

Documentation for P05

How to open:

Compile backend with Lucene, optionally add a command line argument to specify how many documents are being used, default being 1000. It'll run a RESTful server on <http://localhost:1234/> with the only route being /search, taking one query parameter, *query*, which is the query the user searches for.

In order to run the Backend.jar the following command can be issued:

```
java -jar Backend.jar [num_docs]
```

Where [num_docs] is an optional argument specifying the number of documents used, defaulting to 1000.

For the frontend install the necessary packages using npm, by running *npm install*, then run using *npm run serve* to start the web server. The website is accessible under <http://localhost:8080/>

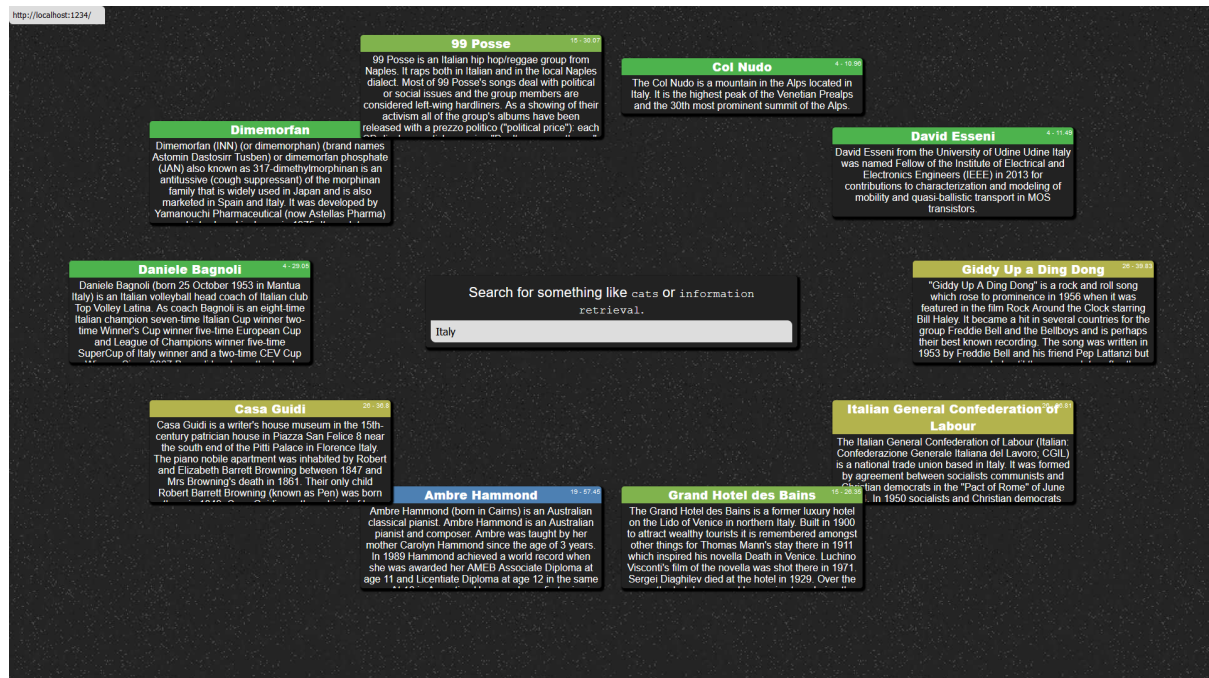
Frontend:

The Frontend is built using *Vue3*. It sends requests to the backend via *Axios*. These requests' answers follow this scheme:

```
{
  query: {
    classification: number,
    title: string,
    vector: number[]
  },
  result: {
    title: string,
    context: string,
    class: number,
    vector: number[]
  }
}
```

Using the given vectors of the query and results the distance is calculated using Euclidean distance and displayed in the top right of the cards. Each class has a different color to easily

distinguish them if multiple classes are returned. Each card shows the title of the document, a preview and the distance to the query. Clicking on a card will open a new tab with the Wikipedia article for the search result, by constructing the URL of the article using the title of the document. E.g. *Albert Einstein* opens https://en.wikipedia.org/wiki/Albert_Einstein.



Backend:

From the Wikipedia dataset topics are downloaded and composed into a csv file that contains these topics and their relevant body texts. After that tf and idf values are calculated for the documents. K-means-clustering algorithm is used in order to cluster documents. The initial clustering centers are initialized by selecting a random document. After that the euclidean distance between documents and central points are calculated and each document is clustered according to these distances. This process continues until clustering converges or a maximum amount of 100 iterations passed. The Spring Boot framework is used in order to create the web application and JSON serialization. After the “query” parameter is taken, its tf-idf vector and its distances to other documents is calculated. Finally, the JSON response which includes the query string and query vector and up to 10 closest document’s tf-idf vector, title, class and context is returned.