QueryShare: Working Together to Facilitate Exploratory Multimedia Searches without Skill in Creating

Masahiro Hamasaki
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki, Japan
masahiro.hamasaki@aist.go.jp

Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki, Japan
m.goto@aist.go.jp

ABSTRACT

This paper describes a music exploratory search interface called *QueryShare*, which provides query searching and recommendation functions for query sharing among users. Most people are not expert users who know how to use various music metadata that include automatically estimated musical features to represent their own information needs as a query. Therefore, it is difficult for them to enter a complex query for music content retrieval. The original feature of our proposed interface is to make users share every query as a public web page. This feature enables users to use search queries, find recommended queries, and revise existing queries. Beginners can use an applicable query, which is more complicated than they might create on their own. Experts can readily reuse a query (web page) of their own making. The interface assists users in finding results for interesting queries and in performing music exploratory search without skills to create complex queries. We developed a prototype system as a web application for music videos on the most popular Japanese video sharing service. Users can search for over 360,000 music videos using our system. Results of a preliminary user study demonstrated that users found the query creation interesting and that they were interested in seeing and using queries created by other users, although some users hesitated to share their queries.

ACM Classification Keywords

H.3.3 Information Search and Retrieval: Query formulation; H.3.5 Online Information Services: Web-based services

Author Keywords

Search interface; Query sharing; Multimedia Content Retrieval; Collective intelligence; User-generated content;

INTRODUCTION

A "query" is an intellectual resource among many on the vast web of multimedia content. This study explores a method used to perform exploratory multimedia searches using open collaboration with query sharing. It is difficult for most people to generate applicable queries for multimedia content retrieval because they have no particular knowledge about these retrieval systems. Many do not even know how to create applicable but complex queries using various metadata and low-level features extracted from multimedia contents. With the growing phenomenon of User Generated Content (UGC), we have become surrounded by unfamiliar but potentially interesting creators and music content (e.g., music videos on video sharing services). To discover these hidden musical talents, it is important to search for music content actively. However, creating queries for such active search is not easy for most people because most users have no particular knowledge about music retrieval systems: they might be unaware of

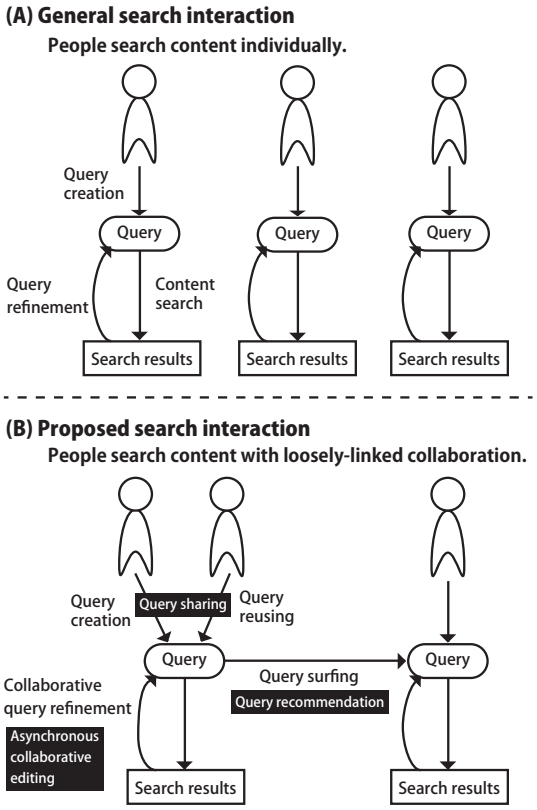

Figure 1. General search interaction compared to proposed search interaction.

OpenSym '17 August 23–5, 2017, Galway, Ireland

ACM ISBN 978-1-4503-5187-4/17/08...$15.00
DOI: https://doi.org/10.1145/3125433.3125447

various metadata or automatically estimated musical features to which systems have access. We propose that sharing user-generated queries can resolve many of the diverse difficulties that people confront.

Many existing technologies can assist users in finding music content without demanding much effort on the part of the user to generate a complex query. For example, some studies use humming [11, 42] or a fragment of music content [26, 5] as a user query. Query-by-Example [40, 21] also uses music content as a query. Music recommendation [35, 39, 28] predicts users' favorable music content based on user profiles or user's listening history. Music visualization with an interactive interface [34, 14] helps users to discover music content. However, no such method assists users in generating queries using rich metadata. Moreover, user needs are often more complex than "I want music I would like" or "I want to track down this specific song."

Figure 1(a) depicts a general search interaction. Into a search interface, a user submits a query that represents the user's own information needs. The search engine outputs search results including content related to the query. Users who are not satisfied with the results or who have no clear goal must modify queries and search again (exploratory search [30]). Query refinement is just as important as query creation; most users do it [22].

We propose a new search interface that enables users to create and share queries, and to explore and reuse the queries of others: *QueryShare*,. The system allows beginners to find an applicable query that is more complicated than one they would be able to create on their own. Similarly, experts can find useful queries created by other experts. Moreover, they can refine queries collaboratively (Fig. 1b). QueryShare provides a new search interaction, which we designate as *Query Surfing*: browsing queries through query searching and query recommendation.

