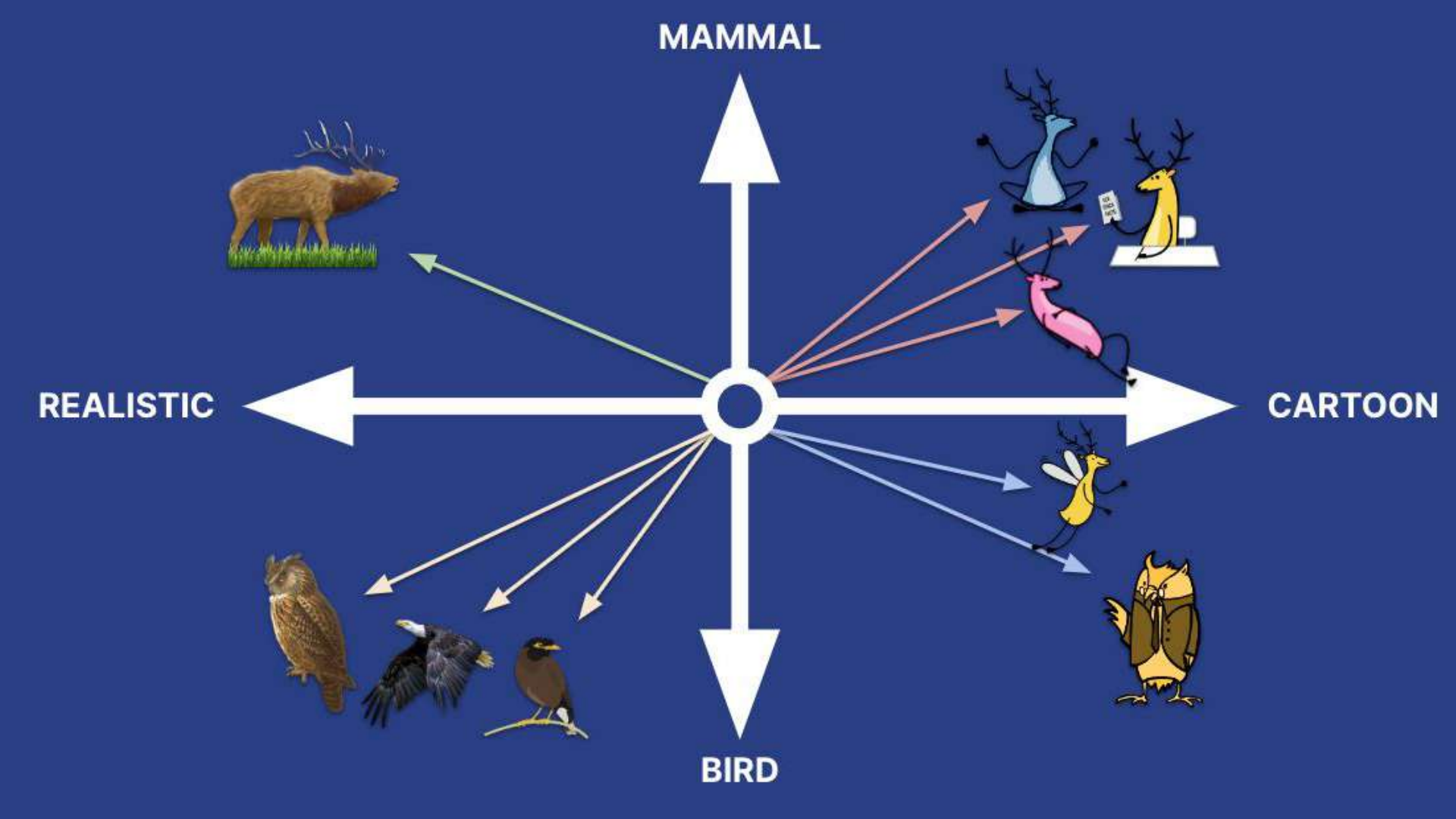
Bag of Words

As a simplified illustration

These are not the droids you are looking for.
No. I am your father.

these: 1, are: 2, not: 1, the: 1, droid: 1, you: 1, look: 1, for: 1
no: 1, i: 1, am: 1, you: 1, father: 1

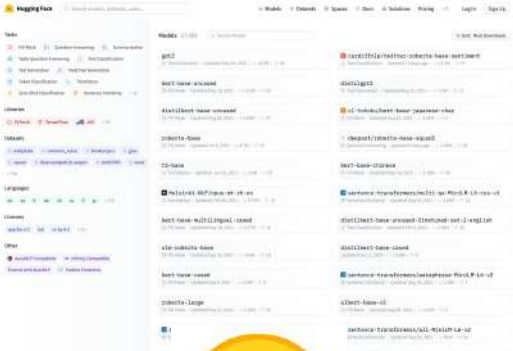
```
[these, are, not, the, droid, you, look, for, no, i, am, father]
[1,     2,   1,   1,   1,     1,   1,    1,   0,  0, 0,  0]
[0,     0,   0,   0,   0,     1,   0,    0,   1,  1, 1,  1]
```


MAMMAL
REALISTIC
CARTOON
BIRD

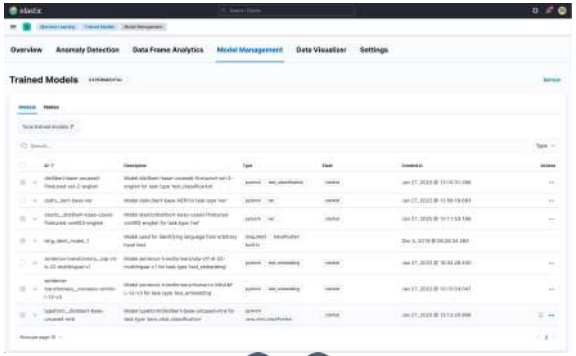
What is a Model?

elastic

Eland Imports PyTorch Models



```
$ eland_import_hub_model
--url https://Cluster_URL
--hub-model-id bert_model
--task-type text_embedding
--start
```



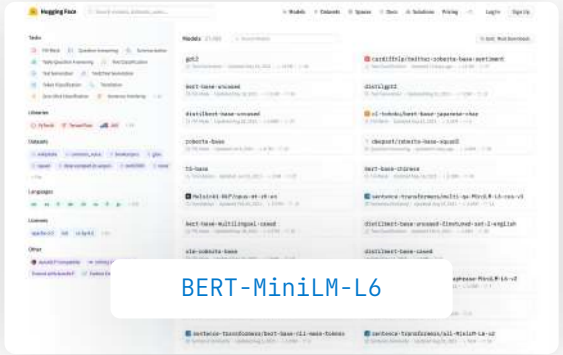
Huggingface.co

eland

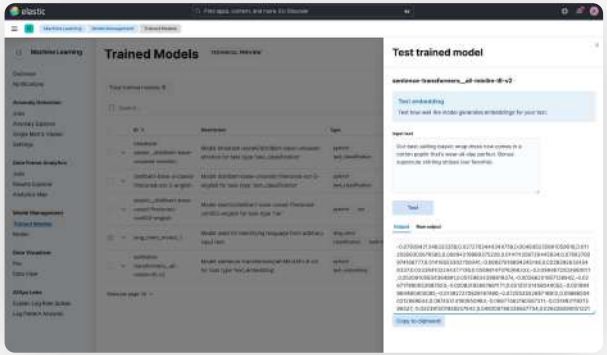
Inference, not training

elastic

Setting Up the Embedding Model




```
$ eland_import_hub_model
--url https://cluster_URL
--hub-model-id BERT-MiniLM-L6
--task-type text_embedding
--start
```



