FACETS: A TOOL FOR MANAGEMENT AND NAVIGATION OF SYMBOLIC MUSIC COLLECTIONS

Tiange Zhu
CEDRIC Laboratory, CNAM Paris
tiange.zhu@lecnam.net

Raphaël Fournier-S'niehotta
CEDRIC Laboratory, CNAM Paris
fournier@cnam.fr

ABSTRACT

This paper presents FACETS, a tool that helps to manage and navigate large digital music libraries, aiming to aid the work of musicologists, composers, MIR researchers and the interested public. It supports queries based on melody, rhythm and metadata, as well as management of symbolic music datasets in MusicXML, MEI, Humdrum and ABC formats. It provides fast collection index and search functions based on Elasticsearch [1], with ranking methods by relevancy and music similarity, and an easy-accessible GUI with score visualization. FACETS is distributed as a Docker Image, with source code available on Github.

1. INTRODUCTION

With a rising number of online music libraries published in the past decades, there is an increasing demand of tools to better organize, navigate and query information in these libraries. Most of the search tools for music libraries are metadata-based, including the ones embedded in music streaming services like *Spotify* [2] and online music databases like *Discogs* [3], which locates music documents by descriptive information such as title, composer, genre. In such cases, queries can be addressed by text-based retrieval methods. Alternatively, some allow search by content in music compositions like melody and rhythm, which involves the extraction and processing of music information. The majority of existing content-based music search tools focused on automatic detection and identification of audio music [4] [5], while only a few dedicated to content-based search in symbolic music datasets, such as PeachNote [6] and Musipedia [7].

FACETS is a tool that helps to manage and navigate symbolic music libraries of different styles and formats. It supports queries by melody, rhythm and metadata. Especially, it provides ranking based on music similarity, and score visualization with highlighted matching results. It relies on Elasticsearch to index and search documents, guarantees a fast response time for query in large collections.

This paper mainly explains the functionalities of FACETS. Details of its design, indexing structures, feature extraction and ranking algorithms are elaborated in our previous work [8].

2. OVERVIEW OF FACETS

2.1 Software Components

FACETS is a software implemented in Python Django framework, which uses Elasticsearch as a back-end, accessible by both REST API and GUI.

2.2 Data Management

The data management module allows users to upload and manage digital music scores of their choice. Index creation, music data upload, and metadata upload are supported in this module.

Users may create new indexes, in order to help categorize documents for different search purposes. Created indexes are stored on Elasticsearch server. After index creation, users may upload digital scores in MusicXML, MEI, Humdrum and ABC formats, or a ZIP file including documents in these formats.

FACETS processes and indexes these documents. Basic descriptive metadata, as well as chromatic interval features, diatonic interval features and rhythmic features are extracted from them, then the features are converted into text-based encoded N-grams, and stored on Elasticsearch.

2.3 Dashboard

The stored data may be explored. The overview of indexes, and list of music documents stored in each index with basic descriptions can be accessed in both GUI and REST API. The GUI exclusively provides visualization of digital score documents using Verovio library.

2.4 Search

2.4.1 Search Input

Both metadata and content of the indexed music documents can be searched through GUI, as shown in Figure 1. An index must be chosen from the list of existing indices to specify the search range.

Users may enter text keywords to find documents by metadata. If search by content is desired, the query may be entered through an embedded piano keyboard, given a set

Attribution: T. Zhu, R. Fournier-S'niehotta, "FACETS: a tool for management and navigation of symbolic music collections", in *Extended Abstracts for the Late-Breaking Demo Session of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

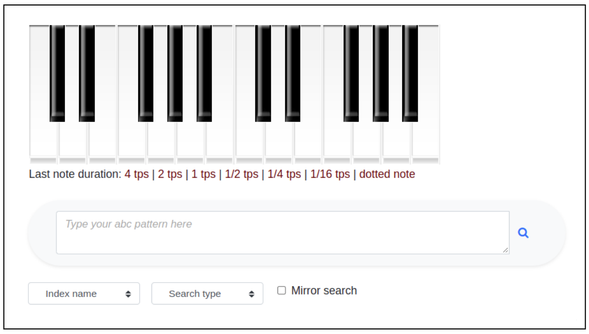

Last note duration: 4 tps | 2 tps | 1 tps | 1/2 tps | 1/4 tps | 1/16 tps | dotted note
Type your abc pattern here
Index name
Search type
□ Mirror search

Figure 1. The search input GUI in FACETS.

of musical notes with pitch and duration information, or in ABC notation.

Four search types are available in FACETS: chromatic, diatonic, rhythmic and text.

Users may search melody using chromatic or diatonic search. When a query is submitted in chromatic search type, a sequence of chromatic interval values between consecutive pitches is extracted from the query, and accordingly, the system finds scores in the selected index which contains one or more fragments that shares the same chromatic intervals. Each of these fragments are presented as a matching pattern in the search result. Similarly, if diatonic search type is chosen, a sequence of diatonic intervals would be extracted from the query for locating its matching patterns.