We developed a prototype system with the QueryShare interface as a web application for music videos on *Niconico*, the most popular Japanese video sharing service. Users can search through more than 360,000 music videos and share user-generated queries using our prototype system. Query recommendation is implemented with a query-similarity-based approach and a content-similarity-based approach.

We also conducted a user study to gain user feedback. The participants accepted query sharing and strove to create more queries despite the use of many search parameters. Results showed that QueryShare helped users to create queries and to perform exploratory searching for music.

This paper is organized as follows. Section 2 reviews existing works. Section 3 explains the concept of QueryShare and our proposed method of search interaction. Section 4 presents a description of implementation of the prototype system with the proposed method. Section 5 presents an explanation of a preliminary user study conducted using the prototype system. Section 6 discusses the importance of sharing queries explicitly. Section 7 presents a summary of this paper's contributions.

RELATED WORK

In this section, we discuss existing work related to the assistance of query creation and query refinement. As described in the previous section, query refinement is a crucial part of the search process. Query suggestion supports query refinement by recommending additional queries or terms. For instance, in the case of a text query, an assistance system finds spelling errors [7, 29], complementary queries [4, 44], or substitute queries [23, 10, 27, 1], or suggests expanded queries [8]. Users can create more complex queries using these suggestions, but most users use them for the replacement of applicable queries.

Some systems support database exploration by recommending complex queries, such as an SQL query [6, 12, 2]. Some such systems are designed to help non-expert users of target databases. Therefore, the purpose of such systems is similar to ours. However, they specifically examine prediction of the intent of users and find acceptable queries from query logs including queries that other users have created. In contrast, we examine sharing of queries among users. Our system treats queries as valuable web content. Zhang reported results of a pilot study of query sharing [45]. Their results indicate that query sharing is valuable in difficult search tasks.

There are some researches related to collaboration in information retrieval [41, 32, 13]. SearchTogeter is an interface for collaborative web search and it aims to assist a search in tandem [33]. It mainly assists communications among users in search processes. It is not designed to help users to create complex queries directly. Additionally, users should have a partner who shares a common goal of searching before starting collaborative search.

Faceted navigation can guide a user to input query parameters step-by-step [3, 31, 24]. It is helpful for general users; it reduces burdens not only on general users but also on experts. However, it does not help users to use unknown facets and their parameters. The proposed system can recommend a user such queries including unknown facets for the user (see Section 4.4 and Figure 6). A user can learn a meaning of a facet through these queries. This function is similar to Query-by-Example (QbE), but our approach is useful for obtaining search results and for learning query parameters.

As explained above, many existing works are not designed to assist users in creating complex queries because the main purpose of information retrieval is to find facts or to gather information. Its requirements are to find targeted items or some good items with short queries, and to burden the user as little as possible. Regarded from this perspective, important issues include personalization, context-awareness, and privacy. No importance is assigned to the creation of complex queries and their reuse. However, from the perspective of exploratory multimedia searches, it is useful to share and reuse various complex queries because they can facilitate retrieval of diverse multimedia content. Kamalzadeh reports that 50% of active music listeners would like to choose songs one after another [25]. Furthermore, just 9% use online recommendations; 10% use shuffle when listening to a collection. These results indicate that active music listeners enjoy listening to

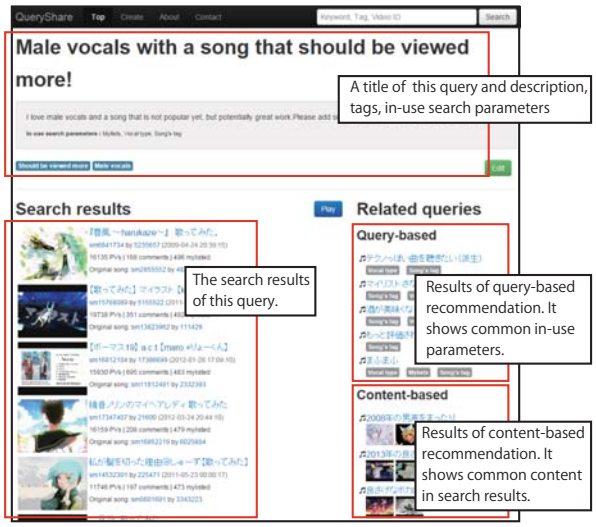

Figure 2. Screenshot of the "Querypage" in QueryShare.

songs and that they enjoy choosing songs. Regarded from this perspective, creating complex queries and reusing them are important compared to existing information gathering. They can provide new ways of choosing songs for active music listeners.

QUERYSHARE

We introduce three key features of QueryShare.

1. Contentification: turning queries into content that can be shared (web pages)
2. Query Surfing: browsing queries using query searching and recommendation
3. Asynchronous collaborative editing for query refinement

Contentification

QueryShare transforms each user-generated query into a public web page called a *querypage* to facilitate the sharing of queries among users. This transformation from machine-readable to human-readable content is designated as *contentification*. Figure 2 portrays a querypage in the prototype system. Each querypage includes the query's search parameters as well as its metadata and search results. The creator who makes the query describes the metadata. The search results from the query are returned from a search engine. All of these data help other users to discover and understand the query. If the creator would like to reuse their query, then they merely revisit the web page.

