FACETS: A Tool for Improved Exploration of Large Symbolic Music Collections

Tiange Zhu**
CNAM
Paris, France
tiange.zhu@lecnam.net

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

Philippe Rigaux
CNAM
Paris, France
philippe.rigaux@cnam.fr

ABSTRACT

Large collections of symbolic music documents need efficient information retrieval tools. We introduce FACETS, a versatile tool for exploring and management of such collections. FACETS is a scalable and flexible content-based search engine, offering melodic and rhythmic querying modes. For improved navigation, a faceted interface orders the results, to reduce information overloading, and it may be used as a primary entry point in the tool. FACETS is available as a standalone Docker image and Github repository, aiming to help musicologists, composers, MIR researchers and the interested public.

KEYWORDS

music documents, digital libraries, faceted search, information retrieval

ACM Reference Format:
Tiange Zhu, Raphaël Fournier-S'niehotta, and Philippe Rigaux. 2024. FACETS: A Tool for Improved Exploration of Large Symbolic Music Collections. In *11th International Conference on Digital Libraries for Musicology (DLfM 2024), June 27, 2024, Stellenbosch, South Africa.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3660570.3660578

1 INTRODUCTION

The abundance of documents, or *information overload*, calls for efficient knowledge organisation systems. Search engines have been developed to obtain an ordered list of results relevant for a query, which may have several forms (text-based or not, structured or not, etc.). Along that, classifications are developed to organize items in a collection according to their features, helping humans to find what they need. Ranganathan's work on classification [21] introduced *facets*, to go beyond standard taxonomies. Faceted classification relies on semantic categories which are combined to create full classification entries (e.g., a music piece "composed by Beethoven", "between 1820 and 1830", "a symphony"). In a search engine, they are implemented as filters, to help users refine their search and find more quickly what they are looking for.

[*]Corresponding author. All authors contributed equally to this research.

DLfM 2024, June 27, 2024, Stellenbosch, South Africa
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1720-8/24/06...$15.00
https://doi.org/10.1145/3660570.3660578

Fostered by the Web and the creation of digitization techniques, large collections of music documents have emerged in the recent decades, giving rise to a pressing need for dedicated retrieval tools. Those large collections usually span a broad spectrum of formats, genres, and cultural contexts. Therefore, a search tool for these collections should be capable of accommodating and navigating the high diversity of music documents. Such a tool also faces classic retrieval challenges: it must present users with relevant documents without overwhelming them with excessive information.

FACETS aims to serve as a versatile tool for exploration and management of digital music libraries. It is compatible with music notations encoded in MusicXML [10], MEI [22], Humdrum [11], and ABC [1] formats. More than 10,000 scores from NEUMA [2] are integrated in the system, and FACETS is designed to easily integrate external digital music resources.

FACETS offers several modes of navigation in the collections, through content-based or metadata search, equipped with a faceting engine for refining the results. Two categories of users are envisioned for FACETS: *collection managers*, who would import their music documents into the system, and their *public*, who would browse the presented collections. Collection managers could belong to cultural heritage institutions and their public be the general public; or both collection managers and public could be musicologists in an academic setting. The back-end and the front-end are decoupled, so that collection managers may use the REST API and develop a custom graphical user interface (GUI) if necessary. A default GUI is provided for immediate adoption. The development is open-source, so that the effort may be improved by the community.

2 RELATED WORK

Collections of music documents have been gathered in corpora for decades, before recording or digitization became even possible. Preserving music information as sheet music has a rich and complex history. Dedicated libraries have been created, and in the past decades, they have been digitally encoded in formats such as MusicXML [10] or MEI [22], enabling possibilities for automatic music analysis and retrieval. The RISM project [5], Neuma [2], and the International Music Score Library Project (IMSLP, [12]) are examples of such endeavours from the academia. Commercial offerings of digital sheet music also exist, namely MuseScore [17], nkoda [19], Henle library App [24] or enote [7].

