



La **recherche** à l'ère de l'**IA**

David Pilato | [@dadoonet](#)



Agenda

- La recherche "classique" et ses limites
- Modèle de ML et usages
- La recherche vectorielle ou hybride dans Elasticsearch
- OpenAI ChatGPT ou LLM avec Elasticsearch

Elasticsearch

You Know, for Search



Elasticsearch

لُصْفَة



66

These are not the droids
you are looking for.

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

These are `not` 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
}]}
```

Recherche
sémantique

≠

Correspondance
littérale

**YOU'RE COMPARING
APPLES TO NECTARINES**



Elasticsearch

You Know, for **Vector** Search

Qu'est-ce qu'un
Vecteur ?

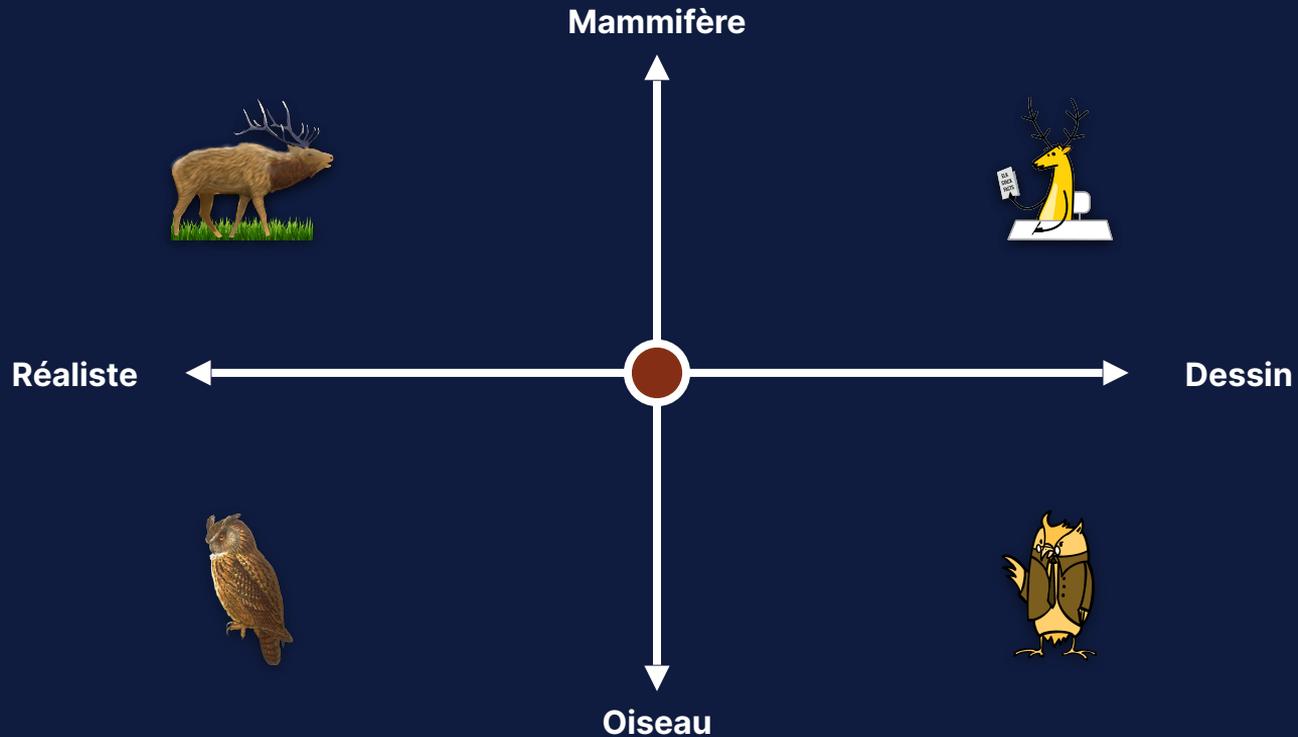


Les embeddings représentent vos données

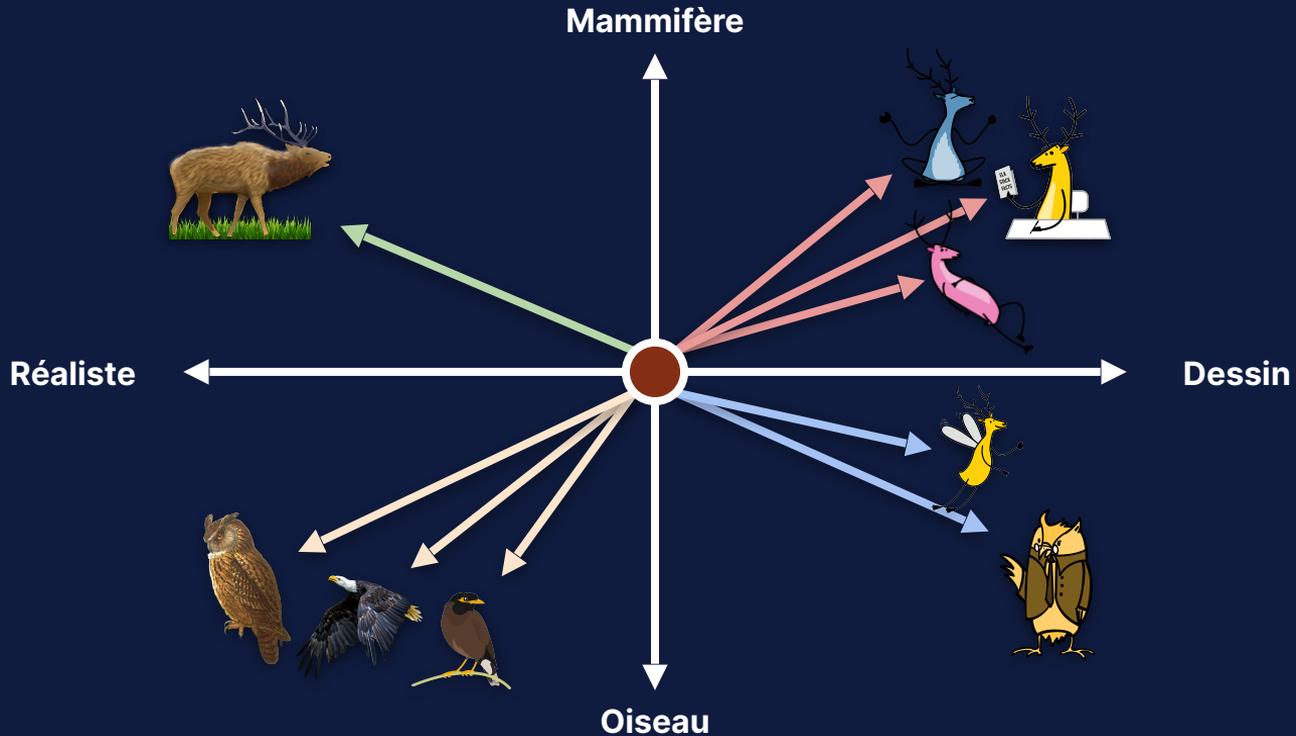
Exemple : vecteur 1 dimension



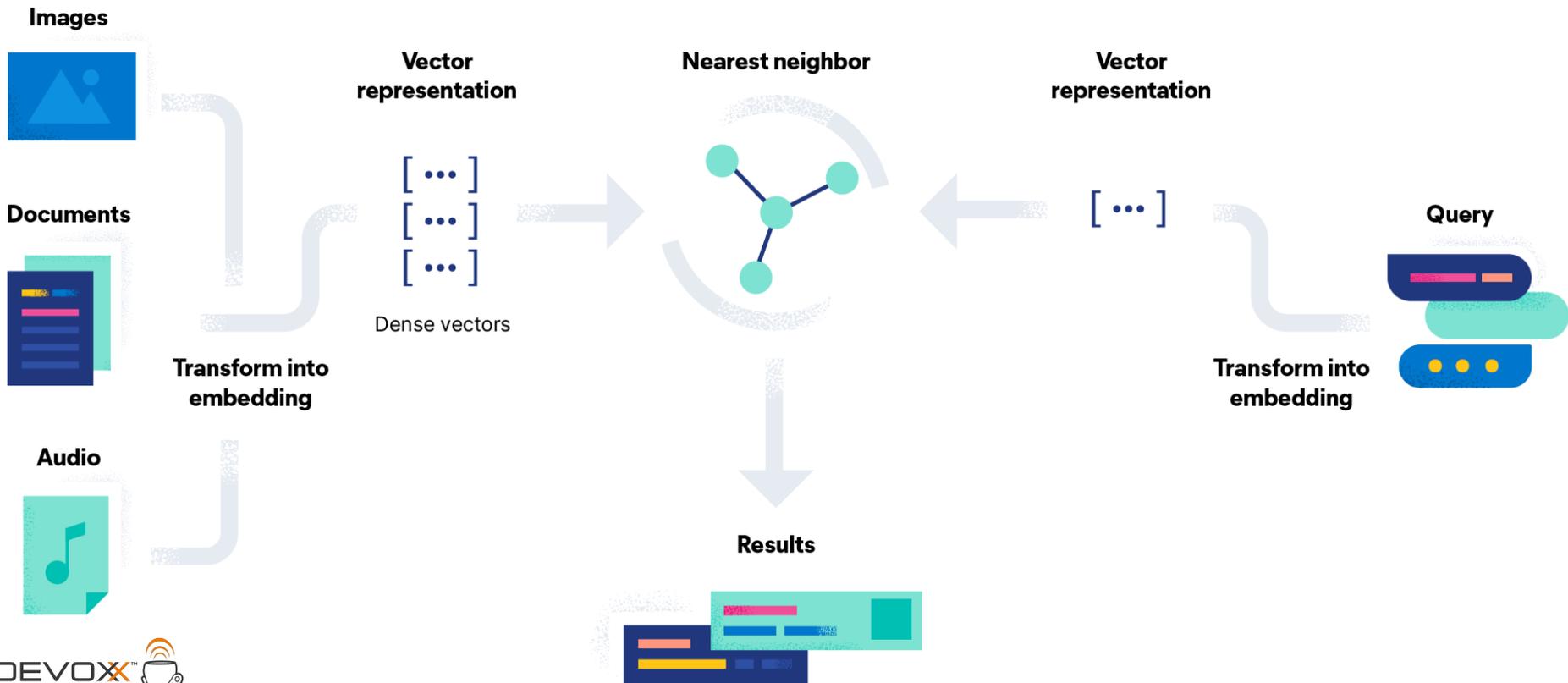
Plusieurs dimensions pour représenter plusieurs aspects



Dans l'espace des embeddings les données similaires sont regroupées



Recherche vectorielle



Choisir son modèle d'Embedding

Commencer avec des modèles sur étagère

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

Développer pour une plus forte pertinence

- Appliquer un scoring hybride
- Bring Your Own Model :
nécessite de l'expertise +
des données labelisées

Comment indexer

avec des **vecteurs** ?

Ingestion des données avec embeddings

You asked, we answered: Our best-selling classic wrap dress now comes in a cotton poplin that's wear-all-day perfect. Bonus: stripes (our favorite).

FIT

- 39" from high point of shoulder

DETAILS

- Cotton
- Lined
- Machine wash
- Import



 **Source data**

POST /_doc