Query Surfing

:

Users without the skill necessary to create a suitable query must discover other queries that meet their needs and which are registered in the system. QueryShare provides two methods to find queries: query searching and query recommendation. Users can search queries using keywords, tags, or content titles. Additionally, each querypage shows related queries as results of query recommendation. Using these links, users can explore the network of related queries. We designate such user activity as *query surfing*.

Asynchronous collaborative editing for query refinement

A querypage can be edited by users at any time. Users who would like to update or improve the query merely access a querypage and click the "Edit" button. The system enables users to edit all data in the query. The original creator of the query and other users can edit the edit parameters collaboratively, as they can with a wiki or Wikipedia. Through such continual and collaborative editing, shared queries can be refined.

The original creator of a querypage can set a password for editing if the user would not like to have their own query edited by others. They can create a branch of the query if other users would like to edit such locked queries. It is a copy of the source query. A user who creates this branch can also set a password if necessary. This is another style of collaborative editing.

Proposed search interaction

We assume that QueryShare users are broadly divisible into two groups: experts who can refine complex queries with many search parameters, and general users who cannot easily do so. In this section, we introduce user behaviors for both groups in a new search interaction based on sharing queries.

Presuming that a user would like to listen to new songs, they might create a query "sort by date published." The search results are too numerous: they must be narrowed down. However, the user does not know how to form a query for such information using the available search parameters. To support such a user, the system recommends other queries based on the similarity between search parameters. One suggestion might include "sort by date published" and other well-chosen parameters (e.g., a lower limit of the play counts) to select the most popular from among numerous new songs. The user listens to songs in the search results and finds some favorites among them. If this querypage is exactly what the user wants, then the user can continue to use this.

Queries might also be recommended based on similarity among their search results. In the previous scenario, presuming that another recommended query uses a parameter that denotes the proportion of male-like or female-like voices, and that the user notices some of their favorite songs among the query's results. The user might conclude that the male–female voice parameter is important for representing their interests. The user can immediately make a new query to incorporate this additional parameter. As explained earlier, an imperfect query has become a starting point to find more relevant, suitable queries. Moreover, from the recommendations, users can learn to make better queries. These are incentives for general users.

Experts can create suitable queries without the system aids. Therefore, they might have less need to share the queries of others, but they can still benefit from doing so. Using complex queries developed by other experts saves them time and trouble. Furthermore, it is not always true that experts know everything about a subject; they can learn from other experts'

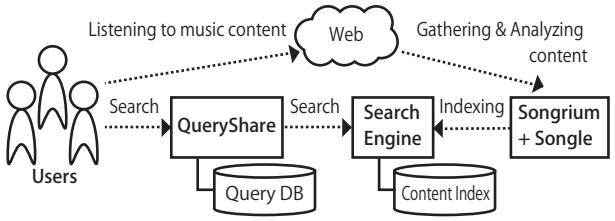

Figure 3. System architecture.

queries and use them as a basis to formulate new queries. These are incentives for experts. In addition, as demonstrated by the cases of open source software (OSS) and user generated content (UGC), using and reviewing queries by others motivates people to create better queries.

PROTOTYPE IMPLEMENTATION

We developed a prototype system as a web application with which users can search music videos and share queries. Figure 3 presents a system overview. QueryShare is a search interface, but it does not manage music videos. QueryShare uses them through the public web services: Songrium [1] [18] and Songle [2] [15]. Songrium gathers music videos and their metadata from which Songle can calculate musical features. In the next two subsections, we introduce the target dataset and its search parameters. Later, we explain how queries are created and recommended within the system.

Datasets

In the prototype system, users can search and watch music video clips uploaded to Niconico [3]. Particularly, "Singing a VOCALOID song" is a target content category of our prototype system; it is currently an extremely popular category of music-related user-generated content on Niconico. A VOCALOID song is an original song composed for the singing synthesizer named VOCALOID [19]. Over two thousand VOCALOID songs and ten thousand singing videos are uploaded every month [17]. It is a good and rich environment to enjoy music, but it is difficult to encounter unfamiliar but potentially interesting songs. As described above, this dataset is a good target for use with QueryShare.

In this prototype system, users can search 363,518 singing videos, which consist of 64,774 singers, 13,138 songs, and 3,514 composers. Such contents have metadata including page views and release date, which are openly available on Niconico. Our system also uses some content analysis results as metadata. Figure 4 shows that our target music content is described using the Music Ontology [37]; as might be readily apparent, the relation between the fields is complex. Green objects are classes. Orange squares are property values. In the prototype system, all of these values are used as search parameters to find a targeted content "Singing video." This picture depicts sources of various metadata. "Social tags" and "Release date" are metadata on Niconico. "Audio feature" and "Voice feature" are calculated from the sound signal. The former is compressed audio feature vectors using learned latent representations [20]. The latter are the estimated male and female singing voice characteristics [18].

[1] http://songrium.jp
[2] http://songle.jp
[3] http://www.nicovideo.jp

Table 1. Example of query: "Hit songs of jazz in 2011 with a nice, but not popular male-like voice". Each user should judge the query correctness. If users think these search parameters are incorrect, then they can update them or create a new query based on them.