Most search engines offer search functionalities by metadata such as title, artist, and genre. While widely adopted by streaming services and digital libraries, they may not meet the needs of users who are trying to find a tune by information in its content like melody and rhythm. A number of search engines have been developed to address this problem. Musipedia [18] allows melodic and rhythmic

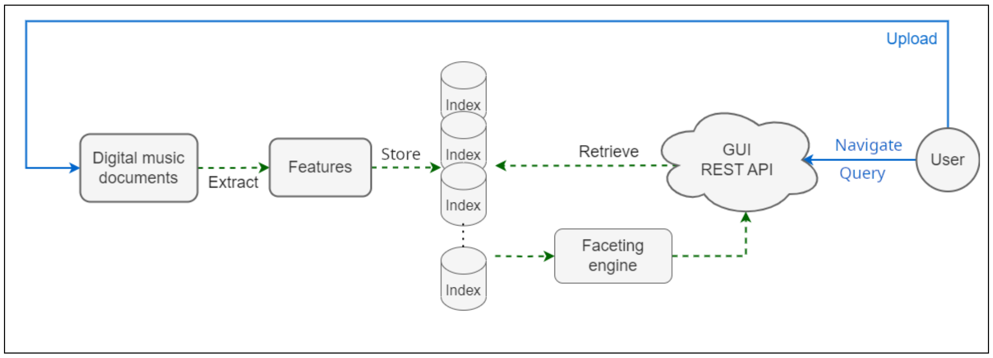


Figure 1: The overall design of FACETS**. Blue lines indicate user behavior, and the green lines represent the workflow in the** FACETS **system, reacting to user requests.**

search in various music datasets including the Essen Folk Song Collection. Themefinder [13] supports retrieval of patterns (e.g., motifs, themes) based on regular expression, allows search by melodic features such as pitch interval, scale degree, or gross contour. Music ngram viewer [25] provides n-gram based melodic pattern search for Petrucci Music Library [12], using Optical Music Recognition [3] to convert printed sheet music into machine-readable formats, to enable search in scanned music scores. The SIMSSA project [9] also enabled retrieval of sheet music images, with a strong focus on early music. Dig that lick project [8] offers a regex-based tool for the search of melodic patterns, helping to study jazz music from the Weimar Jazz Database.

Sacco, Tzitzikas, et al. [23] provide detailed theoretical principles and practical uses of faceted search, and [27] present an academic survey of existing facet-ranking approaches and the key challenges they face. In [15], authors discuss at length the different facets that can help organise music documents with metadata, but they do not address content-centric features (e.g., melodic). Some above mentioned search engines have applied a faceting approach, such as [8, 9]. [8] provides options to refine search by the length of instances and performers. In SIMSSA, users may narrow down results by year, genre, instrument and file formats.

While these tools offer powerful search options, we have spotted that they are typically limited to search within predefined databases, and there is a room for improvement on faceted search and efficiency (some regular expression-based tools report to take a few minutes to run a query). Hence, we present the FACETS tool, to help content-based symbolic music search in an easy accessible and user friendly manner, with improved scalability and flexibility.

3 THE FACETS SYSTEM

3.1 Overview

FACETS has two types of users: the *primary users*, and the *end users*. Primary users are digital library managers, who have documents they want to present to end-users. Whether they already have a platform displaying their documents or not, they may either set up a standalone FACETS on a web server (with a Docker image), or couple FACETS to an existing collection (through API calls, or by reusing the open-source Python project). End users are those who will interact with FACETS to navigate a collection, e.g., through the provided GUI. A musicologist could be both a primary user and an end user, setting up FACETS for their own use, to analyse and efficiently their research corpus of music documents. The overall design of FACETS is illustrated in Figure 1. Primary users would be the ones *uploading* the digital music documents. End users mainly *navigate* and *query* the potentially large collection of music documents. The users may choose the suitable interface to access FACETS, through GUI or REST API.

The core of FACETS consists a modular representation component, which is capable of extracting features from the musical content of the documents. It is based on the work presented in [28]. Melodic and rhythmic features are extracted from documents as sequences, and encoded in a n-gram format. For polyphonic music, such features are extracted from each voice. The encoded features and metadata are both stored in (text-based) indices, akin to traditional search engine practices. Given a query, FACETS retrieves the matching documents, analyzes the retrieved collection, and organizes them as facets to improve the navigation.

3.2 Music Features

A music feature describes a certain property of a composition. Extensive research in music features has been presented due to its importance in many MIR tasks, such as transcription or music classification. Tools like jSymbolic [16] and Music21 [4] are available for extraction of such features (e.g., pitch intervals, IOI) from symbolic music. We curate a list of essential features for the faceting engine that facilitate the categorization of most music databases, and avoid overloading the user with *too many facets*.

For selecting our standard facets, we prioritize high-level features that provide fundamental information about the music documents, that are easy to understand. For instance, "composer" is such a feature, as all music documents are inherently composed by a composer, even if the composer's identity may be unknown. Then, we consider efficiency, to help the system scale. If extracting a feature is excessively time-consuming, it may significantly impact the performance of the system. Accordingly, we select a list of features that are classified into metadata and content-based categories.