```
{
  "_id": "product-1234",
  "product_name": "Summer Dress",
  "description": "Our best-selling...",
  "Price": 118,
  "color": "blue",
  "fabric": "cotton"
}
```

Ingestion des données avec embeddings

You asked, we answered: Our best-selling classic wrap dress now comes in a cotton poplin that's wear-all-day perfect. Bonus: stripes (our favorite).

FIT

- 39" from high point of shoulder

DETAILS

- Cotton
- Lined
- Machine wash
- Import



Source data

PyTorch



python™

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, ...],
  "img_embedding": [0.012, 0.0, ...]
}
```

Avec Elastic ML

You asked, we answered: Our best-selling classic wrap dress now comes in a cotton poplin that's wear-all-day perfect. Bonus: stripes (our favorite).

FIT
• 39" from high point of shoulder

DETAILS
• Cotton
• Lined
• Machine wash
• Import



Source data



POST /_doc

ML Inference pipelines

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

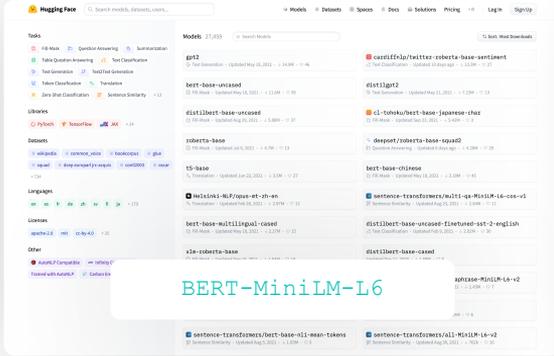
- ml-inference-embedding-generation**
Deployed | pytorch | text_embedding
- ml-inference-emational-analysis**
Deployed | pytorch | text_classification

Learn more about deploying ML models in Elastic

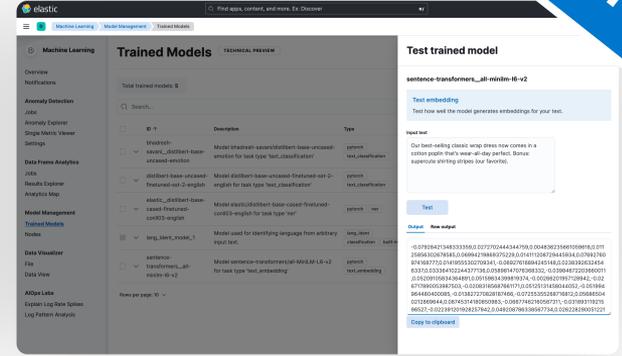
```
{  
  "_id": "product-1234",  
  "product_name": "Summer Dress",  
  "description": "Our best-selling...",  
  "Price": 118,  
  "color": "blue",  
  "fabric": "cotton",  
  "desc_embedding": [0.452, 0.3242, ...]  
}
```