PyTorch → eland →

Select the appropriate model

Load the model to the cluster

Manage models

elastic

Hugging Face NLP Libraries

Hugging Face Model	`task-type`
Name-Entity recognition	`ner`
Text embedding	`text_embedding`
Text classification	`text_classification`
Zero shot classification	`zero_shot_classification`
Question & Answer	`question_answering`

elastic

Choice of Embedding Model

Start with Off-the Shelf Models

- Text data: Hugging Face
- Images: OpenAI's CLIP

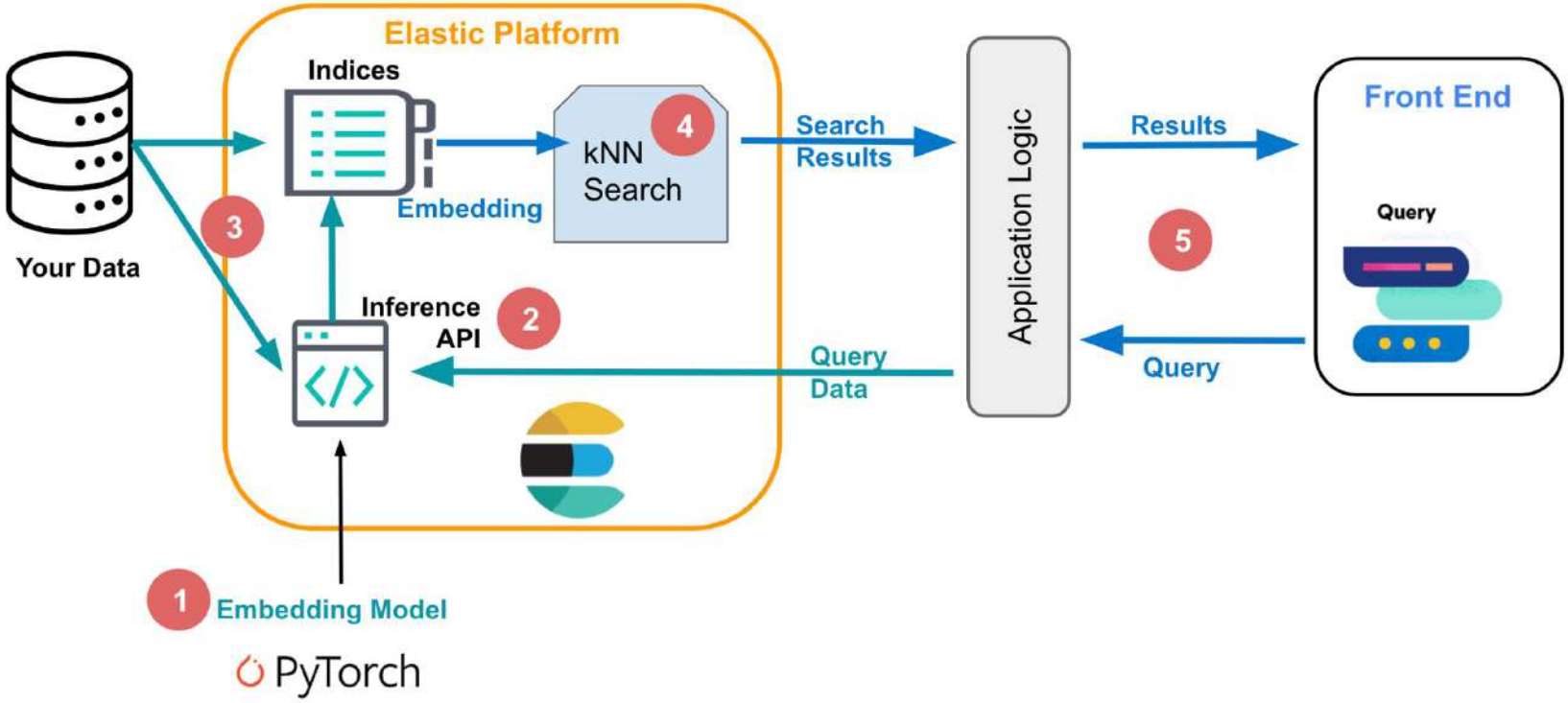
Extend to Higher Relevance

- Apply hybrid scoring
- Bring Your Own Model:
  requires expertise + labeled data

How Do You Search Vectors?