Compatibility condition	Song's social tag = 'Jazz' or 'Fusion' Song's release date $\geq$ '2011-01-01' Song's release date $\leq$ '2011-12-31' Page views $<$ 10000, # of mylists $>$ 500 Voice feature $<$ 0.25
Sort criterion	Song's views (descending order)
Aggregation rule	the same song the same singer

Search parameters

The prototype system has search parameters of three categories. (1) *Compatibility Condition*: used to extract only contents that meet a specified criterion, e.g., "with a social tag of ballad" or "number of views is greater than 100,000." (2) *Sort Criterion*: used to order search results, e.g., "sort by date published" or "sort in ascending order of views." (3) *Aggregation Rule*: used to exclude content that partially duplicates other search results, e.g., "exclude same singer's videos" or "exclude same song videos."

Table 1 presents examples of search parameters in a query. With the prototype system, users can use metadata for compatibility conditions. For example, users can set an upper limit and a lower limit of page views as a compatibility condition. For sort criteria, users can use metadata of an interval scale, e.g., page views and release date. Aggregation rules support three conditions to exclude videos: the same song, the same singer, and the same composer.

In general web search methods, such as those presented by Google, Bing, and Yahoo, users must input a keyword query only. From the keyword query, the search engine extracts web pages and sorts them by mutual relevance. Furthermore, it aggregates web pages in the same web site. In fact, the web search engine extracts, sorts, and aggregates contents without explicit user requests. This search interaction is designed to reduce a user's burden. Our research is aimed at helping users to formulate various applicable queries and to share them among users. Then, QueryShare allows users to edit all search parameters in Figure 4. In section 6, we present discussion of the importance of using complex queries by people.

Query creation

As the previous example shows, QueryShare supports many search parameters. For that reason, it is difficult for beginners and even for experts to coordinate the numerous parameters necessary to identify relevant content. Therefore, QueryShare has two functions to support query editing.

(A) Query preview of each condition: A query consists of multiple conditions. Finding an appropriate balance can be difficult: if results are overly broad, then the search must be refined; but if the range of the conditions is too narrow, then the user receives zero results. In a complex search, a user

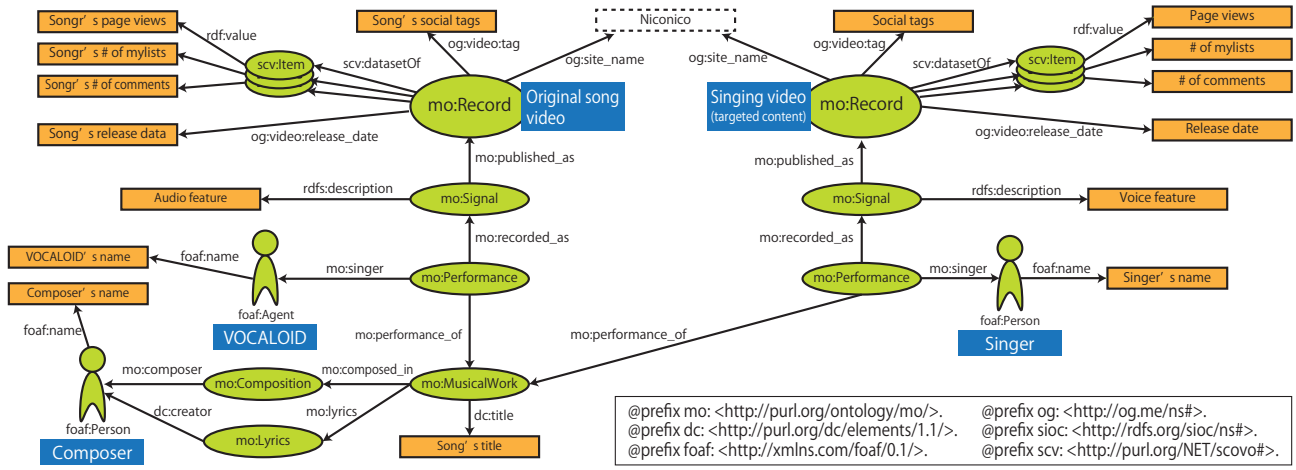

Figure 4. Illustration of all the metadata fields handled by QueryShare (in the style of the Music Ontology).

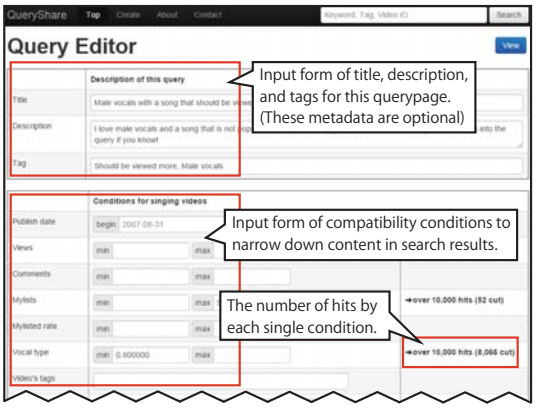

Figure 5. Screenshot of the editor of querypage showing many forms to input various search parameters.

might not know which of the conditions are too restrictive and which are too broad.

QueryShare presents the number of hits of each condition separately (query previews [16]). Additionally, it shows the number of hits of the entire query after the removal of each condition. Using this information, users can easily identify overly strict conditions or overly loose conditions that do not contribute to the search results (Fig. 5).