Metadata features incorporated in FACETS are:

- Composer: the name of the composer of a music composition;

- Period: the era that the composer of a composition lives in, within a span of one or two centuries;
- Instruments: a list of instruments present in a piece.

The selected content-based features are as follows:

- Chromatic intervals: a sequence of chromatic interval values between consecutive pitches;
- Diatonic intervals: a sequence of diatonic interval values between consecutive pitches;
- Rhythmic intervals: a sequence of ratios between consecutive notes of distinctive pitches;
- Key: key mode and key tonic name of a piece of music;
- Time signature: time signature indicated in music notation, which suggests the rhythmic structure and meter of a composition;
- Number of voices: number of distinctive instrumental or vocal lines or sections present in the composition;
- Number of measures: total number of measures indicated in a music score.

The system process the content-based features by encoding them in n-grams, and if the metadata of a file is incomplete, FACETS queries Wikidata [26] to augment its knowledge base with available information. For example, provided the title of a piece, the system may retrieve the composer and period information. The processed features of each document are stored on Elasticsearch servers in its assigned indices for fast retrieval with faceted navigation.

3.3 Facets

A facet is a dimension along which the documents of a collection may be analyzed, and from which several values can be extracted. For a given value of a facet, several documents may exhibit such a value. We call *pre-computed* facets the list of facets that can be computed *offline*, *i.e.*, before the user enters the system. All features presented in the previous section are listed as pre-computed facets, as shown in Figure 2.

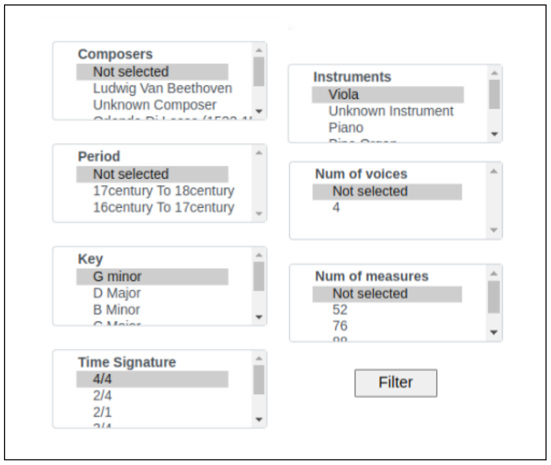


Figure 2: Pre-computed facets for navigating a collection of music documents.

We also define *dynamic facets*, which are computed at runtime, after a query is made. Dynamic facets provide recommendations

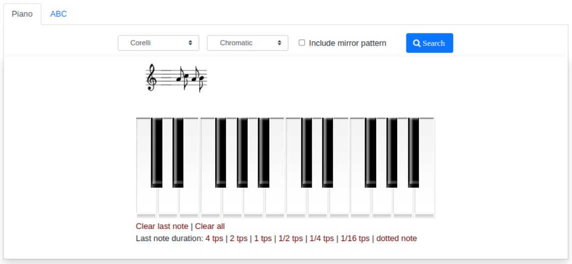

Piano
ABC
Corelli
Chromatic
Include mirror pattern
Search
Clear last note | Clear all
Last note duration: 4 tps | 2 tps | 1 tps | 1/2 tps | 1/4 tps | 1/16 tps | dotted note

Figure 3: Search page with a piano keyboard, for submitting query of a melodic/rhythmic pattern. In this example, we chose a "chromatic" query type to search in an index named "Correlli" which contains a collection of scores composed by Arcangelo Corelli.

of facets that are most likely interest the user based on its query, particularly in the *discovery* mode. For example, user may query a dataset by key "A minor", FACETS computes the recommended facets by statistics: it may discover that most of the retrieved pieces are composed by "Johann Sebastian Bach" including instrumentation "Pipe Organ". Accordingly, it would return a list of dynamic facets: "Johann Sebastian Bach", "Pipe Organ".By leveraging dynamic facets, users can refine their search results to locate pieces of interest more effectively. These facets can also help to uncover relationship of influence among groups of composers, or develop curiosity towards previously unfamiliar music genres or styles.

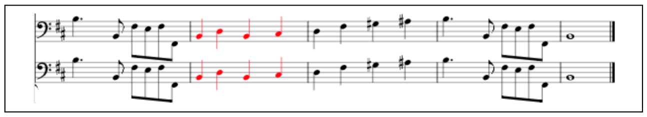
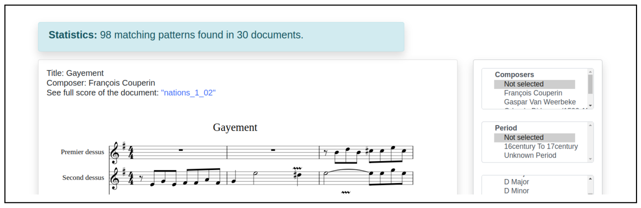


Figure 4: Example of an excerpt of the Violone and Organ parts of Arcangelo Corelli's Trio Sonatas, with segments highlighted, appear as results corresponding to the query example in Figure 3.



