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	https://www.health.harvard.edu/blog/magnets-so...	https://domf5oio6qrcr.cloudfront.net/medialibr...	https://www.health.harvard.edu/topics/child-an...	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...	https://www.health.harvard.edu/blog/no-cost-lo...	https://domf5oio6qrcr.cloudfront.net/medialibr...	https://www.health.harvard.edu/topics/staying-...	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	https://www.health.harvard.edu/blog/what-to-do...	https://domf5oio6qrcr.cloudfront.net/medialibr...	https://www.health.harvard.edu/topics/child-an...	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 :

```
elasticsearch   8.11.0   In use   1 month ago   1.41 GB
7f03b6ec0c6f
Administrator: Windows Powe
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:

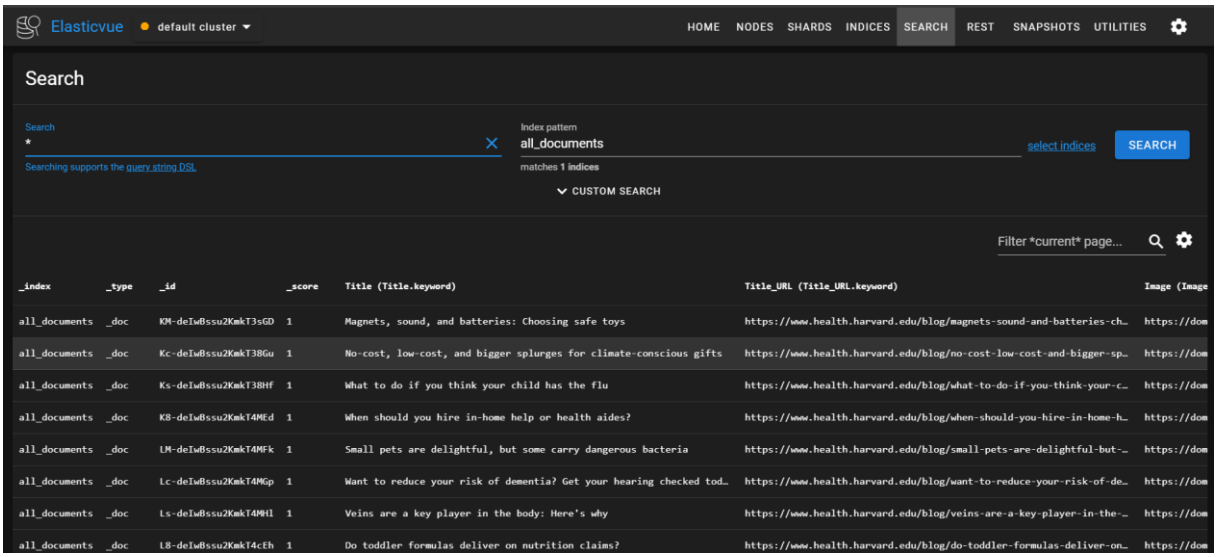
```
indexMapping = {
    "properties" : {
        "condition_label" : {
            "type" : "long"
        },
        "medical_abstract" : {
            "type" : "text"
        },
        "description_vector" : {
            "type" : "dense_vector",
            "dims" : 768,
            "index" : True,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

3. Index the Document:

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



_index	_type	_id	_score	Title (Title.keyword)	Title_URL (Title_URL.keyword)	Image (Image
all_documents	_doc	KM-deIwBssu2KmkT3sGD	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-deIwBssu2KmkT38Gu	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-deIwBssu2KmkT38Hf	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-deIwBssu2KmkT4MEd	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	LM-deIwBssu2KmkT4MFk	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-deIwBssu2KmkT4MGp	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-de…	https://dom
all_documents	_doc	Ls-deIwBssu2KmkT4MHl	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	L8-deIwBssu2KmkT4cEh	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 euclidien 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