ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 5, May 2016

QUERY BY HUMMING SYSTEM USING PERSONAL HYBRID RANKING

ARPITA SHRIVASTAVA, DHVANI PANDYA, PRIYANKA TELI, YASH SAHASRABUDDHE

ABSTRACT

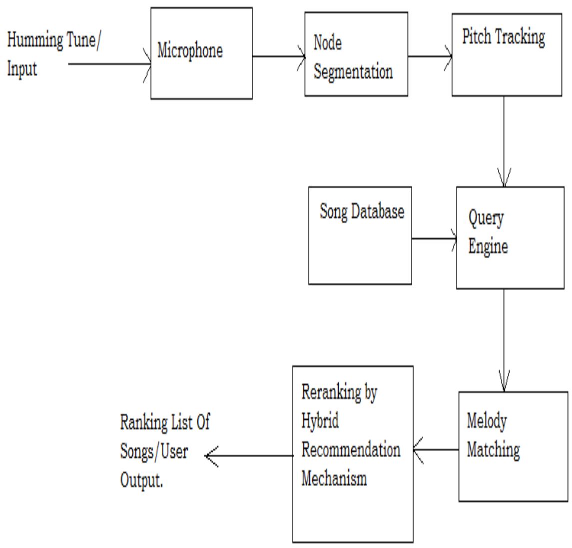
With the increasing use of smart devices, many applications have been developed. Listening to music is one of the facilities provided by one of these applications. Users prefer listening to music during their daily chores like while exercising or driving etc. In terms of song searching, user can use voice recognition systems or user can just type the name of the artist or genre or title of the song. But how to search the tune of the song that has been stuck in our head all day long? To solve such problem we are providing QBH (Query by Humming) system where the user can hum the tune and with the help of signal processing, the system will recognize and rank the results obtained based on Personal Hybrid Ranking Method. The effectiveness of the system is characterized by the parameters such as remarkable accuracy, throughput and response time.

I. INTRODUCTION

With the advent of new astonishing technologies, consumers can now use internet to play any type of music anywhere, anytime by searching it very easily. Automatic playlist creation, music recommendation or music search are related problems. Searching song by singing/humming a section of it is the most natural way to search the song. This search method is useful when user don't have access to audio device or user is unable to recall the attributes of the song like song title, name of artist, name of album. Therefore "Effective Result Ranking of Mobile Query by Humming Using Hybrid Recommendation Mechanism",[1] which is the mechanism to search song sung/hummed by user as well as it recommends user by doing effective ranking of the results obtained. Re-ranking the results according to the user's profile and previous search history is the most challenging task in this system.

In terms of song searching, one can use a song's metadata (for example, song title, artist, or publication date, etc.) or the content of a music file. A user can easily search for a song through a voice recognition system. In order to perform a search, the user can call out the song title or artist. However, people often cannot recall the song title or the name of the artists, and only part of the melody is remembered. Currently there are many search engines on internet which allow user to search the songs using song related details like song title, artist name, movie name etc. But there are very few systems which do search using acoustic query as input. Many search engines based on acoustic input search song using same details of song mentioned earlier but by using voice recognition technology which gives more comfort to user by reducing his/her typing task. There are a limited number of system currently available which gives freedom to user to search the song by singing/humming segment of song and. The QBH System Using Personal Hybrid Ranking [23], provides a facility to search song by using melody sang/hummed by user and re-rank the searched result by using hybrid recommendation method.

II. ARCHITECTURE



Proposed Architecture

III. WORKING

A) The Singing/Humming Signals Process

A hand-held microphone is used to record the audio signals of a user after the user sang or hummed the song. We apply a fixed noise gate to determine the end-point of notes for signals generated by singing/humming in order to segment the notes. In other words, when the energy level is constantly lower than the configured threshold, the energy interval is considered to be a no-pitch interval. A no-pitch interval begins at the end of a certain note, where this end position is the start position of the next note. The audio wave diagram representing the input signals from a user singing/humming a melody is shown. The red columns determine the positions of notes detected by their energy level. We use to represent each note segment after cutting.

We then use a YIN pitch-tracking algorithm to convert the pre-processed audio signal into pitches (fundamental frequencies) [20].We transfer the signal sang/hummed back to the server via the Internet for processing. The YIN, fundamental frequency estimator algorithm is then used to perform the pitch tracking; this method gives frequencies in the form of interval so we calculate average of frequency interval it works as note segmentation and we consider one average frequency as corresponding frequency of one note [1].

A user may have different singing sophistication and skill. The same song segment sang by different people would result in different tonality. In other words, two melodies with the same contour of pitch scales can also be recognized as being identical even though they have different values in the pitch scale. We used the relative pitch notation commonly used in QBH systems to represent the variations in the melodies sang/hummed this is done by comparing each calculated average frequency with frequencies and we assign the note after finding nearest note frequency of calculated frequency. Then we create the related contour string of a melody sang/hummed by user by calculating difference between note in to its previous note simultaneously.