Configurer son modèle



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



Choisir le modèle approprié



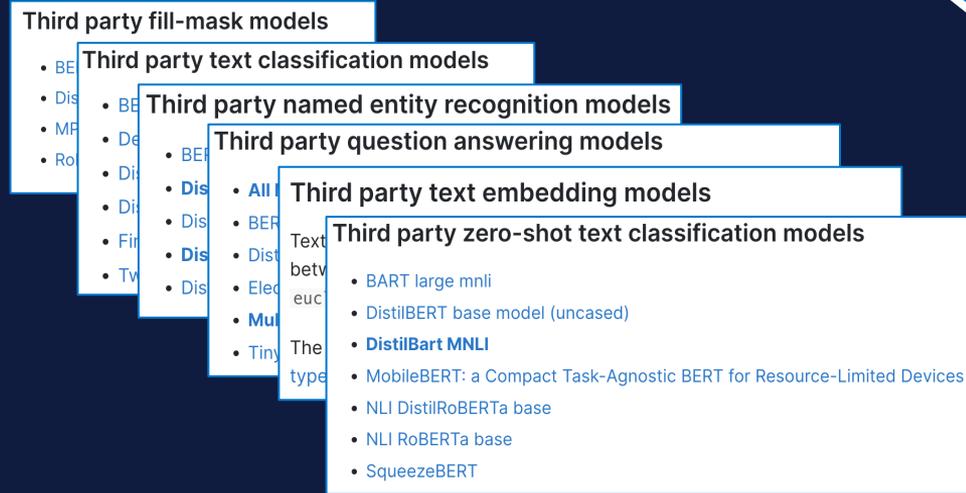
Charger le modèle dans le cluster



Gérer les modèles

Gestion des modèles

- C'est un domaine qui évolue rapidement. La flexibilité vous permet de vous adapter facilement.
- Utilisation des modèles tiers PyTorch
- Support de plusieurs types de modèles NLP



Liste complète des modèles supportés par Elastic : ela.st/nlp-supported-models

Comment chercher

avec des **vecteurs** ?

Requête Vectorielle

🔍 summer clothes | ✕ 🛒

PyTorch



python™

```
GET product-catalog/_search
{
  "knn": {
    "field": "desc_embedding",
    "k": 5,
    "num_candidates": 50,
    "query_vector": [0.123, 0.244, ...],
  },
  "filter": {
    "term": {
      "department": "women"
    }
  },
  "size": 10
}
```

Requête Vectorielle



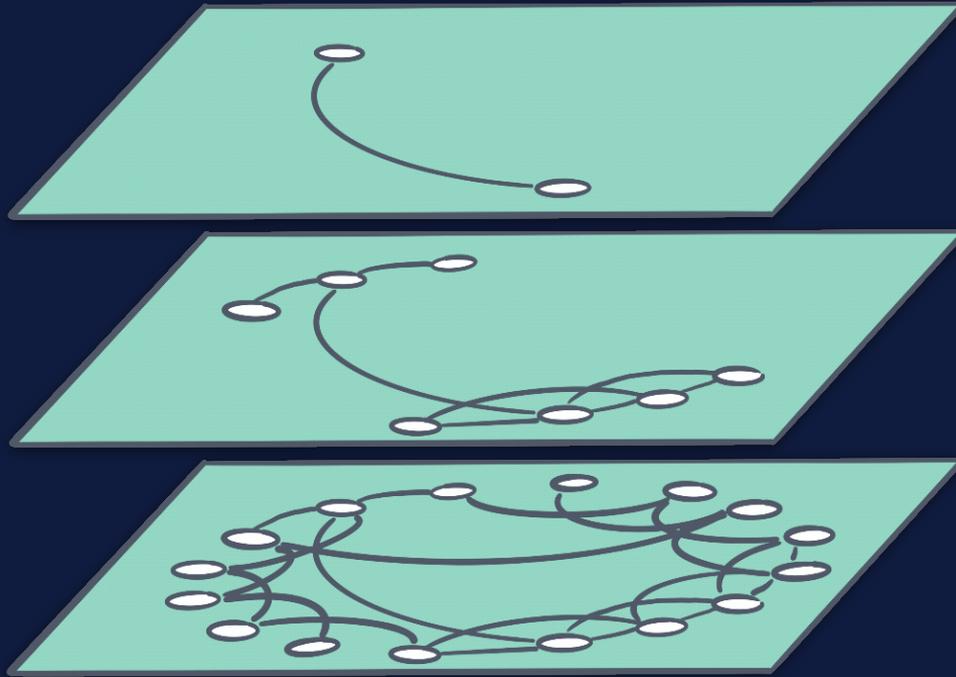
Transformer model

 PyTorch

```
GET product-catalog/_search
{
  "knn": {
    "field": "desc_embedding",
    "k": 5,
    "num_candidates": 50,
    "query_vector_builder": {
      "text_embedding": {
        "model_text": "summer clothes",
        "model_id": <text-embedding-model>
      }
    },
    "filter": {
      "term": {
        "department": "women"
      }
    }
  },
  "size": 10
}
```

Mais comment ça fonctionne en vrai ?

Hierarchical Navigable Small Worlds (HNSW)



Elasticsearch + Lucene

des progrès rapides ❤️

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.



Increase max number of vector dims to 2048 ...

9746994

“Scaler” la recherche vectorielle

Recherche vectorielle

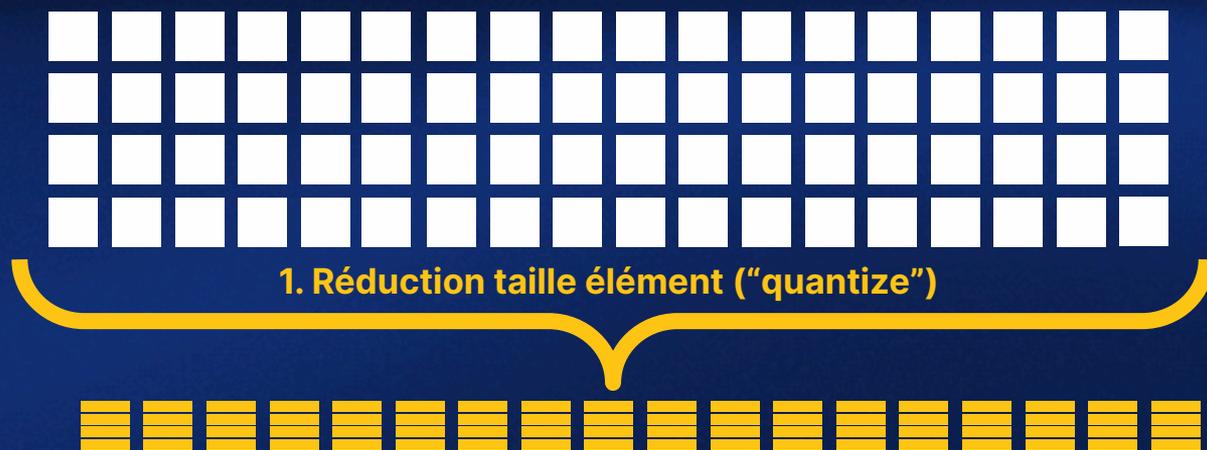
1. Besoin énorme de mémoire
2. Indexation plus lente
3. Merge est plus lent

