

# Project Description: Information Retrieval in a Medical Blog

## Group Members:

- Reda HIMMI
- Hamza EL AAFANI
- EL-FAILALI-LBALGHITI Mohamed
- Oussama DARIF
- Ismail OURAKH

## Introduction

Our application consists of a web medical blog. You can create; read and most importantly search about the relevant posts that will reply your query well. The marking feature of the application is the semantic search of the posts.

## Methodology

### Article scrapping

We scraped posts using Selenium.webdriver from <https://www.health.harvard.edu/blog>

	Title	Title_URL	Image	fontbold_URL	fontbold	block	Field
0	Magnets, sound, and batteries: Choosing safe toys	<a href="https://www.health.harvard.edu/blog/magnets-so...">https://www.health.harvard.edu/blog/magnets-so...</a>	<a href="https://domf5oi06qrcr.cloudfront.net/medialibr...">https://domf5oi06qrcr.cloudfront.net/medialibr...</a>	<a href="https://www.health.harvard.edu/topics/child-an...">https://www.health.harvard.edu/topics/child-an...</a>	Child & Teen Health	\n Updated December 13, 2023\n	If you're choosing gifts to give or donate to ...
1	No-cost, low-cost, and bigger splurges for cli...	<a href="https://www.health.harvard.edu/blog/no-cost-lo...">https://www.health.harvard.edu/blog/no-cost-lo...</a>	<a href="https://domf5oi06qrcr.cloudfront.net/medialibr...">https://domf5oi06qrcr.cloudfront.net/medialibr...</a>	<a href="https://www.health.harvard.edu/topics/staying-...">https://www.health.harvard.edu/topics/staying-...</a>	Staying Healthy	\n Published December 11, 2023\n	If you're looking for gifts to give or donate,...
2	What to do if you think your child has the flu	<a href="https://www.health.harvard.edu/blog/what-to-do...">https://www.health.harvard.edu/blog/what-to-do...</a>	<a href="https://domf5oi06qrcr.cloudfront.net/medialibr...">https://domf5oi06qrcr.cloudfront.net/medialibr...</a>	<a href="https://www.health.harvard.edu/topics/child-an...">https://www.health.harvard.edu/topics/child-an...</a>	Child & Teen Health	\n Updated September 12, 2023\n	If you hear your child start coughing, it's na...

### Encode medical articles using S-bert

#### Tokenization:

Tokenize the medical articles to convert them into a format suitable for input to the S-BERT model.

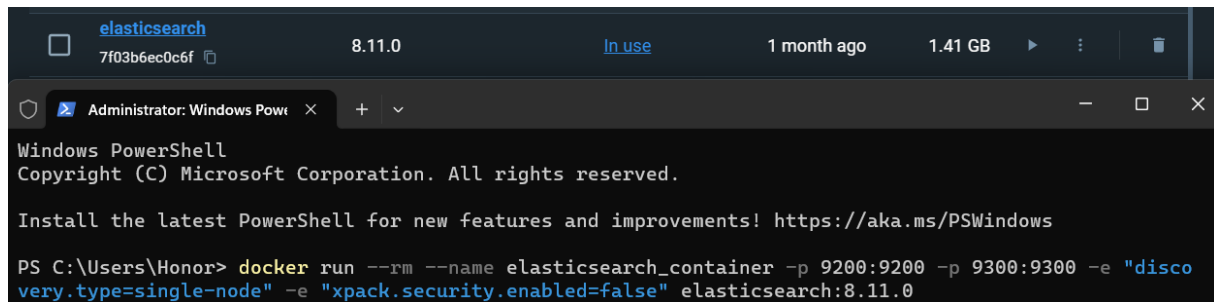
#### Sentence Embedding:

Use the S-BERT model to obtain embeddings for each sentence in the medical articles. S-BERT captures semantic similarity by mapping sentences into a high-dimensional vector space.

```
df.description_vector[:5]
0      [-0.0070151784, 0.0033397812, -0.018809972, 0....
1      [0.041044563, 0.05344586, 0.017541107, 0.07421...
2      [-0.01417845, 0.00091768993, -0.0012399706, 0....
3      [-0.0047708116, 0.0065720384, 0.013290583, -0....
4      [0.025648113, 0.044721454, 0.010994425, -0.055...
Name: description_vector, dtype: object
```