elastic

Architecture of Vector Search



elastic

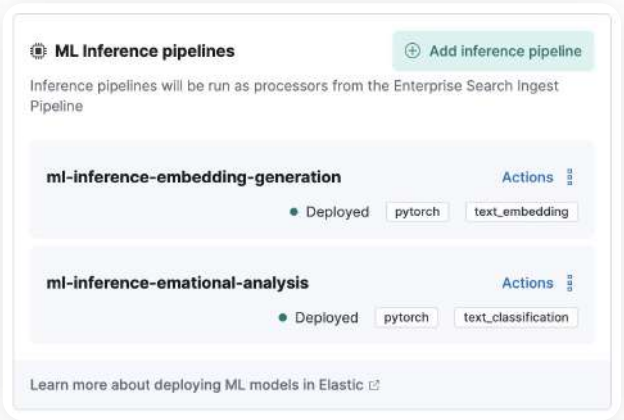
Data Ingestion and Embedding Generation

Source data

POST /_doc

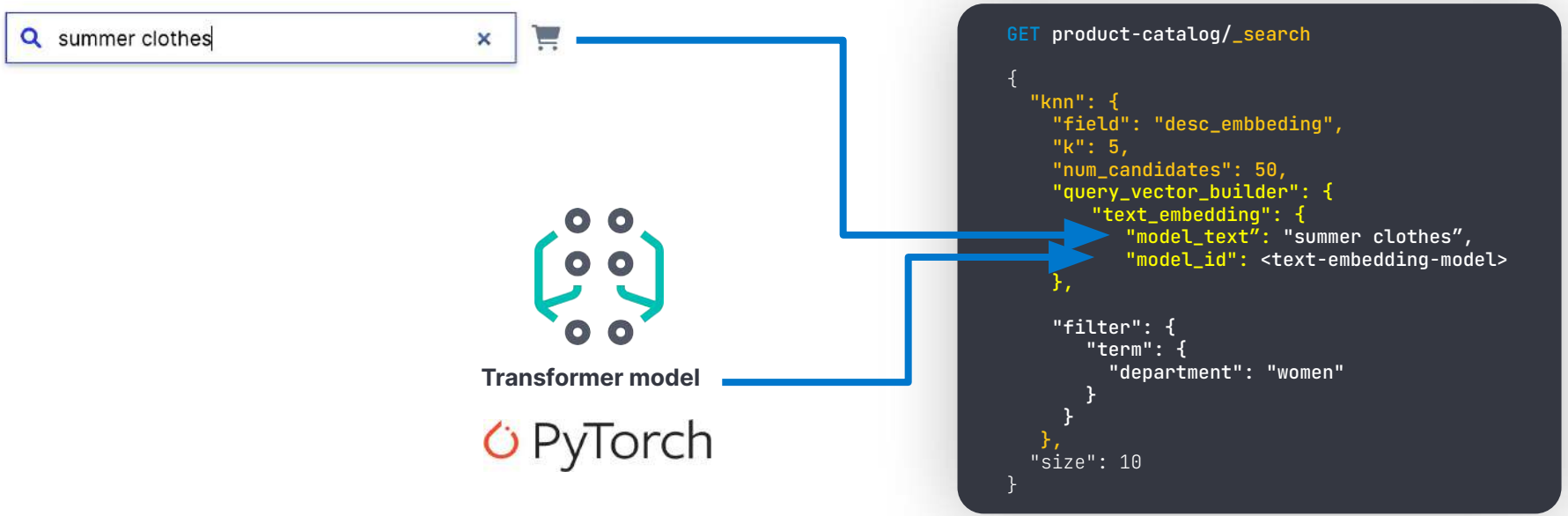
```
{
  "_id":"product-1234",
  "product_name":"Summer Dress",
  "description":"Our best-selling…",
  "Price": 118,
  "color":"blue",
  "fabric":"cotton",
  "desc_embedding":[0.452,0.3242,…]
}
```

POST /_doc



elastic

Vector Query



elastic

But How Does It Really Work?

elastic

Hierarchical Navigable Small Worlds (HNSW)

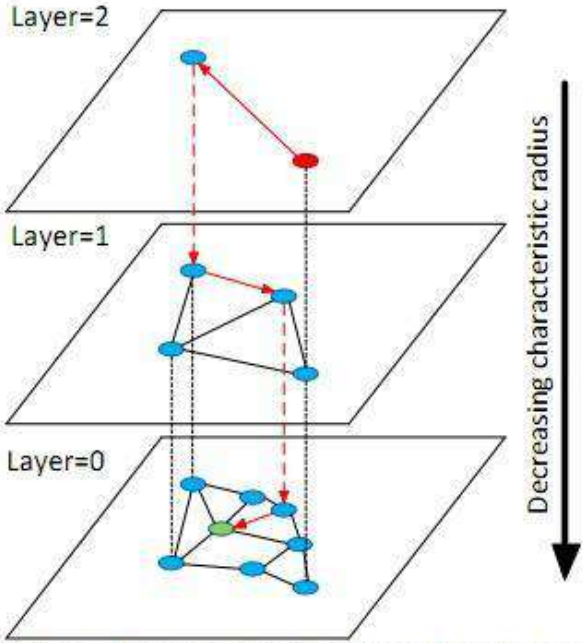


Fig. 1. Illustration of the Hierarchical NSW idea. The search starts from an element from the top layer (shown red). Red arrows show direction of the greedy algorithm from the entry point to the query (shown green).

https://zhuanlan.zhihu.com/p/98028479

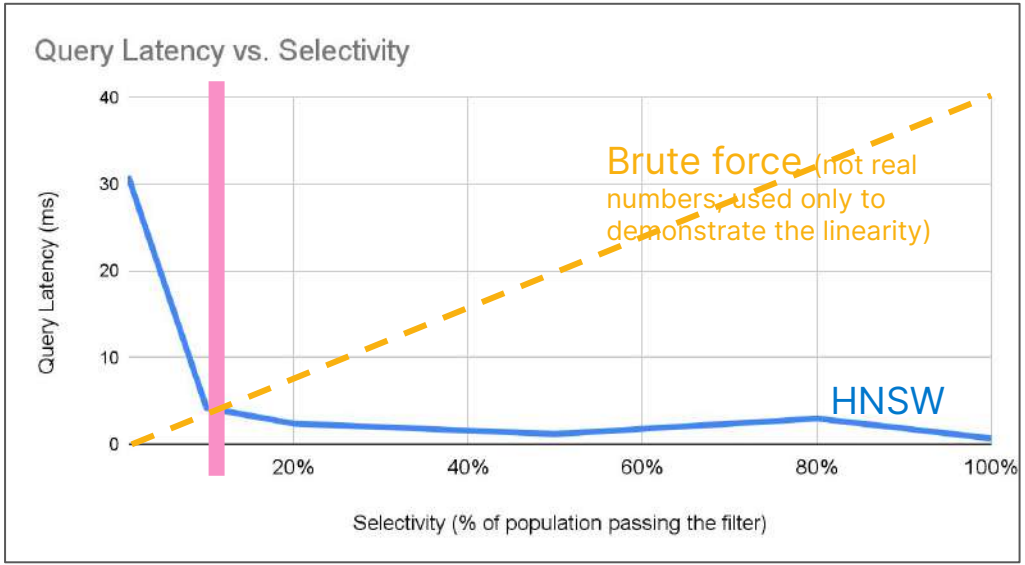
elastic

Filtering KNN Vector Similarity

Automatically choose between brute force and HNSW

Bound worst case to 2*(brute force)

- Brute force scales O(n) of filtered
- HNSW scales ~O(log(n)) of all docs



elastic

Elasticsearch + Lucene: Fast Progress

Increase max number of vector dims to 2048 #95257

Merged **mayya-sharipova** merged 3 commits into `elastic:main` from `mayya-sharipova:vdims_2048` 2 weeks ago

Conversation 9 | Commits 3 | Checks 0 | Files changed 12

mayya-sharipova commented 3 weeks ago — Member

Currently Lucene limits the max number of vector dimensions to 1024.
This commit overrides KnnFloatVectorField and KnnByteVectorField
classes to increase the limit to 2048 for indexed vectors in ES.

13

Increase max number of vector dims to 2048 ··· 9746994

Scaling Vector Search

Vector search:

1. Needs lots of memory
2. Indexing is slower
3. Merging is slow

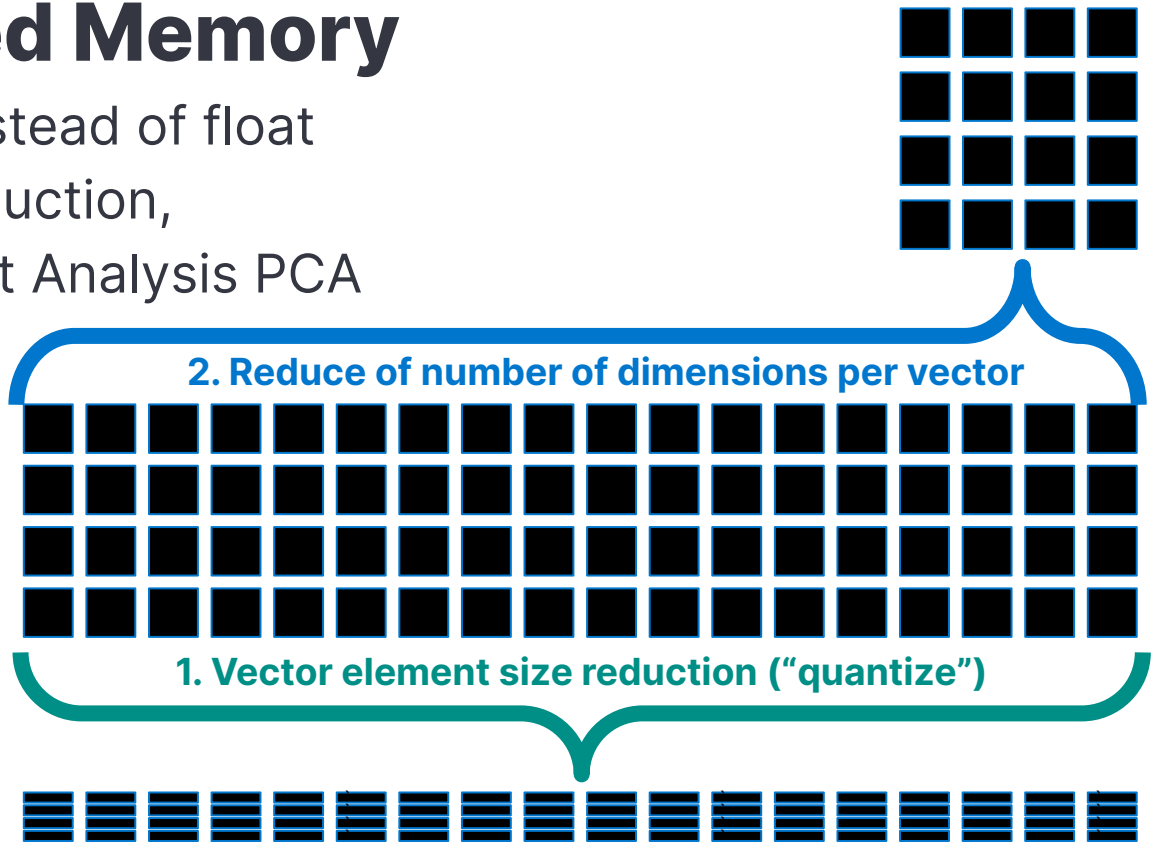
* Continuous improvements in Lucene + Elasticsearch

Best practices:

1. Avoid searches during indexing
2. Exclude vectors from _source
3. Reduce vector dimensionality
4. Use byte rather than float

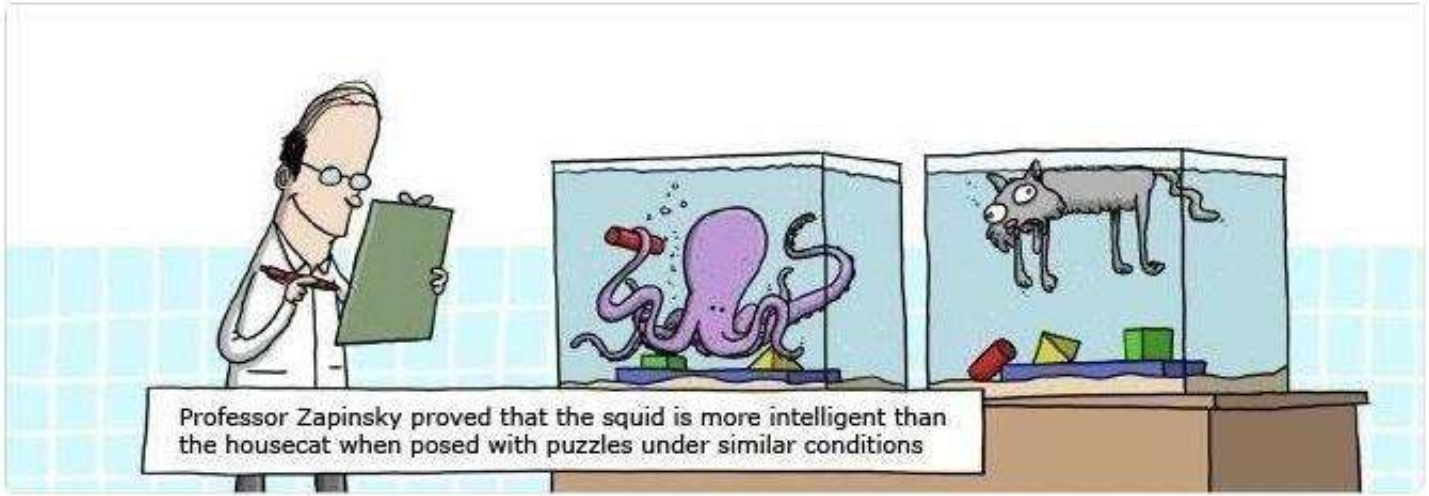
Reduce Required Memory

1. Store vectors as byte instead of float
2. Apply dimensionality reduction,
   e.g. Principal Component Analysis PCA



elastic

Benchmarketing



https://github.com/erikbern/ann-benchmarks

Add Elasticsearch KNN #401

Merged erikbern merged 1 commit into `erikbern:main` from `ceh-forks:elasticsearch` last week

Conversation 5 | Commits 1 | Checks 34 | Files changed 4

ceh commented last week • edited — Contributor

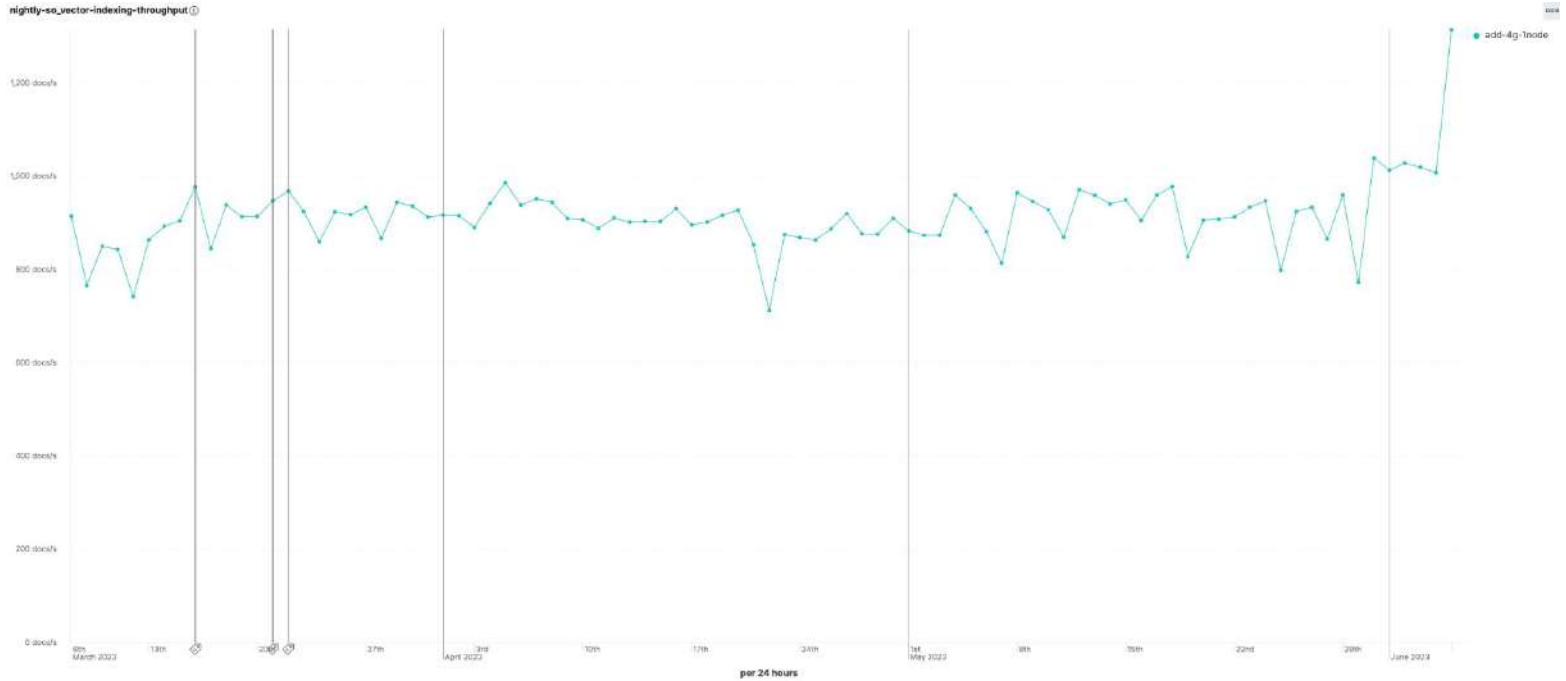
Add an implementation for Elasticsearch KNN search. Fixes #298.