(B) Generating initial parameters from a content list: When users have playlists or lists of favorite songs, QueryShare can create initial parameters automatically from them. Users can then create new queries by modifying the values of the search parameters generated by the system.

QueryShare turns a playlist into a query by finding the full range for each search parameter. For numerical fields, it finds the minimum and maximum value: for example, if all the items in the playlist were published during 2008–2011, then this becomes the acceptable date range of the query. For nominal fields, it takes the union: for example, all the genre labels that appear in the playlist will be a part of the query.

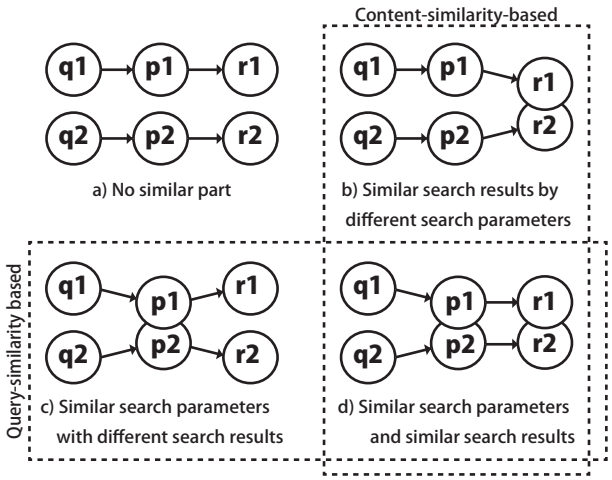

Figure 6. Four types of a pair of queries, where q_n denotes a query, p_n denotes a set of search parameters for q_n, and r_n represents search results of p_n. (a) q_1 and q_2 have no similar parts. They are mutually unrelated. (b) The queries have similar results that were sought using different search parameters. In this case, the content-based one can recommend q_2 for q_1. (c) The queries have similar search parameters with different search results. In this case, query-based one can recommend q_2 for q_1. (d) The queries have similar search parameters and similar search results. Both content-based ones and query-based ones can recommend this pair of queries.

Query recommendation

The system recommends queries to users as related queries when they access a querypage. QueryShare has recommendations of two types: query-based and content-based.

In query-based recommendation, the system recommends queries with search parameters that are similar to the original query (Figs. 6c and 6d). In the case of Fig. 6c, query q_2 can find parallel-hierarchical contents of s_1. For example, "Hit jazz songs with a male-like voice from 2011" and "Hit jazz songs with a male-like voice from 2012" are similar from the viewpoint of search parameters, but their search results include no overlaps.

In content-based recommendation, the system recommends queries based on the similarity of search results (Figs. 6b and 6d), which means that users can obtain similar search results through recommended queries, but users can also obtain search parameters to obtain similar search results. In the case of Fig. 6b, query q_2 shows a different aspect of these search results for query q_1. For example, "Hit jazz songs with a male-like vocal from 2011" and "Cool songs with my favorite 10 singers" do not overlap in terms of search parameters, but their search results might include common contents. Such recommended queries help users to improve or expand a query.

PRELIMINARY USER STUDY

Overviews

We conducted a study to gain user feedback on the QueryShare interface and to investigate the capabilities, limitations, and potential of our proposed search interaction. Five participants who were 20–30 years old took part in the study: P1 is a computer science researcher; P2 is a musician; P3 and P4 are programmers; and P5 is an office worker. P1, who was familiar with the music in the prototype system, very often listens to the music in Niconico. P2 and P3 sometimes listen to Niconico music. P4 and P5 were familiar with the music, but were not regular listeners.

Each participant was given a 5–15 min introduction to the prototype system by the instructor, the first author. The instructor then demonstrated query creation, search queries, and query surfing with query recommendations using the prototype system. Each participant was given 30 min maximum to view, create, and edit queries and listen to music in the search results. The prototype system was pre-loaded with about 100 queries by the instructor. However, this was found to be insufficient for users to appreciate the benefits of sharing queries. P2 reported that he hoped to use this system again after it had more queries.

During this open-ended task, the participants were free to ask any question about the usage of the system. They were allowed to end the study when they felt satisfied using the system. Each participant was asked to fill out a form with seven questions about the system, with responses given according to a seven-point Likert scale.

Results and Lessons Learned

We had four research questions. First, to test the fundamental motivation of our system, we asked (RQ1) Would users like to create complex queries for content retrieval? Next, regarding the feasibility of sharing queries, we wanted to know: (RQ2) Are users interested in others' queries, and (RQ3) Would users likely to share their queries with others? Finally, to establish the usefulness of the query surfing provided, we asked (RQ4) Can query recommendation support exploratory music search?

All participants were able to use the prototype system successfully. Only P5 did not create a query: she merely practiced query surfing. Results of the post-experiment questionnaire, consisting of the mean, standard deviation, and fraction of positive responses are presented in Table 2. The first four questions (Q1–4) pertain to the first research question (RQ1); the remaining questions (Q5–7) pertain to the other research questions (RQ2–4).

Table 2. Results of the post-experiment questionnaire. 'Mean' represents the average on a seven-point Likert scale. 'SD' denotes the standard deviation. 'Positive' denotes the fraction of positive responses (> 4 on a seven-point).