A mirror search option is available for chromatic and diatonic search, which finds matching patterns of same pitch intervals in both original and the opposite direction. Example and details can be found in [8].

When a query is entered in rhythmic search type, a sequence of ratios between consecutive notes with distinctive pitches would be extracted from the query. The system attempts to find matching patterns in the selected index based on such sequence.

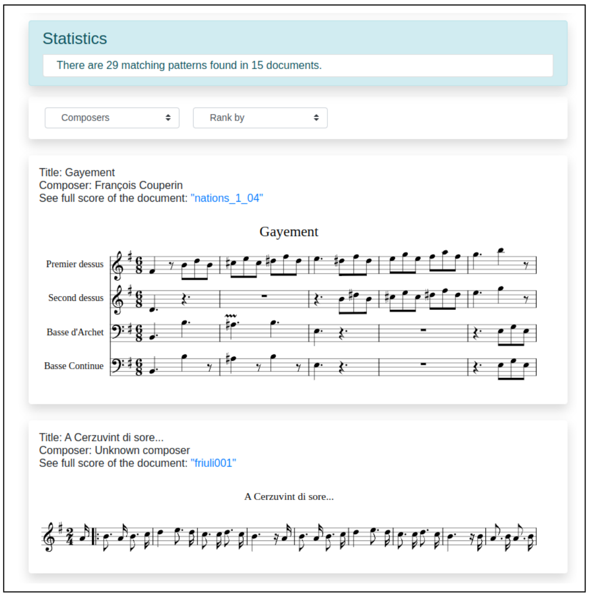
Text search locates music documents by querying keywords in their metadata.

2.4.2 Search Result

Figure 2 shows the search result page in GUI. Statistics displays the number of matching patterns and documents. The list of documents may be filtered by composer, while more aspects such as genre and time period may be used to filter the search result in the near future. Users may also choose ranking methods. The search result page gives previews of matching documents, with synthetic information and links to explore full scores with highlighted matching patterns.

2.4.3 Ranking

Users may choose to rank search results by relevancy or similarity. Rank by relevancy refers to ranking of matching documents based on occurrences of matching keywords or patterns, thus the document with highest numbers of occurrences may appear on the top in search results. Ranking by similarity is specially made for content-based queries, which measures music similarity between the query and each matching patterns, so the document that contains the most similar matching pattern may appear on the top. Edit distance [9] is adopted to measure similarity. Details of ranking algorithms are presented in [8].


Statistics

There are 29 matching patterns found in 15 documents.

Title: Gayement
Composer: François Couperin
See full score of the document: "nations_1_04"

Title: A Cerzuvint di sore...
Composer: Unknown composer
See full score of the document: "friuli001"

Figure 2. An example of search result page, corresponds to a query.

2.5 Distribution

FACETS is distributed as a Docker image [10]. A demo is available on Huma-num [1] . The source code is available on Github [2] .

3. PERSPECTIVES

FACETS provides fast information retrieval service for digital music resources, aiming to help the work of musicologists, composers, MIR researchers and the interested public.

In the near future, endeavours would be made to improve the faceted navigation [11] in search result GUI. We also envision an advanced version of data visualization with clustering and knowledge graphs, in order to provide potential links between music compositions based on content. Furthermore, the list of search features may be upgraded in the long term, towards more comprehensive capability in music search.

4. ACKNOWLEDGEMENTS

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004746 (Polifonia: a digital harmoniser for musical heritage knowledge).

[1] `http://neuma-dev.huma-num.fr/`

[2] `https://github.com/polifonia-project/facets-search-engine`

5. REFERENCES

[1] ElasticSearch, "The elastic search engine," accessed: 2021-12.

[2] B. Carterette, R. Jones, G. F. Jones, M. Eskevich, S. Reddy, A. Clifton, Y. Yu, J. Karlgren, and I. Soboroff, "Podcast metadata and content: Episode relevance and attractiveness in ad hoc search," *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul 2021. [Online]. Available: http://dx.doi.org/10.1145/3404835.3463101

[3] "Discogs: Music database and marketplace," accessed: 29.09.2021.

[4] S. Fenet, G. Richard, and y. Grenier, "A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting," in *ISMIR*, Miami, United States, Oct. 2011, pp. 121–126. [Online]. Available: https://hal-imt.archives-ouvertes.fr/hal-00657657

[5] A. Kotsifakos, P. Papapetrou, J. Hollmén, D. Gunopulos, and V. Athitsos, "A survey of query-by-humming similarity methods," in *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2413097.2413104

[6] V. Viro, "Peachnote: Music score search and analysis platform." in *ISMIR*, 2011, pp. 359–362.

[7] "Musipedia," http://www.musipedia.org.

[8] T. Zhu, R. Fournier-S'niehotta, P. Rigaux, and N. Travers, "A framework for content-based search in large music collections," *Big Data and Cognitive Computing*, vol. 6, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2504-2289/6/1/23

[9] E. Ristad and P. Yianilos, "Learning string-edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[10] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.

[11] O. Ben Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev, "Beyond basic faceted search," 01 2008, pp. 33–44.