- Supports specifying `index_options` via arg groups, with one entry for Elasticsearch's default settings for M and EF.
- Supports specifying `num_candidates` via query args.

What do you think **@erikbern**, **@alexklibisz**?

1

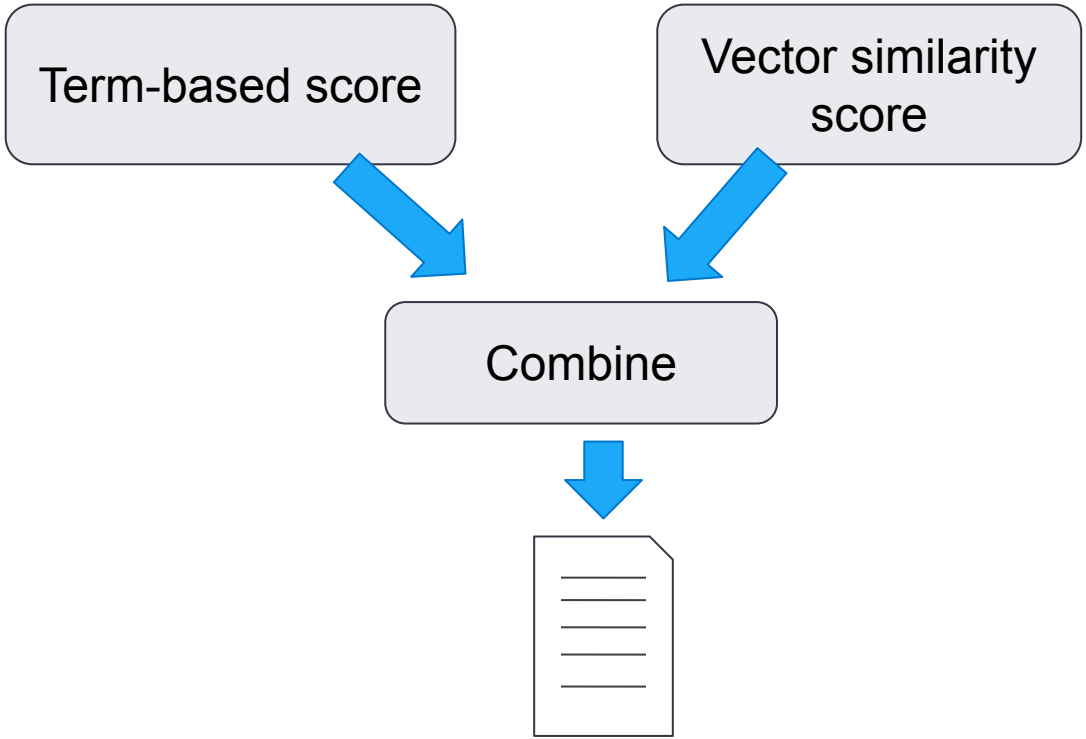
https://elasticsearch-benchmarks.elastic.co/index.html#tracks/so_vector/nightly/default/30d



elastic

Elasticsearch: You Know, for Hybrid Search

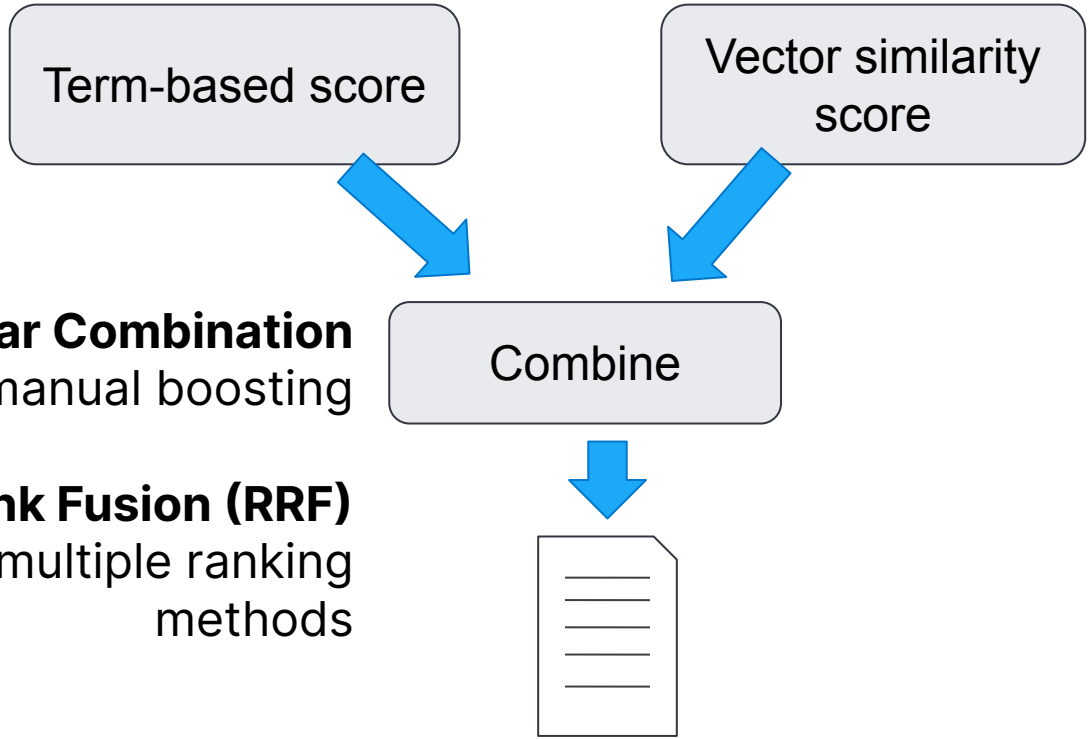
Hybrid Scoring



```
GET product-catalog/_search
{
  "query": {
    "match": {
      "description": {
        "query": "summer clothes",
        "boost": 0.9
      }
    }
  },
  "knn": {
    "field": "desc_embbeding",
    "query_vector": [0.123, 0.244,...],
    "k": 5,
    "num_candidates": 50,
    "boost": 0.1,
    "filter": {
       "term": {
         "department": "women"
       }
     }
   },
  "size": 10
}
```

```
GET product-catalog/_search
{
  "query": {
    "match": {
      "description": {
        "query": "summer clothes",
        "boost": 0.9
      }
    }
  },
  "knn": [ {
    "field": "image-vector",
    "query_vector": [54, 10, -2],
    "k": 5,
    "num_candidates": 50,
    "boost": 0.1
  },
  {
    "field": "title-vector",
    "query_vector": [1, 20, -52, 23, 10],
    "k": 10,
    "num_candidates": 10,
    "boost": 0.5
  }],
  "size": 10
}
```

