

Search a new era











Elasticsearch

You Know, for Search





APACHE TLUCENE









These are not the droids you are looking for.



GET / analyze { "char filter": ["html strip"], "tokenizer": "standard", "filter": ["lowercase", "stop", "snowball"], "text": "These are not the droids you are looking for."



"char_filter": "html_strip"

These are not the droids you are looking for.



These are not the droids you are looking for.



"tokenizer": "standard"

These are not the droids you are looking for.

These are not the droids are looking for



"filter": "lowercase"

These	these
are	are
not	not
the	the
droids	droids
you	you
are	are
looking	looking
for	for



"filter": "stop"

These	these	
are	are	
not	not	
the	the	
droids 🔶	droids	droids
you	you	you
are	are	
looking	looking	looking
for	for	



"filter": "snowball"





```
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 ≠ Literal matches S similar

YOU'RE COMPARING APPLES TO NECTARINES

Elasticsearch

You Know, for Search







What is a **Vector**?





Embeddings represent your data Example: 1-dimensional vector





Multiple dimensions represent different data aspects



Similar data is grouped together





Vector search ranks objects by similarity (~relevance) to the query





How do you index vectors?



Architecture of Vector Search



Choice of Embedding Model

Start with Off-the Shelf Models

Extend to Higher Relevance

Text data: Hugging Face (like Microsoft's E5)

Images: OpenAl's CLIP

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



Problem training vs actual use-case





dense_vector field type

```
PUT ecommerce
{
    "mappings": {
        "properties": {
            "description": {
               "type": "text"
            }
        "desc_embedding": {
               "type": "dense_vector"
            }
        }
      }
    }
}
```



Data Ingestion and Embedding Generation





With Elastic ML







How do you search vectors?



Architecture of Vector Search











semantic_text field type

```
PUT ecommerce
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "copy_to": [ "desc_embedding" ]
      "desc embedding": {
        "type": "semantic text"
                 GET ecommerce/ search
                   "query": {
                     "semantic": {
                       "field": "desc embedding"
                       "query" : "I'm looking for a red dress for a DJ party"
```



POST ecommerce/ doc "description": "Our best-selling..."

```
elastic
```

Architecture of Vector Search





But how does it

really work?



Similarity





Similarity: cosine (cosine)





Similarity: Dot Product (dot_product or max_inner_product) θ $|\tilde{q}| \times cos(\theta)$

 $\vec{q} \times \vec{d} = |\vec{q}| \times cos(\theta) \times |\vec{d}|$

$$_score_{float} = \frac{1 + dot_product(q, d)}{2}$$
$$_score_{byte} = \frac{0.5 + dot_product(q, d)}{32768 \times dims}$$



Similarity: Euclidean distance (12_norm)



$$l2_norm_{q,d} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

_score = $\frac{1}{1 + (l2_norm_{q,d})^2}$



Brute Force







Hierarchical Navigable Small Worlds (HNSW) One popular approach



HNSW: a layered approach that simplifies access to the nearest neighbor



Tiered: from coarse to fine approximation over a few steps



Balance: Bartering a little accuracy for a lot of scalability



Speed: Excellent query latency on large scale indices









Scalar Quantization









float32

Recall: High Precision: High Rescore: Likely Not Needed

Full RAM Required

int8

Recall: Good Precision: Good Oversampling: Moderate

Rescore: Reasonable

4X RAM Savings

int4

Recall: Low Precision: Low Oversampling: Needed Rescore: may be slower 8X RAM Savings

bit

Recall: Bad Precision: Bad Oversampling: Needed

Rescore: Expensive and Limiting

32X RAM Savings



BBQ aka Better Binary Quantization



BBQ: 32X RAM savings. Faster & more accurate than Product Quantization



Memory required



Memory required in MB for 500k 1024 vectors



100M vectors? Only 12GB!?! One single node.



Benchmarketing



https://djdadoo.pilato.fr/

https://github.com/dadoonet/music-search/

Hybrid scoring

Manual boosting

```
GET ecommerce/ search
  "query" : {
    "bool" : {
      "must" : [{
        "match": {
          "description": {
            "query": "summer clothes"
        "semantic": {
          "field": "desc embbeding",
          "query": "summer clothes",
          "boost": 100.0
```



