Design Document

Advanced Information Retrieval, WS24

Group 11		
First name	Last name	Matriculation Number
Maksim	Madžar	12415532
Naida	Nožić	12336462
Faruk	Šahat	12336460
Petra	Buršić	12408012

1 Introduction and Project Description

This project's goal is to create a semantic song retrieval system that is able to analyze the meaning of songs' lyrics and relate user queries to them. Without interpreting the meaning, keyword-based search often fails to recognize the relationships between user queries and song lyrics, leading to results that are incorrect or irrelevant. With this in mind, we will use transformer-based models to establish a connection of how one might express their music needs using natural language and how lyrics are understood computationally. We will achieve this by developing a song retrieval pipeline that fine-tunes pre-trained models so it could process song lyrics and user queries. By focusing on semantic understanding, the system will try to match songs even when the user misremembers the correct lyrics. The user can simply query the song description, theme, or context instead of providing exact phrases.

We will be using a dataset of song lyrics (datasets/saurabhshahane/music-dataset-1950-to-2019/data) from a seven-decade period (1950–2019), which will be utilized for both training and testing purposes.

Research Question: *How can fine-tuned transformer-based models be utilized to accurately retrieve songs based on lyrical semantics, even when user queries are incomplete, incorrect, or descriptive?*

2 Dataset + Processing

Due to copyright concerns, datasets of complete song lyrics are not frequent on the internet, and are mostly provided through APIs on a single-song basis. Luckily, there exist some workarounds - DBs that provide the lyric set as a Bag-of-Words model, arguing that it is transformative to the lyrics, and bases that collect words that describe the general feeling of the song. Both can be used for finding relevant songs if the queries are well-structured, but for the purposes of our project the former is a better solution.
The "Lyrics and Metadata from 1950 to 2019"[1] dataset will be used as we think it best suits our goals. It contains a vast collection of almost 24 thousand tracks from over five thousand authors, separated in csv format with the release date, genre and lyrics in the file. There are additional elements that describe the tracks further, such as a measure of energy, sadness, song topics etc., which may be used to extend and fine tune our search engine if the categories are fine-grained enough to permit that - but the focus of our search will be in the lyrics category, where there is a collection of a stemmed BoW (with repetition) for every song, sorted by order of appearance. The dataset is covered by the Creative Commons 4.0 community license, so we are free to use it with correct attribution. According to the description on the Mendeley Data website: *"The audio data was scraped using Echo Nest API integrated engine with spotipy Python's package. The spotipy API permits the user to search for specific genres, artists,songs, release date, etc. To obtain the lyrics we used the Lyrics Genius API as baseURL for requesting data based on the song title and artist name."*[2]
The original use for this database was for temporal analysis and visualization of music, with lyrics added to expand the amount of data. By focusing on the text and perhaps adding certain other categories from

[1] Source: Moura, Luan; Fontelles, Emanuel; Sampaio, Vinicius; França, Mardônio (2020), "Music Dataset: Lyrics and Metadata from 1950 to 2019", Mendeley Data, V3, doi: 10.17632/3t9vbwxgr5.3

[2] Description taken from the dataset page on Mendeley Data

the set, it should prove a comfortable fit for an advanced search of misheard and misinterpreted lyrics. To effectively enable semantic song retrieval, we need to embed the lyrics into a vector space in which we can do vector similarity search. The lyrics will be converted into numerical vector representations using pre-trained sentence transformers. These embeddings will capture the semantic meaning of the text. To do that, we will utilize sentence-transformers/all-MiniLM-L6-v2 to calculate embeddings. Alongside the vectorized lyrics, other matadata will be stored in the vector database as well.

3 Methods/Models

The song retrieval system relies on transformer-based models, in order to understand the meaning behind song lyrics and to be able to connect them with appropriate user queries. Since we cannot feed a neural network with raw text, we will need numbers. For this we will utilize pre-trained models, in our case the `sentence-transformers/all-mpnet-base-v2`, to encode text data into a vector embedding. The mentioned model is a transformer that maps sentences and paragraphs to a 768 dimensional dense vector space and can be used for clustering and semantic search. Depending on computational efficiency and available resources, we might opt to use `sentence-transformers/all-MiniLM-L6-v2`, which is 5 times faster and also good for general purpose semantic search. As an addition, we could include a method for query expansion by using ChatGPT. It would generate supplementary phrases or keywords to improve the user query context and overall recall.

Therefore, in initial retrieval the chosen sentence transformer will act as a Bi-Encoder and together with ChatGPT we will obtain the top k-songs with the most similar lyrics. Similarity will be determined using cosine-similarity.

Afterwards, to refine our results, we will utilize a Cross Encoder that will process the query and each retrieved lyric candidate together, in order to re-rank the results and provide the user with the appropriate song. The `cross-encoder/ms-marco-MiniLM-L-6-v2` is a good candidate for the task since it has been trained on the MSMARCO Passage Ranking Dataset which contains 500k real queries from Bing search together with the relevant passages from various web sources. Its diversity can work on a variety of domains, including our song retrieval system.