Combine



Linear Combination
manual boosting

Reciprocal Rank Fusion (RRF)
blend multiple ranking methods

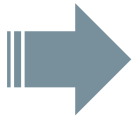
Reciprocal Rank Fusion (RRF)

$$RRFscore(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)}$$

D - set of docs
R - set of rankings as permutation on 1..|D|
K - typically set to 60 by default

Ranking Algorithm 1		
Doc	Score	r(d)
A	1	1
B	0.7	2
C	0.5	3
D	0.2	4
E	0.01	5

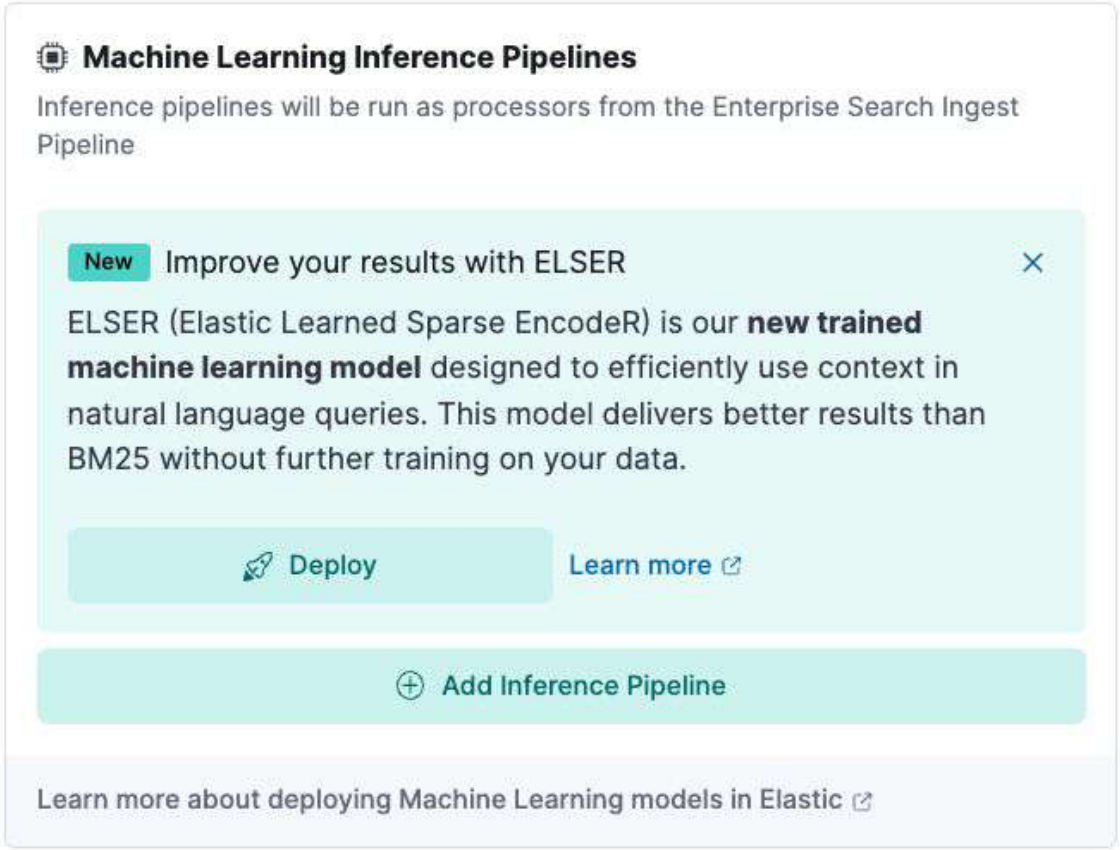
Ranking Algorithm 2		
Doc	Score	r(d)
C	1,341	1
A	739	2
F	732	3
G	192	4
H	183	5



Doc
A
C
B
F
D

elastic

ELSER: Elastic Learned Sparse EncodER



elastic

```
POST /_ingest/pipeline/elser-v1-demo/_simulate
{
  "docs": [
    {
      "_index": "my_index",
      "_id": "id",
      "_source": {
        "text_field": "These are not the droids you are looking for."
      }
    }
  ]
}
```

elastic

```
"text_field": "These are not the droids you are looking for.",
          "ml": {
            "tokens": {
              "lucas": 0.50047517,
              "ship": 0.29860738,
              "dragon": 0.5300422,
              "quest": 0.5974301,
              "dr": 2.1055143,
              "space": 0.49377063,
              "robot": 0.40398192,
              "these": 0.19085139,
              "lei": 0.23646113,
              …
```

Free vs Paid (Platinum)