Statistics: 98 matching patterns found in 30 documents.

Title: Gayement
Composer: François Couperin
See full score of the document: "nations_1_02"

Figure 5: Search result page, where the same pattern is queried in a different collection including works of various composers. Facets module is positioned to the right in the figure.

3.4 User Interface

The user interface has two components: a Graphical User Interface (GUI) and a REST API. The GUI is a key contribution for improving

Question	Median score	Average score
How easy was it to use the tool?	4	3.66
Were the search options clear and intuitive?	3	3.40
Did the tool accurately identify the pieces or patterns that you look for?	5	4.56
Did you find the ranking convincing?	4	4.0
Did you find the faceted search helpful in refining your search?	4	4.31
How was the response time during searches?	5	4.31
How satisfied are you with the overall design?	4	3.87

Table 1: User satisfaction survey results.

the navigation in large digitized collections of scores. The GUI proposes two modes of navigation, according to the needs:

- a *standard* search mode, in which the user may query by a melodic or rhythmic pattern;
- a *discovery* mode, where the user navigate the collection by our listed facets instead of patterns.

We show a pattern search example in Figure 3, where a pattern has been queried as chromatic intervals within an index named "Correlli" containing a collection of scores composed by Arcangelo Corelli. Figure 4 shows a matching example, highlighting the retrieved patterns. Figure 5 displays the search results of the same query for another collection, which consists works of various composers including François Couperin and Gaspar van Weerbeke.

3.5 Implementation

FACETS has been implemented with the Python Django framework, coupled with Music21 [4] and jSymbolic [16] for feature extraction. FACETS accommodates symbolic music collections in various formats including MEI, MusicXML, Humdrum, and ABC. The backend relies on ElasticSearch [6], which handles extracted features' storage in indices, and document retrieval through faceting. We use Verovio library [20] for data visualization in standard Western music notation format. FACETS also offers a GUI for collection management, where users can upload music collections in batches via ZIP archives, with additional option to upload metadata file in JSON format. A dashboard is available for displaying the list of documents under each index. The code for our implementation is available on Github[1], as well as Docker instructions.

4 EVALUATION

Generally, an IR system should be evaluated in various dimensions such as relevance, speed, user satisfaction, usability, efficiency and reliability [14]. Previously [28], we have performed a quantitative evaluation of FACETS on its speed. The average execution time of FACETS is around 1 ms to process a single query, which is more than 200 times faster compared to regular expressions (adopted in systems like [13, 8]). This result shows that FACETS is efficient to handle large collections and rapidly retrieve documents with complex queries.

For evaluating other aspects, we created a survey consisting of seven questions regarding user utility, search usability, retrieval effectiveness and the overall design quality. Sixteen participants has taken part in the survey. Despite the number of participants is limited, the distribution in their cultural and musical background remains diverse. Nine of participants are music amateurs who have played an instrument or learned music theories, five are professional musicians, only two have no previous training in music. Six participants have used content-based music search engines, two of which are domain experts who built such applications, while ten participants have never tried to use one upon survey.

The participants were given a short guide, and required to spend a minimum of five minutes experimenting with FACETS prior to completing the survey. They indicated their satisfaction on a Lickert scale, from 1 to 5, and some have given additional comments after taking the survey. Table 1 presents the median and average scores from the responses of each question. The survey results indicate a generally satisfactory outcome. In particular, aspects like accuracy, response time, and faceted search gained positive feedback. However, FACETS needs improvement in functionality, accessibility, and overall design. From the comments we have received, some participants called for more query examples and comprehensive guidance. Additionally, one participant suggested to include a MIDI player to improve accessibility for those lacking musical expertise.

5 CONCLUSION

We present the FACETS tool for content-based exploration in large collections of music scores with improved scalability and flexibility. The tool provides an easy-accessible interface, with a faceted navigation module to help refine queries by content-based and metadata features. A working implementation of the tool has been made publicly available. It aims to help users from various backgrounds, including musicologists, composers, research scientists, and the interested public: composers may find inspiration using FACETS to search and browse music scores, musicologists may discover useful information for comparative musicological analysis through faceted navigation. A librarian may find FACETS particularly helpful for the management of symbolic music collections from various sources. In the future, we intend to improve the accessibility of the tool, and incorporate music patterns as facets to further expand the functionality.