### Indexation automatic using elasticsearch

1. Install and Run Elasticsearch in docker :



The screenshot shows a Docker container named 'elasticsearch' with ID '7f03b6ec0c6f' running version '8.11.0'. The container is 'In use' and was created '1 month ago' with a size of '1.41 GB'. Below the container info, a Windows PowerShell terminal window is open, showing the command to run the container: `docker run --rm --name elasticsearch_container -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" -e "xpack.security.enabled=false" elasticsearch:8.11.0`. The terminal output shows the PowerShell prompt and the command execution.

```
elasticsearch
7f03b6ec0c6f 8.11.0 In use 1 month ago 1.41 GB
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Honor> docker run --rm --name elasticsearch_container -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" -e "xpack.security.enabled=false" elasticsearch:8.11.0
```

## 2. Set Up an Index Mapping:

```
indexMapping = {
    "properties" : {
        "condition_label" : {
            "type" : "long"
        },
        "medical_abstract" : {
            "type" : "text"
        },
        "description_vector" : {
            "type" : "dense_vector",
            "dims" : 768,
            "index" : True,
            "similarity" : "l2_norm" # dist euclid
        },
    },
}
```

## 3. Index the Document:



The screenshot shows a Jupyter Notebook with the title 'Create new index'. It contains several code cells. The first cell imports the 'indexMapping' from a module. The second cell creates a new index named 'all\_documents' with the 'indexMapping'. The third cell converts a DataFrame 'df' to a list of dictionaries 'records\_list'. The fourth cell is a loop that indexes each record in 'records\_list' into the 'all\_documents' index, with error handling. The fifth cell counts the number of documents in the index. The output of the last cell is an 'ObjectApiResponse' showing a count of 180 documents.

```
Create new index

In [26]: from indexMapping import indexMapping

In [27]: es.indices.create(index="all_documents", mappings= indexMapping)

...

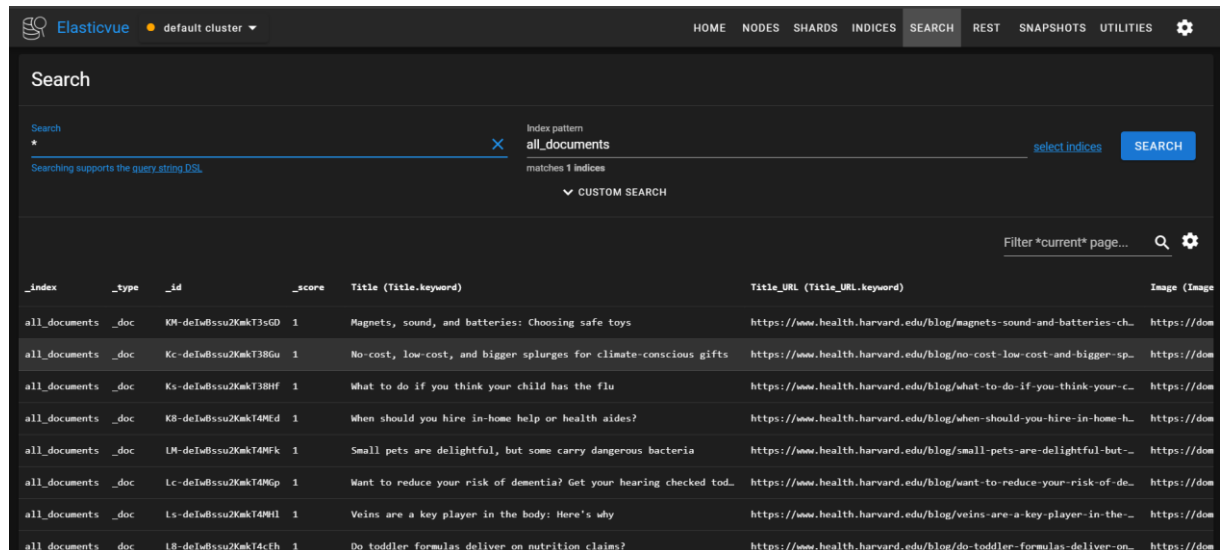
In [32]: records_list = df.to_dict("records")

In [33]: for record in records_list:
    try:
        es.index(index="all_documents", document=record)
    except Exception as e:
        print(e)

In [34]: es.count(index="all_documents")

Out[34]: ObjectApiResponse({'count': 180, '_shards': {'total': 1, 'successful': 1, 'skipped': 0, 'failed': 0}})
```

