Gossip Semantic Search

Your mission

vsd.fr and public.fr are among the most popular French people websites. Your mission is to create a semantic search engine for these sources. It should allow a user to search for anything and get, as a result, a list of vsd/public article links matching the search in the order you see fit.

What is semantic search?

Semantic search is an advanced search technique that aims to improve search accuracy by understanding the meaning of the query, rather than just matching keywords. Unlike traditional search methods that rely on exact word matches, semantic search considers the context, intent, and relationships between words to deliver more relevant results.

This can be achieved using the vectorization method. Queries and documents are converted into vectors (mathematical representations) that capture the semantic meaning. This allows the system to compare the "closeness" of ideas, even if the exact words differ.

What do you actually need to build?

In order to build a semantic search service, you will need to implement:

1. A script to import data which will probably
    a. Create a dataset of articles from vsd.fr and public.fr. This can be done by scraping the websites but here's a hint, it may not be the most efficient way of doing it ;)
    b. Use a sentence-transformer model to create an embedding for each article. Many tools and APIs are available out there for you to achieve this. If you need an API key to access a paid service, let us know!
    c. Store the embeddings for future use (search time). Again, many tools and APIs are available out there. If you need an API key, let us know!
2. A backend to
    a. Transform the query into an embedding
    b. Search for similarity

    c. Return the results along with associated metadata you think is relevant
3. A frontend for the user to search and get results

Requirements

Almost none. Do this like you were doing it for yourself using whatever language, methods, tools you think is relevant but keep in mind that the result should be a locally working solution with good quality code and a few tests. We think this can be achieved within 4 hours.

One last thing

Do not hesitate to reach out to denis@linkup.so if you have any question or feel stuck, and most importantly, have fun!!