* Améliorations permanentes dans Lucene et Elasticsearch

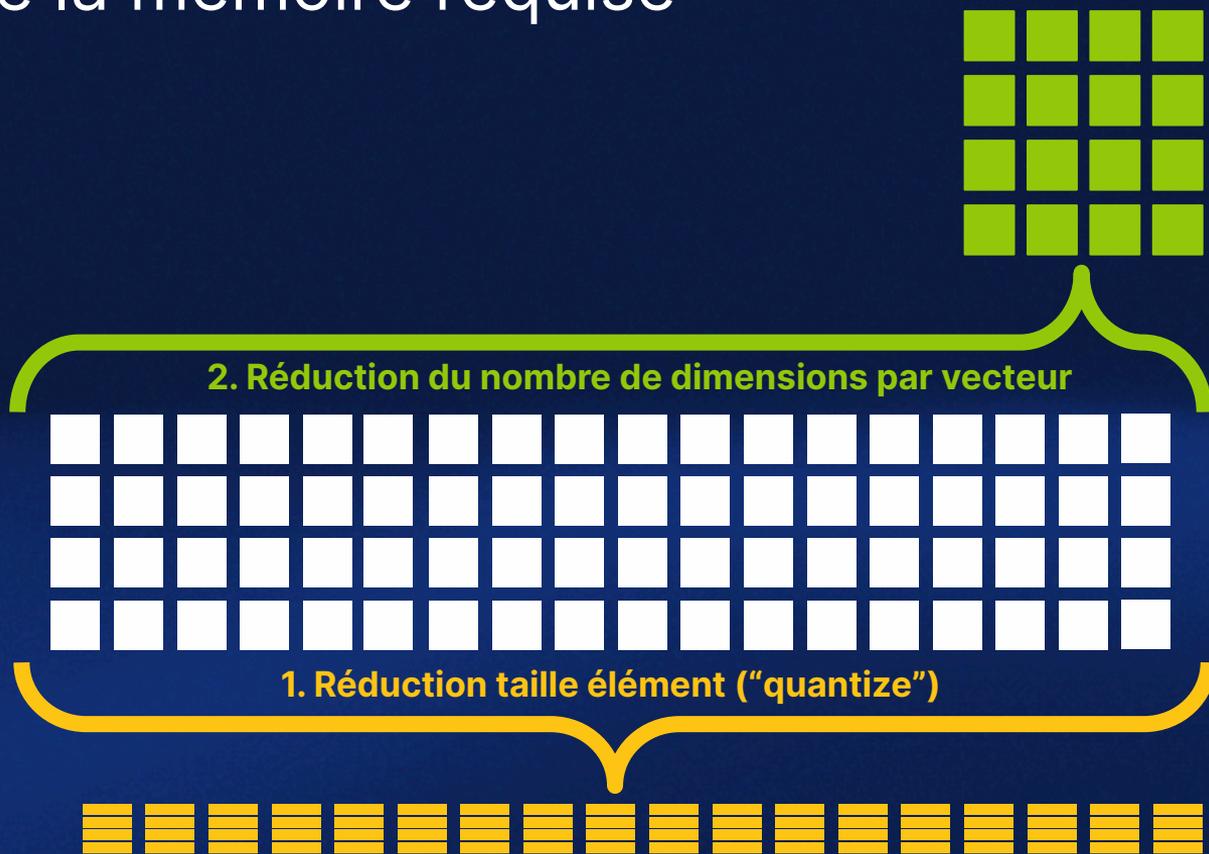
Bonnes pratiques

1. Eviter les recherches pendant l'indexation
2. Exclure les vecteurs du champ `_source`
3. Réduire le nombre de dimensions des vecteurs
4. Utiliser des bytes plutôt que des float

Réduire la mémoire requise

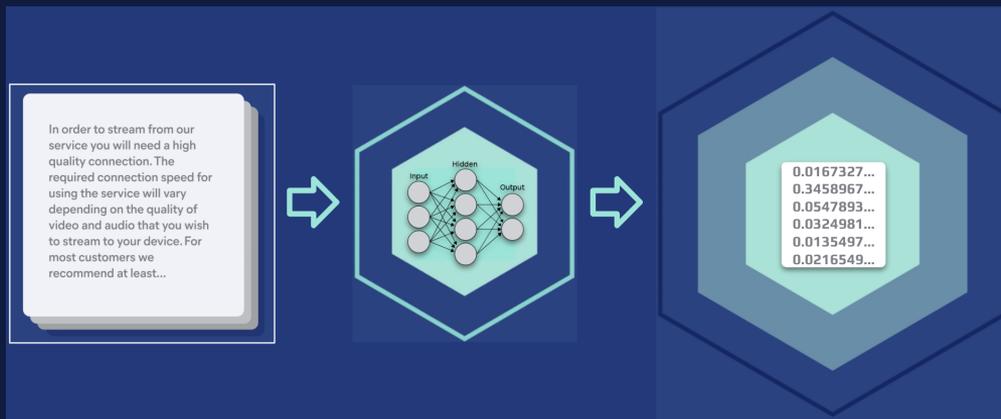


Réduire la mémoire requise



Base de données vectorielles

- **Stockage efficace** de vecteurs numériques, support des opérations **CRUD**
- **Recherche rapide** de vecteurs
- Conçu pour de la recherche vectorielle à grande échelle (**scalable**)



Elasticsearch

bien plus qu'une base de données vectorielles

Elasticsearch est capable de :

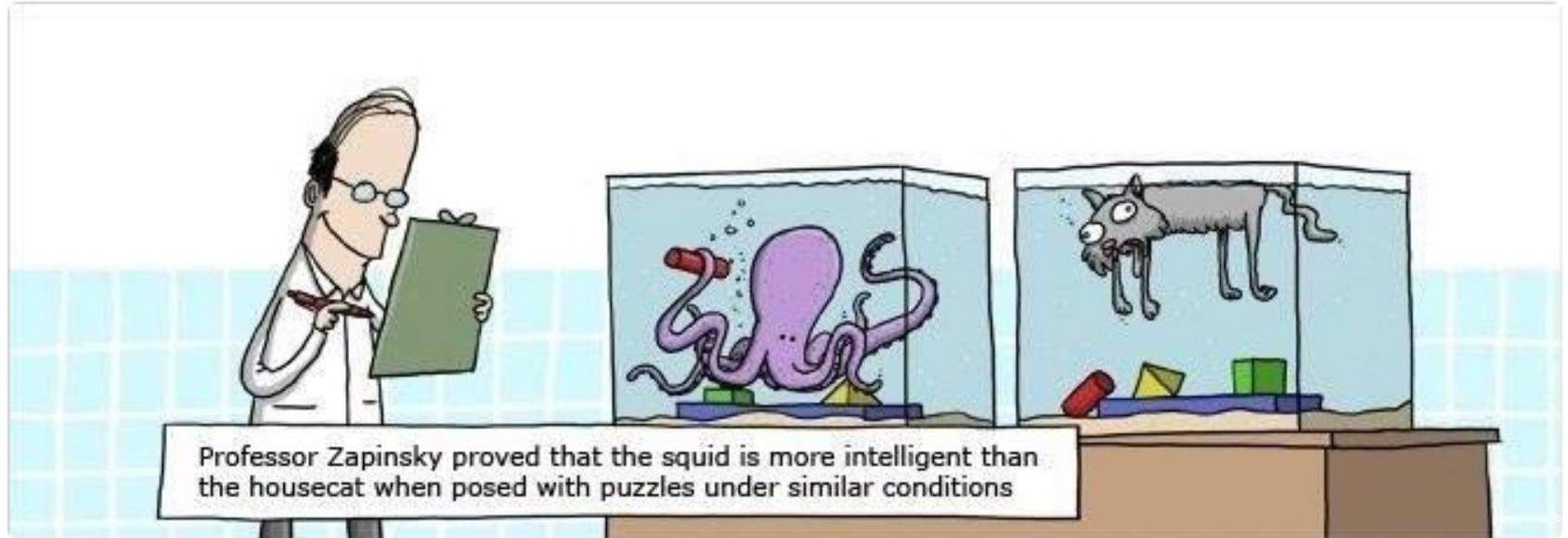
- **Créer des vector embeddings** (représentations numériques des données)
- Est **optimisé pour stocker** des vecteurs éparpillés et denses, en volume