Question	Mean	SD	Positive
1. I'd like to reuse my queries.	4.5	1.73	3/4
2. I'd like to create queries more.	5.5	0.57	4/4
3. I'd like to use QueryShare again.	4.8	1.64	4/5
4. I'd like to use QueryShare for other domains.	5.8	1.30	4/5
5. I'd like to share others' queries.	4.75	1.89	3/5
6. I'd like to share my queries.	3.4	1.14	1/5
7. I encountered unexpected but potentially interesting songs.	4.6	0.89	2/5

RQ1: Would users like to create complex queries? The Q1 and Q2 results excluded the answer of P5 because she did not create queries. Only P3 reported negative impressions related to reusing personal queries (Q1). He responded that he had clear needs, but he was unable to represent such needs using search parameters. This response underscores the importance of assistance in creating queries. Although it is burdensome for people to create queries that include numerous search parameters, all respondents were positive about creating queries (Q2). Both P2 and P4 responded that creating queries was fun. In addition, each participant hoped either to use the prototype system again (Q3) or to use QueryShare in other domains (Q4). These results emphasize the remaining matter of assistance to create queries, but people accepted content searching with complex queries including many search parameters and hoped to use QueryShare. The response to RQ1 was strongly "Yes."

RQ2: Are users interested in others' queries? Two participants reported negative impressions to sharing of others' queries (Q5). However, both said that they were interested in queries created by people with similar musical interests. The other three participants were positive. They were asked a follow-up question "With what kind of people would you like to share?" They answered: my favorite creators (P1), eager listeners (P2), and people with similar taste (P5). In this experiment, we were unable to prepare a sufficient number of queries, but we found that everyone was interested in using others' queries. Based on these results, the answer to RQ2 was "Yes."

RQ3: Would users like to share their queries? Many participants reported negative impressions related to sharing their own queries (Q6). However, no one responded, "I refuse to share my queries." Both P1 and P2 responded that "I accept sharing of my queries, but I do not want to do that proactively." Some hoped to distinguish shared queries and private queries (P1, P4, P5) or to choose members to share them with (P1, P2). In contrast, P3 said that he does not care about any such matters. Even more noteworthy is that, P4 expected collaborative editing and reported that query sharing motivates him to formulate better queries. The answer to RQ3 is

a somewhat divided "Yes": opinion is divided, but most are willing to share some of their queries.

RQ4: Is query recommendation useful? Three participants gave neutral answers, but two other participants had encountered unexpected but potentially interesting songs (Q7). Everyone said that they would like to discover a query that yields search results that include unknown content. Those results demonstrate that they expected to encounter interesting content through query recommendation. Results show some potential for query recommendation for exploratory content search, but it is insufficient to show proof of the utility of query recommendation. The answer of RQ4 remains unclear. We need additional results from long-term experiments.

Limitations

The following presents a discussion of the limitations of this preliminary experiment. The current user study used a small test dataset for use in a small amount of time. We can discuss the possibilities of the proposed approach from this experiment, but we are unable to discuss the effectiveness of the proposed approach. For example, query recommendation has only a slight effect on query browsing because it often happens that the dataset has no applicable queries for recommendation. A user must use query recommendation to do query browsing if the system has many queries. Then, we can evaluate a performance of query recommendation and investigate the effect of a query recommendation. Therefore, future studies should use a larger dataset for testing.

Gathering data presents important difficulties for an open collaboration system. The proposed system can generate queries automatically from a list of items (see Section 4.3). It means that the system can get many queries automatically with crawling playlists that are published on the web. The quality of such automatically generated queries might be lower than human generated queries. However, people can refine such queries. Therefore, we think this function is a powerful way to solve a cold start problem of the proposed system.

DISCUSSION

Sharing playlists versus sharing queries

Many services, e.g., video sharing services and music streaming services, have a function that enables sharing of mylists in which users save favorites. Sharing queries include sharing of the search results, which are collections of contents. From this perspective, sharing queries is similar to sharing playlists because both share user-generated content collections. The most important difference is the mode of creation by users. Users create mylists by listing contents. Consequently, a mylist represents favorite or interesting contents with an extensional definition. However, users create queries by setting search parameters in QueryShare. Other users can see not only a collection of contents but also their search parameters. Therefore, a query represents a favorite or interesting content with an intentional definition.

This difference affects (1) the readiness for numerous contents that are growing day-by-day, and (2) the evolution of the growth of users' ability to discover contents. Mylists cannot adapt to new contents without the owner's maintenance. However, queries apply new contents automatically without the owner's effort. Sharing queries is a kind of sharing of tiny codes [9].

Folk + Query = Folkquery

Weinberger pointed to the future of information categorization with folksonomy, which is organized by people rather than by experts [43]. To a great degree, only knowledge organized by experts was distributed to people before the emergence of the internet. Those circumstances have changed drastically. Currently it is difficult for us to obtain information without some search or recommendation because vast amounts of information exist in front of us and in various locations worldwide. Can one truly encounter vast amounts of information and annotate them for subsequent categorization?