## The index table from ElasticVue



The screenshot shows the ElasticVue search interface. At the top, there's a navigation bar with links: HOME, NODES, SHARDS, INDICES, SEARCH (active), REST, SNAPSHOTS, UTILITIES. Below the navigation bar, the 'Search' section is active. It shows a search bar with a query string, an index pattern 'all\_documents', and a 'SEARCH' button. Below the search bar, there's a table of search results. The table has columns: \_index, \_type, \_id, \_score, Title (Title.keyword), Title\_URL (Title\_URL.keyword), and Image (Image). The table contains 8 rows of results, all from the 'all\_documents' index and '\_doc' type. The results are sorted by '\_score' in descending order.

_index	_type	_id	_score	Title (Title.keyword)	Title_URL (Title_URL.keyword)	Image (Image)
all_documents	_doc	KM-deIaBssu2KakT3sGD	1	Magnets, sound, and batteries: Choosing safe toys	https://www.health.harvard.edu/blog/magnets-sound-and-batteries-ch...	https://dom
all_documents	_doc	Kc-deIaBssu2KakT3BGu	1	No-cost, low-cost, and bigger splurges for climate-conscious gifts	https://www.health.harvard.edu/blog/no-cost-low-cost-and-bigger-sp...	https://dom
all_documents	_doc	Ks-deIaBssu2KakT3BHf	1	What to do if you think your child has the flu	https://www.health.harvard.edu/blog/what-to-do-if-you-think-your-c...	https://dom
all_documents	_doc	K8-deIaBssu2KakT4MEd	1	When should you hire in-home help or health aides?	https://www.health.harvard.edu/blog/when-should-you-hire-in-home-h...	https://dom
all_documents	_doc	UM-deIaBssu2KakT4Mfk	1	Small pets are delightful, but some carry dangerous bacteria	https://www.health.harvard.edu/blog/small-pets-are-delightful-but-...	https://dom
all_documents	_doc	Lc-deIaBssu2KakT4Mfp	1	Want to reduce your risk of dementia? Get your hearing checked tod...	https://www.health.harvard.edu/blog/want-to-reduce-your-risk-of-do...	https://dom
all_documents	_doc	Ls-deIaBssu2KakT4Mfi	1	Veins are a key player in the body: Here's why	https://www.health.harvard.edu/blog/veins-are-a-key-player-in-the-...	https://dom
all_documents	_doc	U8-deIaBssu2KakT4cFh	1	Do toddler formulas deliver on nutrition claims?	https://www.health.harvard.edu/blog/do-toddler-formulas-deliver-on...	https://dom

### 4. Querying the Index:

```
query = {
  "field": "description_vector",
  "query_vector": vector_input,
  "k": 5,
  "num_candidates": 60,
}
```