kNN with HNSW

1. Bring your own embeddings
2. Enough memory for vectors to fit in off-heap RAM

Inference in Elasticsearch

1. Generate embeddings (like HuggingFace transformers)
2. ELSER

ChatGPT:
Elastic and LLM

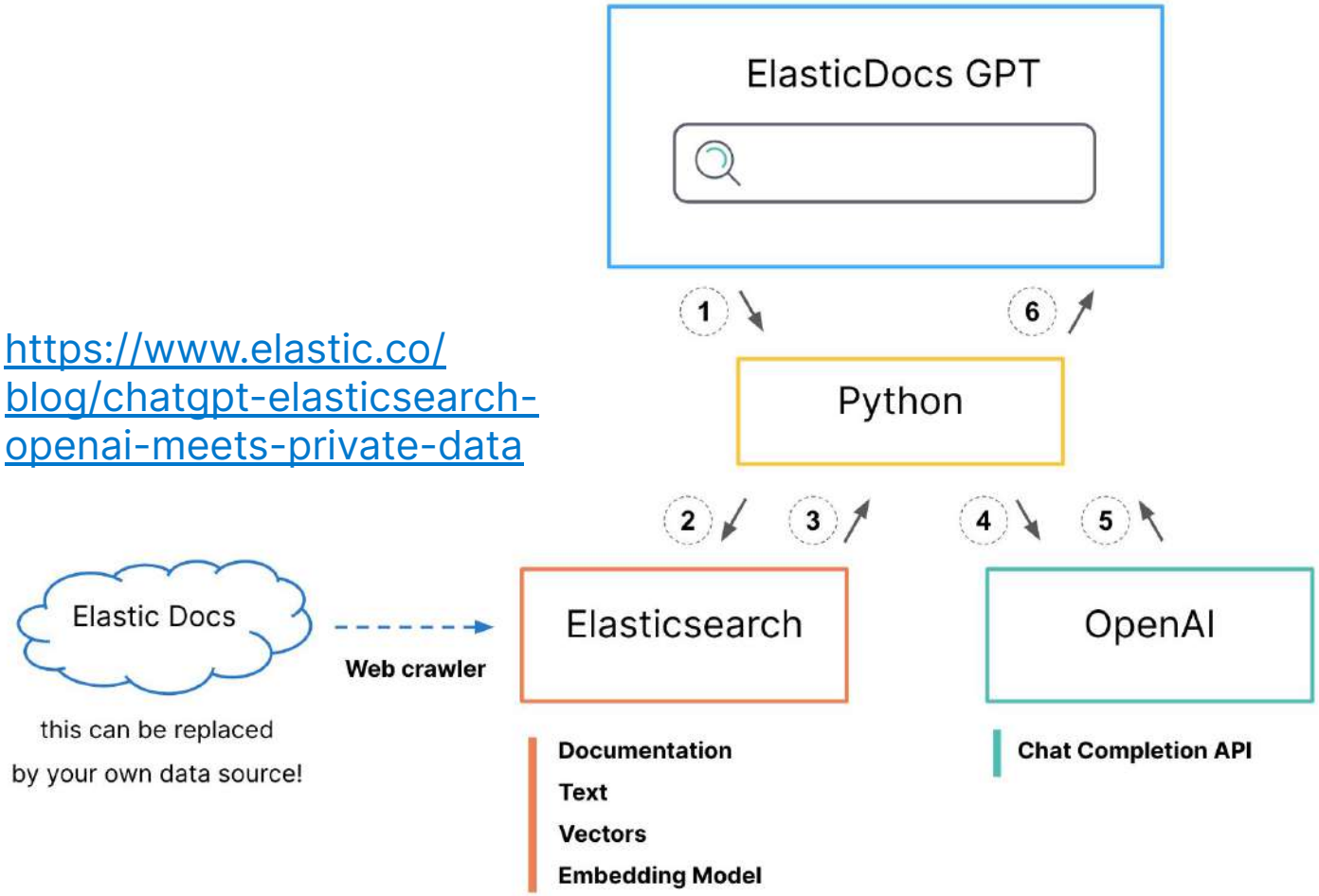
What's (Not) Great about ChatGPT?

What's (Not) Great about ChatGPT?

An error occurred. Either the engine you requested does not exist or there was another issue processing your request. If this issue persists please contact us through our help center at help.openai.com.