```
PUT starwars
  "mappings": {
    "properties": {
      "text.tokens": {
        "type": "sparse vector"
          "These are not the droids you are looking for.",
          "Obi-Wan never told you what happened to your father."
             GET starwars/ search
                "query":{
                   "sparse vector": {
                     "field": "text.tokens",
                     "query vector": { "lucas": 0.50047517,
                                       "ship": 0.29860738,
                                       "dragon": 0.5300422,
                                       "quest": 0.5974301, ... }
```


ELSER Elastic Learned Sparse EncodER

sparse_vector Not BM25 or (dense) vector Sparse vector like BM25 Stored as inverted index

Machine Learning Inference Pipelines

Inference pipelines will be run as processors from the Enterprise Search Ingest Pipeline

New Improve your results with ELSER

ELSER (Elastic Learned Sparse EncodeR) is our **new trained machine learning model** designed to efficiently use context in natural language queries. This model delivers better results than BM25 without further training on your data.

🖉 Deploy

Learn more 🕑

Add Inference Pipeline

Learn more about deploying Machine Learning models in Elastic 🕑

Commerciar

×

Hybrid ranking

Dense Vector				BM25			
Doc	Score	r(d)	k+r(d)	Doc	Score	r(d)	k+r(d)
Α	1	1	61	С	1,341	1	61
В	0.7	2	62	Α	739	2	62
С	0.5	3	63	F	732	3	63
D	0.2	4	64	G	192	4	64
E	0.01	5	65	н	183	5	65

Reciprocal Rank Fusion (RRF)

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

	Doc	RRF Score
	Α	1/61 + 1/62 = 0,0325
	С	1/63 + 1/61 = 0,0323
	В	1/62 = 0,0161
	F	1/63 = 0,0159
	D	1/64 = 0,0156


```
GET index/ search
  "retriever": {
    "rrf": {
     "retrievers": [{
          "standard" { "query": {
              "match": {...}
        },{
          "standard" { "query": {
              "sparse vector": {...}
        },{
          "knn": { ... }
```


ChatGPT

Elastic and LLM

LLM: opportunities and limits

11:39 C ChatGPT 3.5 > You What is the current version of Elasticsearch? ChatGPT As of my last knowledge update in January 2022, the latest stable version of Elasticsearch was 7.10. However, software versions may have changed since then. I recommend checking the official Elasticsearch website or other reliable sources for the most up-to-date information on the current version.

dife

You How much money do I have on my Revolut personal account?

ChatGPT Sorry, but I can't assist with that.

You Why?

ChatGPT

I don't have access to your personal information, including details about your Revolut account. My purpose is to respect user privacy and provide general information and assistance. If you have questions about your Revolut account, it's best to check directly through the Revolut app or contact their support.

-hE

😞 elastic

Retrieval Augmented Generation

Demo

Elastic Playground

Home	Online banking	Enviroment setup

me Transaction search Financial summary Customer support

Search your transactions:

This search is not enabled by Elastic and reflects the kirch of functionality available to customers today.

Submit

Date	Account	Description	Value	Opening balance	Closing balance
18/06/24	EL03-130981-Transmission	Inbound payment made from EL03-130981-Transmission, St.james's Plac (STJ): 864dce1b-bb95-47d5-87dd- 7d02f3b10c3f	7419.0	-825.0	6594.0
18/06/24	EL03-130981-Transmission	Purchase at merchant: Southeastern Grocers, LLC, location: Fayetteville,AR	82.0	6594.0	6512.0
18/06/24	EL03-130981-Transmission	Purchase at merchant: Müller Holding Ltd. & Co. KG, location: Glendale,AZ	188.0	6512.0	6324.0
17/06/24	EL03-130981-Transmission	Payment made from EL03-130981-Transmission to Elwood Erickson, Mitie Grp. (MTO): d37085fc-1382-4593-9cb8- 26e5526bd9a0	533.0	20.0	-513.0
17/06/24	EL03-130981-Transmission	Payment made from EL03-130981-Transmission to Classie Johns, Barclays (BARC): 75b603a2-1c1b-45e9-a7ec- 4a551bf98a8d	312.0	-513.0	-825.0
16/06/24	EL03-130981-Transmission	Purchase at merchant: E-MART Inc., location: Fayetteville,AR	31.0	51.0	20.0
14/06/24	EL03-130981-Transmission	Purchase at merchant: Dick's Sporting Goods, Inc., location: Montgomery,AL	182.0	329.0	147.0
14/06/24	EL03-130981-Transmission	Purchase at merchant: Valor Holdings Co., Ltd., location: Louisville,KY	96.0	147.0	51.0
13/06/24	EL03-130981-Transmission	Purchase at merchant: The Save Mart Companies, location:	34.0	363.0	329.0

Elasticsearch

You Know, for Semantic Search

Search a new era