Pariser reported that personalization enables a person to be holed up in a small bubble called a "Filter Bubble" [36, 38]. Consequently, users become separated from information that disagrees with their viewpoints, effectively isolating them in their own small bubbles of opinion and information. From the beginning, searching is an original means to obtain information that lies within someone's grasp. Nevertheless, people will be in a small bubble generated by search engines if queries are not elaborated and are simply entrusted to the search engines and their managers.

We proposed sharing queries, which are working together to produce queries to obtain increasing amounts of information. This practice might be called "Folkquery". Folkquery solves the missing part of folksonomy in an age of searching.

CONCLUSION

As described in this paper, we propose an exploratory music search interface called *QueryShare*, which has three key features: *contentification*, *query surfing*, and asynchronous collaborative editing. We developed the prototype system as a web application for music videos on *Niconico*. Results of a preliminary user study with the prototype system revealed that users found the query creation interesting and that they were interested in seeing and using queries by other users. On the other hand, many users reported negative impressions related to sharing their own queries. It is necessary to design a more appropriate, powerful and clear incentive for users to create queries and share them. Furthermore, we discuss the importance of people's use of complex queries.

Open collaboration creates large amounts of user-generated content. Organizing this content to be searchable is an important issue. Within QueryShare, our proposed solution allows users to query creation, query reusing, query surfing, and query refinement. These search interactions contribute to facilitate exploratory music searches without skills in creating queries. We consider QuerySharing to be a simple yet powerful means of searching any type of content. QuerySharing can bring more complex queries to users and content. Therefore, they can evolve to provide a rich environment for understanding music content.

As future work, we expect to open the prototype system to the public and to improve it based on user feedback. Furthermore, we would like to analyze user behaviors in the proposed search interaction.

ACKNOWLEDGMENTS

We thank Keisuke Ishida for web service implementation. We thank Tomoyasu Nakano and Graham Percival for helpful comments on earlier drafts of this paper. This work was supported in part by JST ACCEL Grant Number JPMJAC1602, JSPS KAKENHI Grant Number JP15H02781 Japan.

REFERENCES

1. Ibrahim Adepoju Adeyanju, Dawei Song, M-Dyaa Albakour, Udo Kruschwitz, Anne De Roeck, and Maria Fasli. 2012. Adaptation of the Concept Hierarchy Model with Search Logs for Query Recommendation on Intranets. In *Proc. SIGIR '12*. 5–14.
2. Javad Akbarnejad, Gloria Chatzopoulou, Magdalini Eirinaki, Suju Koshy, Sarika Mittal, Duc On, Neoklis Polyzotis, and Jothi Swarubini Vindhiya Varman. 2010. The QueRIE system for Personalized Query Recommendations. In *Proc. the VLDB Endowment*, Vol. 3.
3. Senjuti Basu Roy, Haidong Wang, Gautam Das, Ullas Nambiar, and Mukesh Mohania. Minimum-effort Driven Dynamic Faceted Search in Structured Databases.
4. Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query Suggestions in the Absence of Query Logs. In *Proc. SIGIR '11*. 795–804.
5. Vijay Chandrasekhar, Matt Sharifi, and David A. Ross. 2011. Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications. In *Proc. ISMIR 2011*. 801–806.
6. Gloria Chatzopoulou, Magdalini Eirinaki, and Neoklis Polyzotis. 2009. Query Recommendations for Interactive Database Exploration. In *Proc. SSDBM '09*. 3–18.
7. S. Cucerzan and E. Brill. 2004. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proc. EMNLP 2004*. 293–300.
8. Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proc. WWW 2002*. 325–332.
9. Koichiro Eto, Masahiro Hamasaki, and Hideaki Takeda. 2012. Wedata: a wiki system for service oriented tiny code sharing. In *Proc. of WikiSym '12*. 30:1–30:4.
10. Henry Feild and James Allan. 2013. Task-aware query recommendation. In *Proc. SIGIR '13*. 83–92.
11. Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. 1995. Query by humming: musical information retrieval in an audio database. In *Proc. ACM Multimedia '95*. 231–236.
12. Arnaud Giacometti, Patrick Marcel, Elsa Negre, and Arnaud Soulet. 2009. Query recommendations for OLAP discovery driven analysis. In *Proc. DOLAP '09*. 81–88.
13. Gene Golovchinsky, Abdigani Diriye, and Jeremy Pickens. 2011. Designing for Collaboration in Information Seeking. In *Proc. of HCIR '11*.
14. Masataka Goto and Takayuki Goto. 2009. Musicream: Integrated Music-Listening Interface for Active, Flexible, and Unexpected Encounters with Musical Pieces. *IPSJ Journal* 50, 12 (2009), 2923–2936.
15. Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. 2011. Songle: A Web Service for Active Music Listening Improved by User Contributions. In *Proc. ISMIR 2011*. 311–316.
16. Stephan Greene, Egemen Tanin, Catherine Plaisant, Ben Shneiderman, Lola Olsen, Gene Major, and Steve Johns. 1999. The end of zero-hit queries: query previews for NASA's Global Change Master Directory. *International Journal on Digital Libraries* 2, 2-3 (1999), 79–90.
17. Masahiro Hamasaki and Masataka Goto. 2013. Songrium: a music browsing assistance service based on visualization of massive open collaboration within music content creation community. In *Proc. WikiSym '13*. 4:1–4:10.
18. Masahiro Hamasaki, Masataka Goto, and Tomoyasu Nakano. 2014. Songrium: A Music Browsing Assistance Service with Interactive Visualization and Exploration of a Web of Music. In *Proc. WWW 2014*. 523–528.
19. Masahiro Hamasaki, Hideaki Takeda, and Takuichi Nishimura. 2008. Network Analysis of Massively Collaborative Creation of Multimedia Contents - Case Study of Hatsune Miku videos on Nico Nico Douga -. In *Proc. of uxTV '08*. 165–168.
20. Philippe Hamel, Matthew E. P. Davies, Kazuyoshi Yoshii, and Masataka Goto. 2013. Transfer Learning In MIR: Sharing Learned Latent Representations For Music Audio Classification And Similarity. In *Proc. of ISMIR 2013*. 9–14.
21. Katsutoshi Itoyama, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. 2008. Instrument Equalizer for Query-by-Example Retrieval: Improving Sound Source Separation Based on Integrated Harmonic and Inharmonic Models. In *Proc. ISMIR 2008*. 133–138.
22. Bernard J. Jansen, Amanda Spink, and Jan Pedersen. 2005. A temporal comparison of AltaVista Web searching: Research Articles. *J. Am. Soc. Inf. Sci. Technol.* 56, 6 (2005), 559–570.
23. Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating Query Substitutions. In *Proc. WWW '06*. 387–396.
24. Tomoko Kajiyama and Shin'ichi Satoh. 2013. A Video Navigation Interface Using Multi-faceted Search Hierarchies. In *Proc. of MMSys '13*. 136–140.