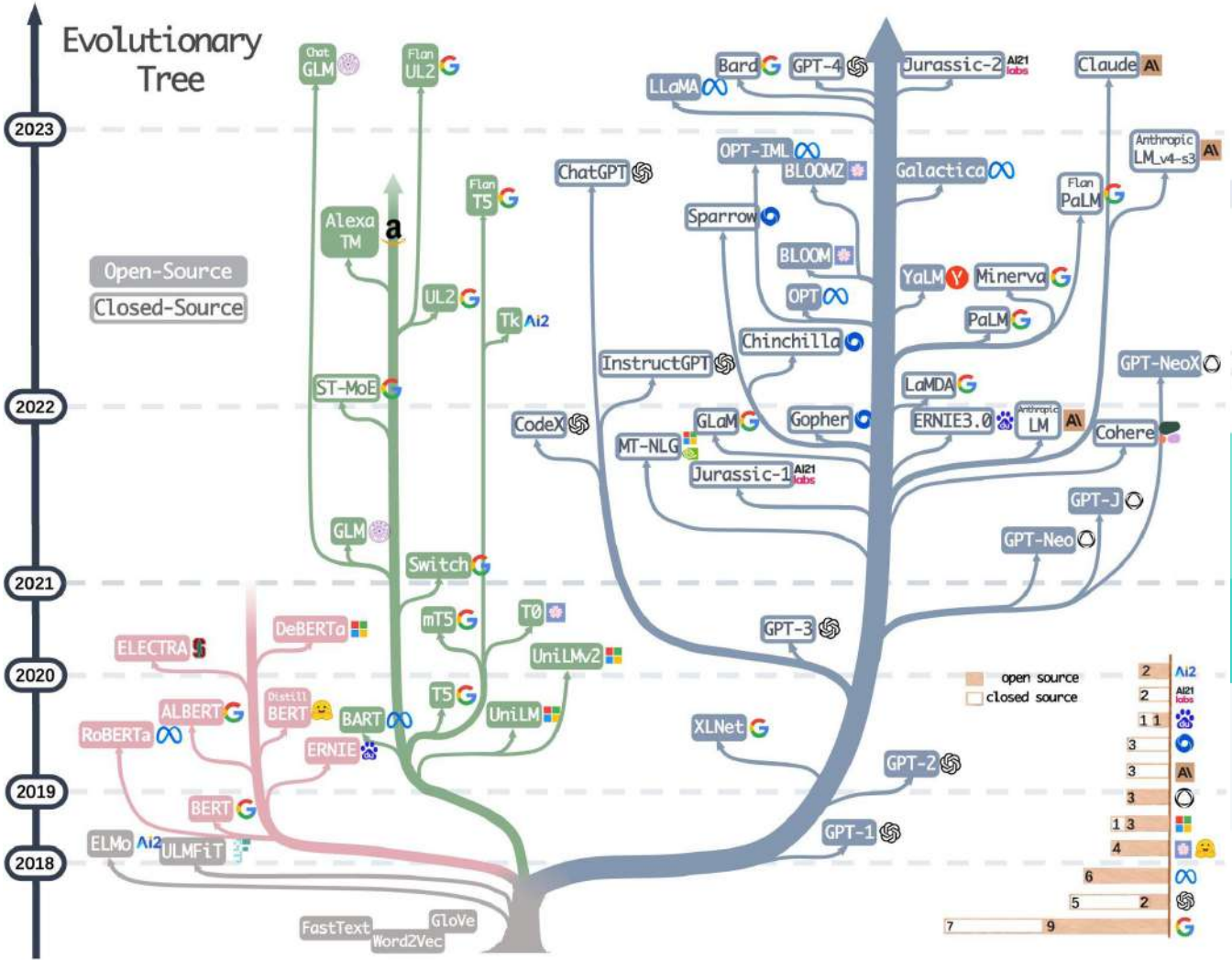
elastic

https://www.elastic.co/blog/chatgpt-elasticsearch-openai-meets-private-data



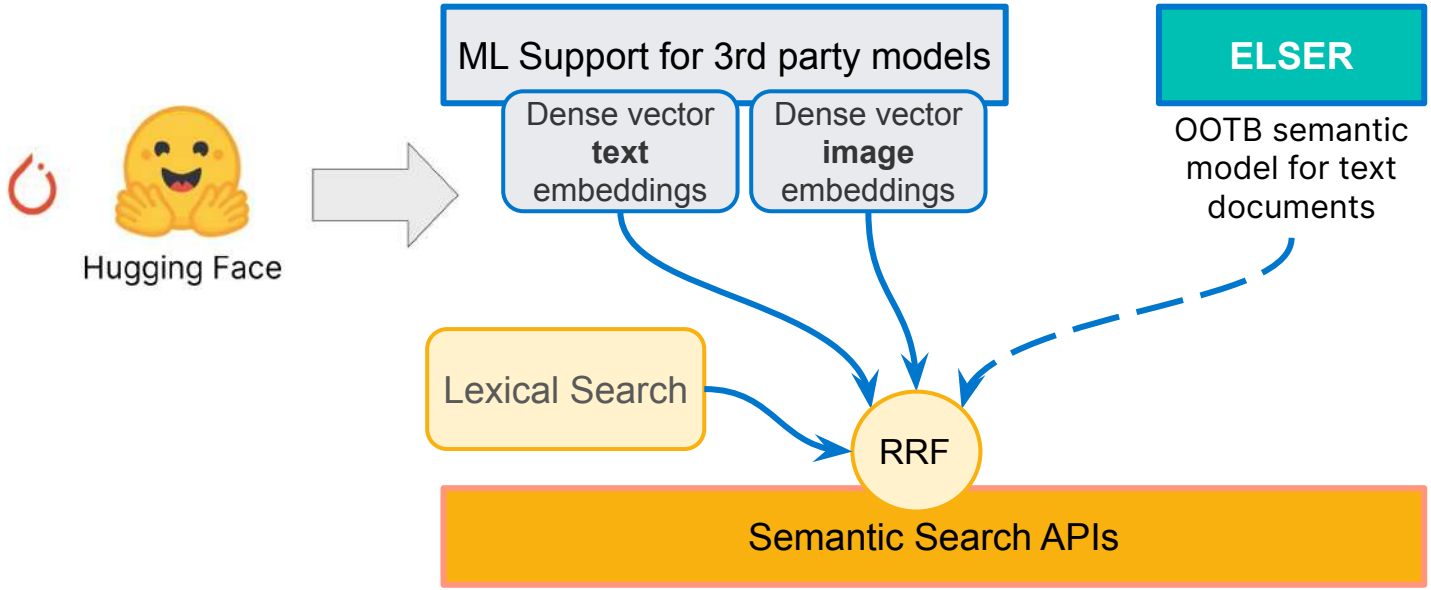
elastic

https://github.com/Mooler0410/LLMsPracticalGuide

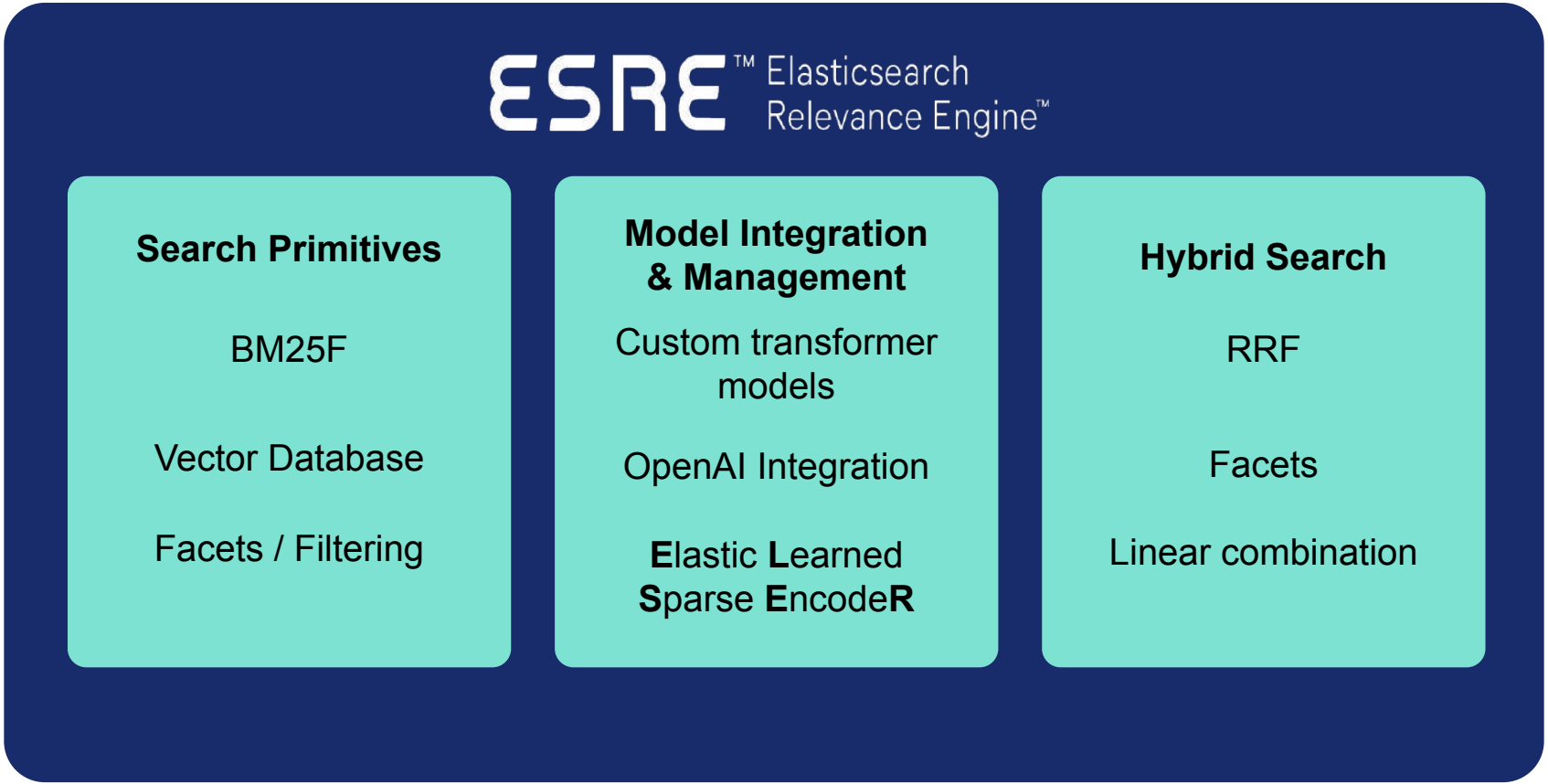


Conclusion

Search: from Literals to Semantics



elastic


ESRE™ Elasticsearch Relevance Engine™
Search Primitives
BM25F
Vector Database
Facets / Filtering
Model Integration & Management
Custom transformer models
OpenAI Integration
Elastic Learned Sparse EncodeR
Hybrid Search
RRF
Facets
Linear combination

Elasticsearch:
You Know, for Search