Elasticsearch a aussi :

- **Filtres et agrégations** : sur l'ensemble des résultats
- **Sécurité native au niveau document*** : utilisé en production par les clients entreprise
- **Types de données** : Geo, full text, support des langues
- **Outils d'ingestion** : connecteurs, API, crawler web et des intégrations tierces
- Communauté, adoption massive par les entreprises, Track record, Elastic stack...

* nécessite une license commerciale

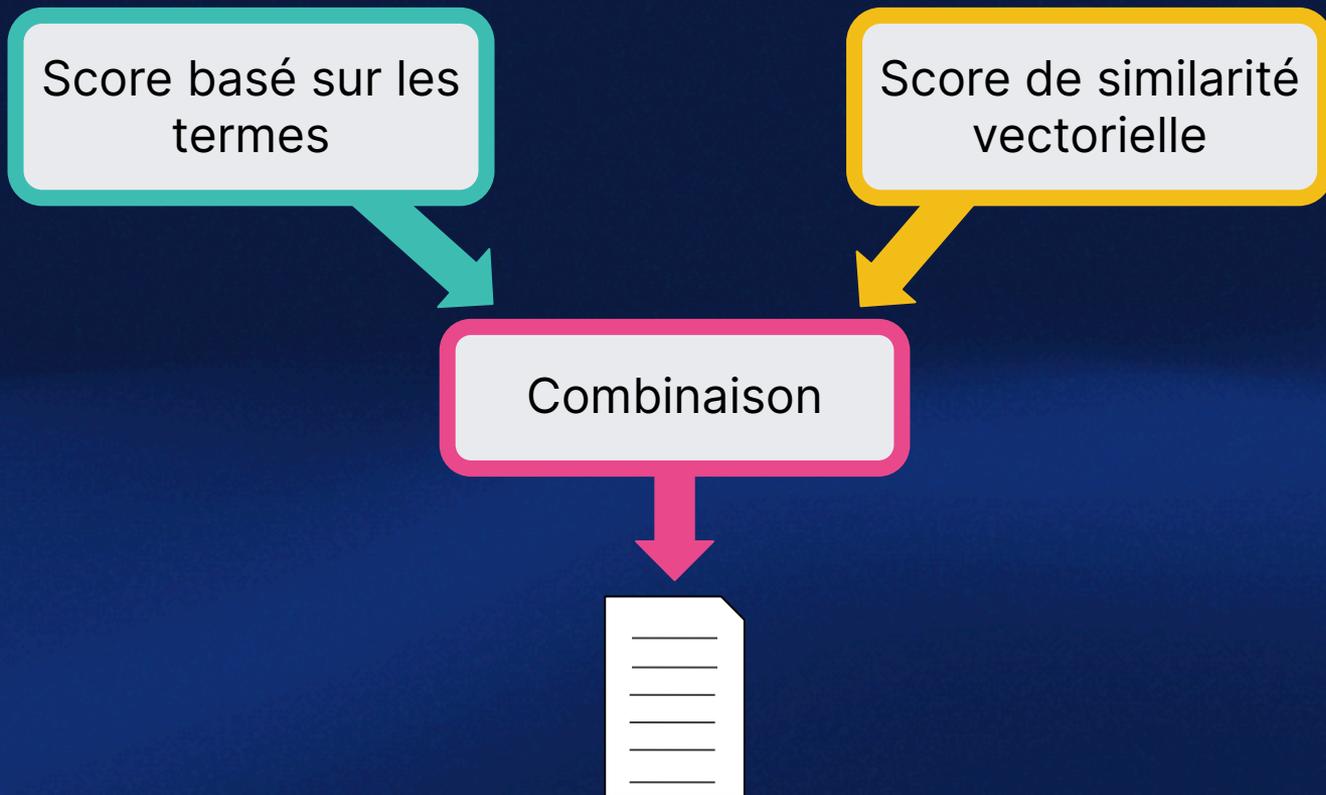
Benchmarking



Elasticsearch

You Know, for **Hybrid** Search

Calcul hybride du score de pertinence



```
GET product-catalog/_search
{
  "query": {
    "match": {
      "description": {
        "query": "summer clothes",
        "boost": 0.9
      }
    }
  },
  "knn": {
    "field": "desc_embedding",
    "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 }
}
```

ELSER

Elastic Learned Sparse Encoder

text_expansion

Not BM25 or (dense) vector

Sparse vector like BM25

Stored as inverted index

Commercial

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](#) 

ELSER - pertinence sur étagère, multi domaines

- Permet d'implémenter la recherche sémantique sans avoir à entraîner votre propre modèle
- Généraliste : plein de domaines sans entraînement
- Excellente pertinence, sur étagère

Search results ranking models

Data sets (BEIR benchmark)

[Démó](#)

	Average	TREC-COVID	NFCorpus	NQ	HotpotQA	FIQA	ArguAna	Touche-2020	DBPedia	SCIDOCs	FEVER	Climate-FEVER	SciFact
BM25	0.416	0.688	0.327	0.326	0.602	0.254	0.472	0.347	0.287	0.165	0.649	0.186	0.69
RRF (BM25/Dense)	0.449	0.697	0.317	0.445	0.611	0.318	0.474	0.354	0.353	0.159	0.746	0.238	0.671
Linear (BM25/Dense)	0.471	0.787	0.335	0.485	0.62	0.341	0.444	0.346	0.378	0.164	0.778	0.272	0.698
SPLADE	N/A	0.726	0.347	0.537	0.687	0.347	0.526	0.246	0.436	0.158			0.703
Elastic Learned Sparse Encoder	0.471	0.747	0.351	0.524	0.67	0.339	0.5	0.263	0.415	0.156	0.777	0.218	0.695
RRF (BM25 / Elastic Learned Sparse Encoder)	0.478	0.797	0.352	0.468	0.674	0.311	0.497	0.347	0.411	0.166	0.762	0.24	0.712