### 5. Matching the query with document

```
res = es.knn_search(index="all_documents", knn=query, source=["Title", "Field", "Image"])

results = res["hits"]["hits"]

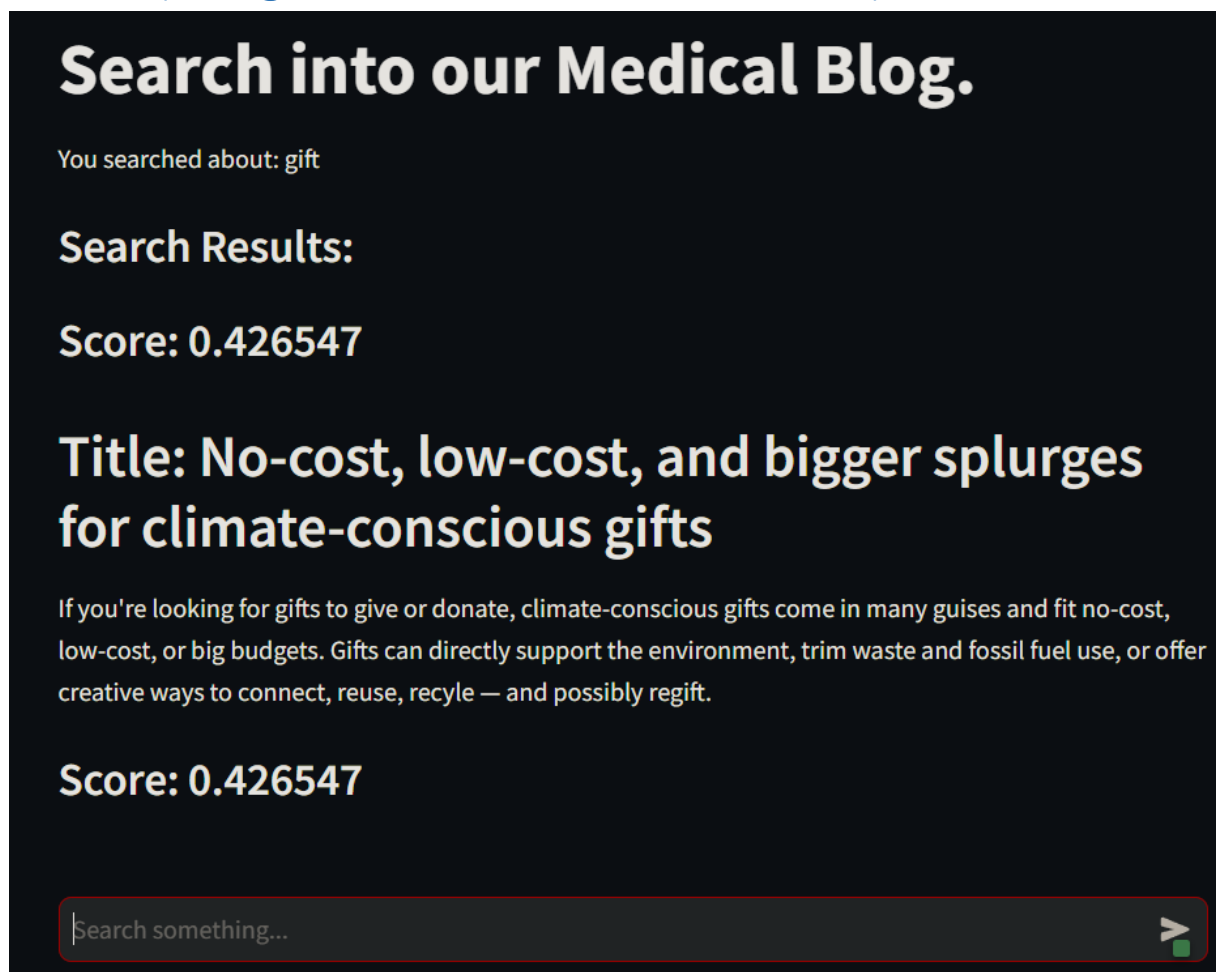
for result in results:
    if "_source" in result:
        try:
            print(f"Document score: {result['_score']}")
            print(f"Document Title: {result['_source']['Title']}")
            print(f"Document Text: {result['_source']['Field']}")
            print(50*" ")
        except Exception as e:
            print(e)
```

To Match the the relevant document with the query we used the eucliden distance in the similarity field of our Mapping in elasticsearch, avec le classement des resultat selon cette distance

## UI and Backend part

We used Spring Boot for the backend of our website and ReactJS for the frontend.

Result (using streamlit to test the model)



## Future Work

We will parse the index from MongoDB to Elasticsearch using logStach

Integrate the search in the image too by creating a description from the image then index this description and make it available for searching