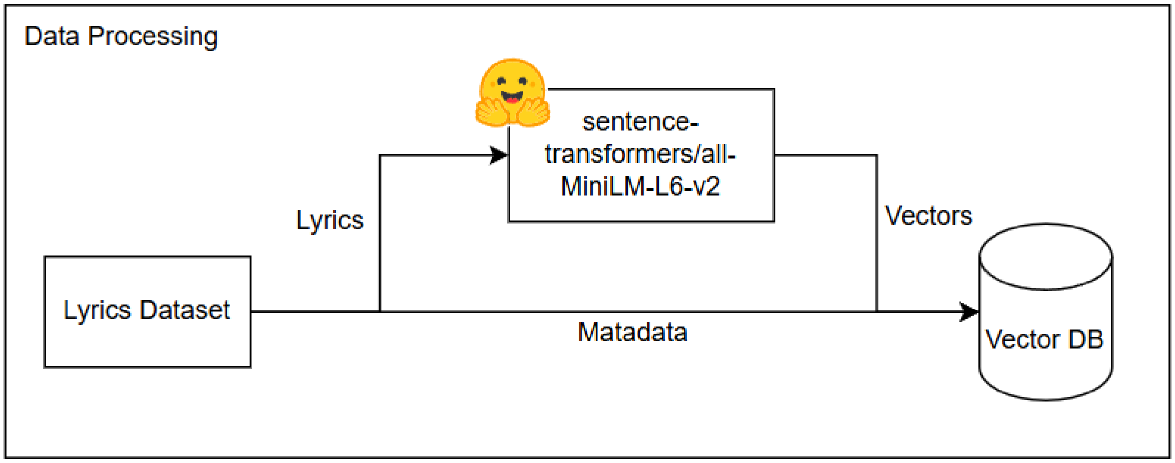

Figure 1: Data Processing

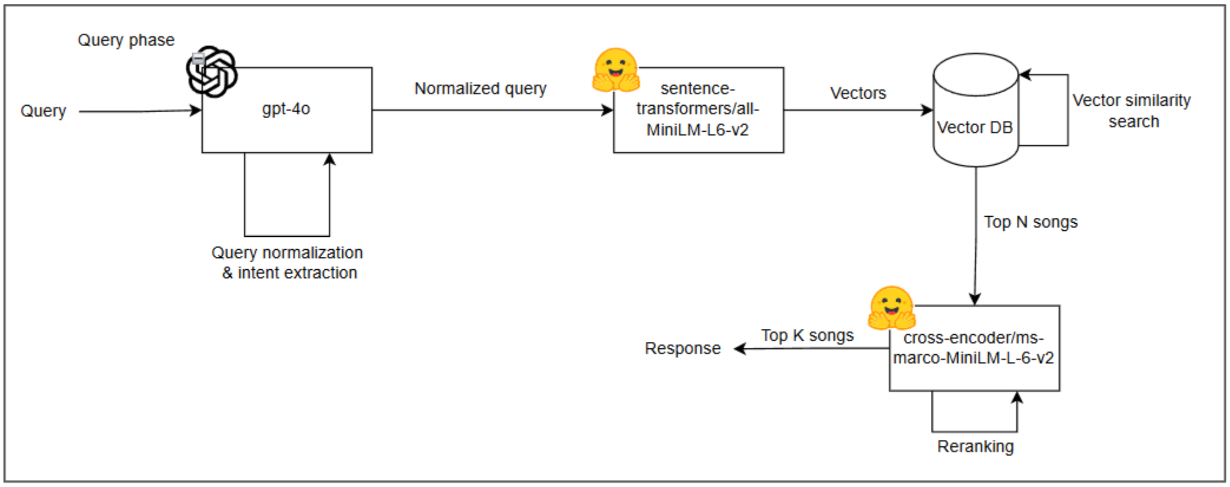

Figure 2: Query Phase

4 Evaluation

For evaluation, we will generate a separate smaller test dataset with simulated queries and appropriate relevance query scores. The queries will be created by selecting portions of the lyrics from the dataset and associating them with the corresponding songs. With random sampling and text preprocessing we will ensure diverse and realistic query generation.

After running our retrieval system (Bi-Encoder + Cross-Encoder pipeline) on the test dataset, we will use library built-in utilities for evaluating semantic search results. With the sentence-transformers and cross encoders we will use metrics such as Precision, Recall and Accuracy, as well as evaluating the overall performance and efficiency of the developed system. Due to the decision to include additional context to the queries with ChatGPT, we will evaluate the difference in the obtained results with and without the language model.

5 Responsibilities

The responsibilities that are enumerated in the table below have been divided roughly by our estimates of a fair workload. Even though this order is fixed for the initial stage of the project, roles can change due to expertise in a certain area or better dynamism. Any changes will be addressed in the post-development report.
For now the work is distributed like this:

Name	Responsibility
Naida Nožić	Retrieval pipeline creation, fine tuning
Petra Buršić	Retrieval pipeline creation, fine tuning
Faruk Šahat	Test set creation, evaluation
Maksim Madžar	GPT integration, user query expansion