25. Mohsen Kamalzadeh, Dominikus Baur, and Torsten Moller. 2012. A Survey on Music Listening and Management Behaviors. In *Proc. of ISMIR 2012*. 373–378.
26. Kunio Kashino, Takayuki Kurozumi, and Hiroshi Murase. 2003. A quick search method for audio and video signals based on histogram pruning. *IEEE Trans. on Multimedia* 5, 3 (2003), 348–357.
27. Youngho Kim and W. Bruce Croft. 2014. Diversifying Query Suggestions Based on Query Documents. In *Proc. SIGIR '14*. 891–894.
28. P. Knees and M. Schedl. 2013. A Survey of Music Similarity and Recommendation from Music Context Data. *ACM TOMM* 10, 1 (2013), 1–21.
29. Mu Li, Muhua Zhu, Yang Zhang, and Ming Zhou. 2006. Exploring Distributional Similarity Based Models for Query Spelling Correction. In *Proc. ACL 2006*. 1025–1032.
30. Gray Marchionini. 2006. *Exploratory search: from finding to understanding*. Communications of the ACM, Vol. 49. ACM, 41–46.
31. Yevgeniy Medynskiy, Mira Dontcheva, and Steven M. Drucker. 2009. Exploring Websites Through Contextual Facets. In *Proc. of CHI '09*. 2013–2022.
32. Meredith Ringel Morris. 2008. A survey of collaborative web search practices. In *Proc. of CHI '08*. 1657–1660.
33. Meredith Ringel Morris and Eric Horvitz. 2007. SearchTogether: an interface for collaborative web search. In *Proc. of UIST '07*. 3–12.
34. Elias Pampalk, Andreas Rauber, and Dieter Merkl. 2002. Content-based Organization and Visualization of Music Archives. In *Proc. ACM Multimedia 2002*. 570–579.
35. Bryan Pardo (Ed.). 2006. *Special issue: Music information retrieval*. Communications of the ACM, Vol. 49. ACM, 28–58.
36. Eli Pariser. 2011. *The Filter Bubble: What The Internet Is Hiding From You*. Penguin.
37. Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. 2007. The Music Ontology. In *Proc. of ISMIR '07*. 417–422.
38. Paul Resnick, Joseph Konstan, and Anthony Jameson. 2011. Panel: Recommender Systems and the 'Filter Bubble'. In *RecSys '11*. `http://recsys.acm.org/recsys11/panel/`
39. Yading Song, Simon Dixon, and Marcus Pearce. 2012. Survey of Music Recommendation Systems and Future Perspectives. In *Proc. CMMR 2012*. 395–410.
40. Wei Ho Tsai, Hung Ming Yu, and Hsin Min Wang. 2005. A Query-by-Example Technique for Retrieving Cover Versions of Popular Songs with Similar Melodies. In *Proc. ISMIR 2005*. 183–190.
41. Michael B Twidale and David M Nichols. 1997. Designing interfaces to support collaboration in information retrieval. *Interacting with Computers* 10, 2 (1997), 177–193.
42. Chung-Che Wang, Jyh-Shing Roger, and Jang Wennen Wang. 2010. An Improved Query by Singing/Humming System Using Melody and Lyrics Information. In *Proc. ISMIR 2010*. 45–50.
43. David Weinberger. 2007. *Everything Is Miscellaneous: The Power of the New Digital Disorder*. Times Books.
44. Stewart Whiting and Joemon M. Jose. 2014. Recent and Robust Query Auto-Completion. In *Proc. WWW 2014*. 971–981.
45. Xiangmin Zhang and Yuelin Li. 2005. An Exploratory Study on Knowledge Sharing in Information Retrieval. In *Proc. HICSS '05*. 245c–245c.