Search quality *worse* than benchmark (BM25) █ Search quality *better* than benchmark (BM25) █

```
POST /_ingest/pipeline/_simulate
{
  "pipeline":{"processors":[
    {
      "inference":{"model_id":".elser_model_1"}
    }
  ]},
  "docs": [
    {"_source":{"text_field":"These are not the droids you are looking for."}},
    {"_source":{"text_field":"Obi-Wan never told you what happened to your father."}},
    {"_source":{"text_field": "No. I am your father!"}}
  ]
}
```

66

These are not the droids you are looking for.

```
"ml": {  
  "inference": {  
    "predicted_value": {  
      "lucas": 0.50047517,  
      "ship": 0.29860738,  
      "dragon": 0.5300422,  
      "quest": 0.5974301,  
      "dr": 2.1055143,  
      "space": 0.49377063,  
      "robot": 0.40398192,  
      ...  
    }  
  }  
}
```

Combinaison de score

Commercial

Score basé sur les termes

Score de similarité vectorielle

Extension du score ELSER

Combinaison linéaire
Boosting manuel

Combinaison

Reciprocal Rank Fusion (RRF)
Mélange de plusieurs méthodes de ranking



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	RRF Score
A	$1/61 + 1/62 = 0,0325$
C	$1/63 + 1/61 = 0,0323$
B	$1/62 = 0,0161$
F	$1/63 = 0,0159$
D	$1/64 = 0,0156$

```
GET product-catalog/_search
{
  "sub_searches": [
    {
      "query": {
        "match": {...}
      }
    },
    {
      "query": {
        "text_expansion": {...}
      }
    }
  ],
  "knn": {...},
  "rank": {
    "rrf": {
      "window_size": 50,
      "rank_constant": 20
    }
  }
}
```

BM25f

+

ELSER

+

Vector



Recherche Hybride avec RRF

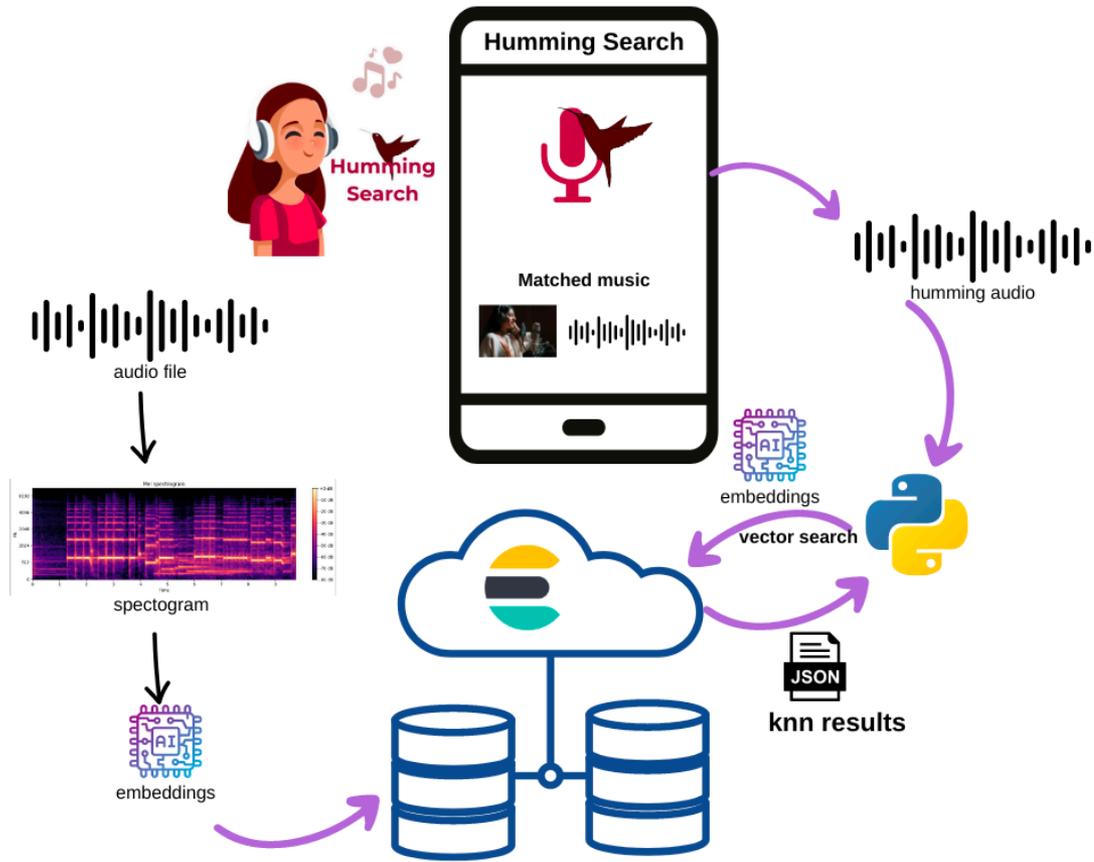
Gratuit vs Payant (platinum)