ACKNOWLEDGMENTS

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]See: https://github.com/polifonia-project/facets-search-engine.

REFERENCES

[1] 2024. ABC Notation. https://abcnotation.com/. Last accessed Jan. 2024. (2024).

[2] L. Abrouk et al. 2009. The NEUMA Project: Towards On-line Music Score Libraries. In *Intl. Workshop on Exploring Musical Information Spaces (WEMIS'09).*

[3] Jorge Calvo-Zaragoza, Jan Hajič Jr, and Alexander Pacha. 2020. Understanding optical music recognition. *ACM Computing Surveys (CSUR)*, 53, 4, 1–35.

[4] Michael Scott Cuthbert and Christopher Ariza. 2010. Music21: a toolkit for computer-aided musicology and symbolic music data.

[5] Répertoire International des Sources Musicales (RISM). 1952. Base de données du répertoire international des sources musicales (rism). Date de consultation : 15 Novembre 2023. (1952). http://rism.info/.

[6] ElasticSearch. 2023. The elastic search engine. https://www.elastic.co/. (June 2023).

[7] enote GmbH. 2024. Enote. (2024). https://enote.com.

[8] Klaus Frieler, Frank Höger, Martin Pfleiderer, and Simon Dixon. 2018. Two web applications for exploring melodic patterns in jazz solos. In *International Society for Music Information Retrieval Conference.* (July 2018).

[9] Ichiro Fujinaga, Andrew Hankinson, and Julie E. Cumming. 2014. Introduction to simssa (single interface for music score searching and analysis). In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology* (DLfM '14). Association for Computing Machinery, London, United Kingdom, 1–3. ISBN: 9781450330022. DOI: 10.1145/2660168.2660184.

[10] 2001. *"musicxml for notation and analysis". The Virtual Score: Representation, Retrieval, Restoration.* W. B. Hewlett and E. Selfridge-Field, MIT Press, 113–124.

[11] 1997. Vol. 1. The MIT Press. Chap. Humdrum and Kern: Selective Feature Encoding.

[12] 2023. International Music Score Library Project. https://imslp.org/. (June 2023).

[13] Andreas Kornstaedt. 1998. Themefinder: a web-based melodic search tool. *Computing in Musicology*, 11, 231–236.

[14] Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. 2020. Pirhdy: learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music. In *Proceedings of the 28th ACM international conference on multimedia*, 574–582.

[15] Devika P Madalli, B Preedip Balaji, and Amit Kumar Sarangi. 2015. Faceted ontological representation for a music domain. *KO KNOWLEDGE ORGANIZATION*, 42, 1, 9–25.

[16] Cory McKay and Ichiro Fujinaga. 2006. jSymbolic: A feature extractor for MIDI file. In *ISMIR*, 302–305.

[17] [n. d.] MuseScore. Accessed: 01.12.2023.

[18] 2023. Musipedia. http://www.musipedia.org. (June 2023).

[19] nkoda limited. 2024. Nkoda. (2024). https://www.nkoda.com.

[20] Laurent Pugin, Rodolfo Zitellini, and Perry Roland. 2014. Verovio: a library for engraving mei music notation into svg. In *ISMIR*, 107–112. http://verovio.org.

[21] Shiyali Ramamrita Ranganathan. 1967. *Prolegomena to library classification.* Asia Publishing House (New York).

[22] Perry Rolland. 2002. The Music Encoding Initiative (MEI). In *Proceedings of the International Conference on Musical Applications Using XML*, 55–59.

[23] Giovanni Maria Sacco, Yannis Tzitzikas, et al. 2009. *Dynamic taxonomies and faceted search: theory, practice, and experience.* Vol. 25. Springer.

[24] G. Henle Verlag. 2024. The henle library. (2024). https://www.henle-library.com/.

[25] Vladimir Viro. 2011. Peachnote: music score search and analysis platform. In *ISMIR*, 359–362.

[26] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57, 10, (Sept. 2014), 78–85. DOI: 10.1145/2629489.

[27] Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Xiaoyu Fu, and Boqin Feng. 2013. A survey of faceted search. *Journal of Web engineering*, 041–064.

[28] Tiange Zhu, Raphaël Fournier-S'niehotta, Philippe Rigaux, and Nicolas Travers. 2022. A framework for content-based search in large music collections. *Big Data and Cognitive Computing*, 6, 1. DOI: 10.3390/bdcc6010023.