kNN with HNSW

1. Fournissez vos propres embeddings
2. Attention à la mémoire requise pour faire tenir les vecteurs en RAM (off-heap)

Inference in Elasticsearch

1. Génération des embeddings (transformers HuggingFace, autres)
2. ELSER

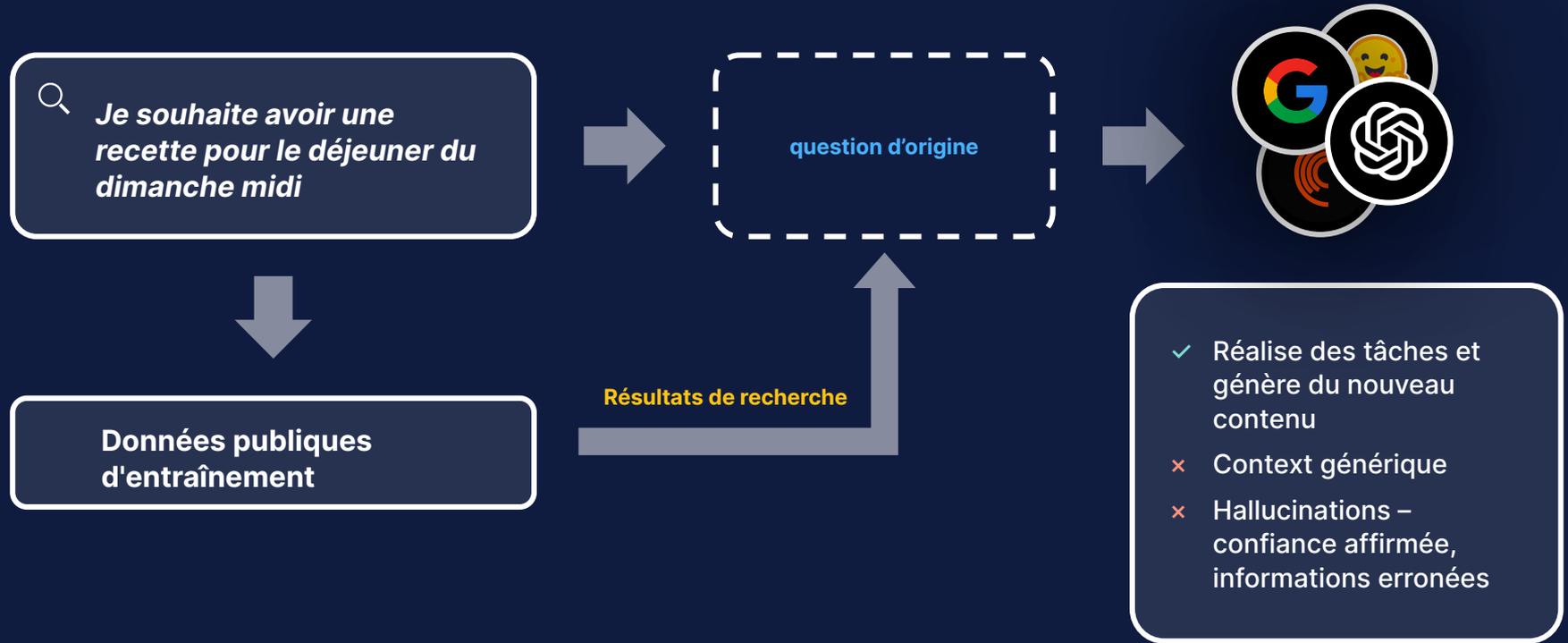


<https://github.com/dadoonet/music-search/>

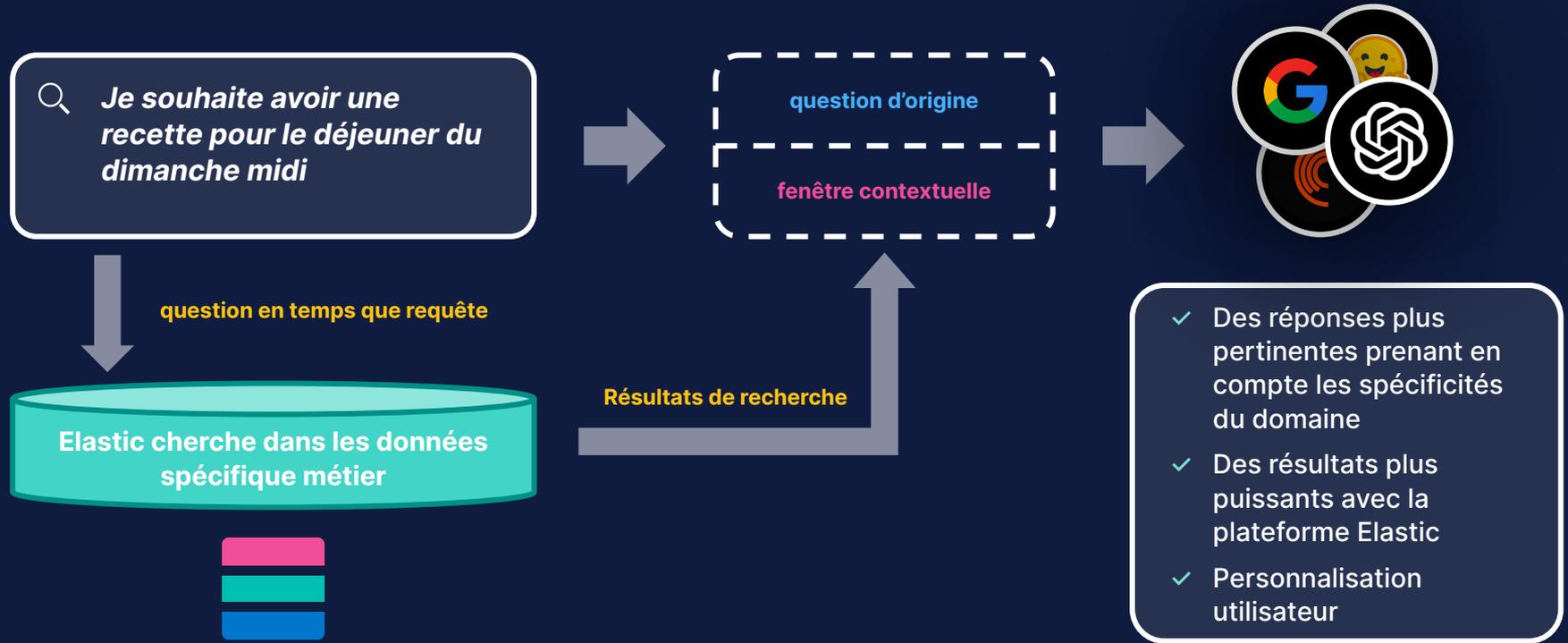
ChatGPT

Elastic et les LLMs

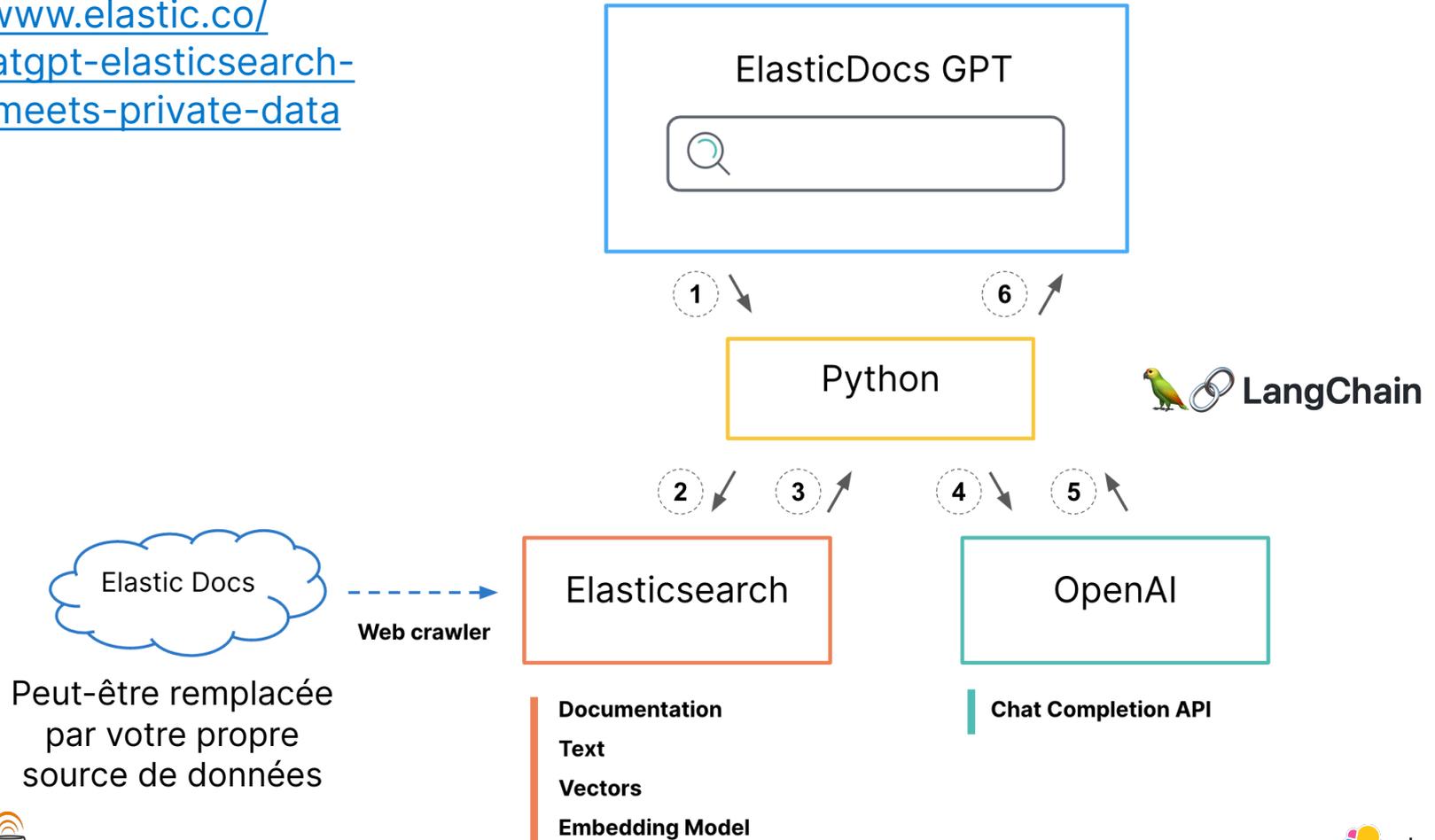
LLMs : opportunités et limitations



Elastic fournit une technologie complémentaire



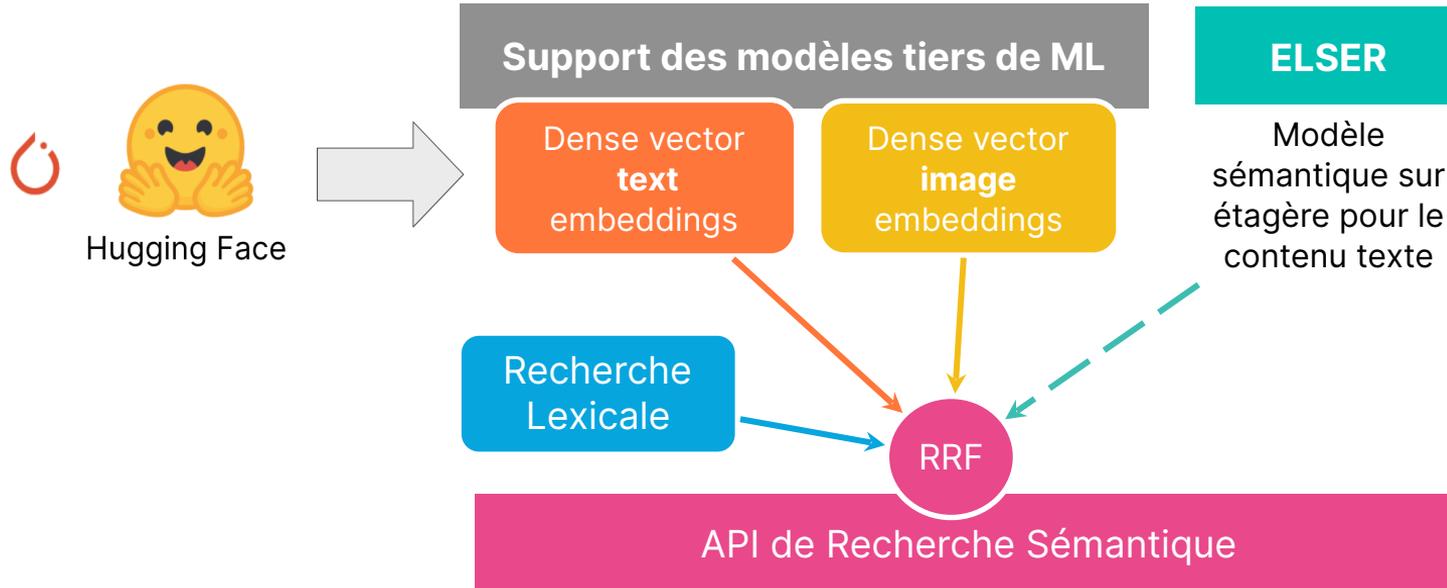
<https://www.elastic.co/blog/chatgpt-elasticsearch-openai-meets-private-data>



Peut-être remplacée
par votre propre
source de données

Conclusion

La recherche lexicale enrichie en sémantique



ESRE™ Elasticsearch Relevance Engine™

Composants de recherche

BM25F

Vector Database

Facets / Filtering

Intégration et Gestion de modèles

Modèles de
transformation custom

Intégration OpenAI

**Elastic Learned
Sparse Encoder**

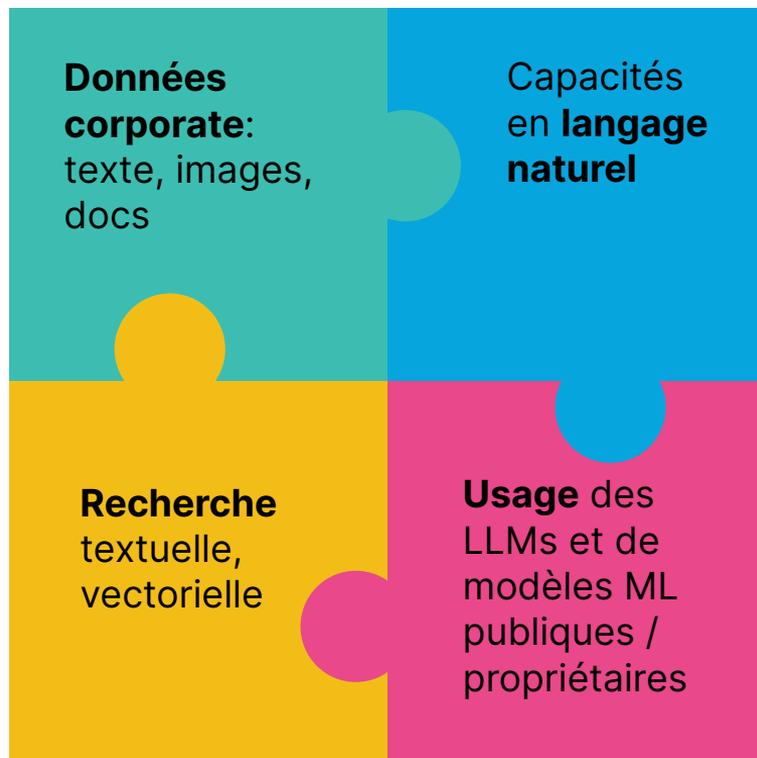
Recherche Hybride

RRF

Facettes

Combinaison
linéaire

Les ingrédients nécessaires pour la recherche IA...



... sont les
fonctionnalités
que nous livrons
avec ESRE™

**Ingestion,
Sécurité niveau
document** pour
les données
corporate

**Recherche
Vectorielle**
permettant
l'usage du
langage
naturel

**Recherche
Hybride**
(BM25,
Vecteur..)

**Flexibilité des
Modèles** pour
rester à jour :
nouveaux
modèles tiers et
LLM



La **recherche** à l'ère de l'**IA**

David Pilato | [@dadoonet](#)