B) Matching Method

Common sound formats (for example, MP3 or wave) are often polyphonic. Direct comparison of a user's singing/humming with the polyphonic format is not possible. Methods that automatically extract the melody from polyphonic audio exist, but the accuracy is still an area for improvement. In our system, the music library not only collects standard music files, but also collects the corresponding MIDI files. A manual process is required to mark the melody by using FL-studio Software tool this software allows to edit MIDI file and remove the unwanted event of MIDI. The melody is then represented in monophonic form. We acquire each frequency of MIDI event to find the MIDI contour string as query is converted in to contour string the music in the database also contain there related contour string [4]. The difference between the query contour string and the melody contour string is calculated based on the edit distance [6]. Since we cannot determine the end of the line when a user sings/hums, the main melody contour strings generate substrings with arbitrary length at arbitrary starting points. We perform the edit distance method on the query contour string and all substrings in every song in the database. We determine possible results based on the edit distance [5]. Since this involves a significant level of processing. To avoid a song with two or more segments matching a query, and all of these segments becoming the search results, for every song, we only consider the segment with the smallest edit distance to be the determining factor used to decide if this segment can be a search result or not. For the search process, k songs with small edit distances are considered to be the potential results. They are called k candidates.

i. Re-ranking by Similar Queries

When a user performs a query for a song, it does not matter how many times that song has been searched, the segment sang/ hummed by the user often shows similar signals [9]. If the user is searching for a song that has been searched before, to avoid repeating the time spent on transferring and detecting the same song, we record every query made by the user, and the corresponding results that are said to be correct by the user we request user to select song that are accurately search by system to maintain user's history, if song is not accurately given by primary list then we compare query with queries in user's history and recommends result to user. To perform this we have used Content Based Retrieval System [2][3].

ii. Re-ranking by User's Preference

In the first search stage, we compare the query with the previous search results. In the ranking process that follows, we consider the users preferences. A user often searches for a song that is related to their interests. For example, for a user who likes Rock music, their probability of searching for a Rock song is higher than that of a classical song. Based on this hypothesis, in the second searching stage, in addition to comparing the query with previous history, we also consider the genre of music files that were searched by the user previously. This method takes primary list as an input and calculates user preference by using his search history for example suppose in k candidates, song M1 has corresponding attribute values of singer gender= "male", Year= "1990",genre= "rock" and tempo= "fast" and user search history contain 5 songs which includes 3-male,2-female singer gender,3 songs from 1990,2-has rock genre and 2 songs in history has tempo fast so we calculate probability of user likes the song M1 is (3/5 +3/5 + 2/5 +2/5)/4= 0.5 and song M2 has corresponding attribute values are singer gender= "female", Year= "1990", genre= "hip-hop" and tempo= "fast" therefore probability of song M2 is (2/5 +3/5 + 0/5 +2/5)/4= 0.35 upon comparison of this two songs, the user would be more likely to search for M1[1] .

iii. Re-ranking through Similar Users' Records

The ranking of search results based on user preference is affected by the number of queries performed by individuals. In other words, when there are only a few queries performed, it is difficult to estimate a user's music preference. This is called the cold-start problem. To resolve this, we have designed a collaborative method to rank the songs in candidates. We determined that different age groups have different music preferences from our observations of the music behaviour of average users. Different education levels also affect the type of music that people listen to. When designing our collaborative method, we use the basic information collected from each user and perform preference analysis on each attribute. We use the age and profession of a user to find his similar user and if similar user is found we recommend songs that are similar to user's query from the similar user's search history.

IV. IMPORTANT MODULE AND ALGORITHMS

1. Login Module- 1) User Login- This module will display all the User name and password in the table of that particular category.

2) Admin Login- This module will display the admin name and password in the table.

2. Registration Module- This module is used to add the new user whole information. Adding the User name, id, and password are the compulsory fields to be filled. Special comment is optional field.

3. Song upload Module- This module is used to upload the details of the song on which user has clicked in Display the song and also search the song. Details include the song id, song name, genre, year, MIDI song, MP3 song.

4. Recording Module- Display the recorded song counter string in the database.

5. Processing Module- 5.1) Pitch Tracking Module- In this module is used to convert the pre-processed audio signal into pitches (fundamental frequencies).

5.2) Note Segmentation - The audio signal sang/hummed cannot constantly stabilize on a particular frequency. The frequencies of each sampling point in the audio signals are matched to the frequencies of the notes. Segmentation of notes is already performed on the input audio signal. After converting every sequence of the input audio signals into notes, the segments with the largest number of notes is chosen as the notes represents this segment.

6. Query Contour string - In this module generate the Query Contour String by calculating the difference of each note and its previous note simultaneously.

7. MIDI Contour String Module- In this Module Generate Contour string of MIDI file is generated then song will be uploaded [19].

8. Matching Module - In this Module to match the Query Contour String and MIDI contour String.

9. Ranking Module- In this Module to match the Query Contour String and MIDI contour String after matching it display the primary list of song and these songs is Rank using this module. We used genetic algorithm technique to perform this re-ranking [7].

10. Re-ranking using search history - This module includes matching of query contour string with the query in user's search history.

11. Re-ranking using user Preference- This module gets primary list as an input and calculates probabilities of each song to be user like.

12. Re-ranking using similar user history - In this module system firstly finds similar user by using age and profession and search song by comparing query with query of similar user history.

V. TEST CASES AND RESULTS

1) Validation Testing: Validation testing has been performed to verify that incorrect entries are not made in incorrect fields.

A) If, enter the name of user, the user tries to enter the number in the field named "Enter the Alphabets only" then an error message is prompted.

B) If enter the id will be less than 5 no then the error message is prompted.

C) If enter the password in user less than 7 no then the error message is prompted.

D) A validation has been performed on every field in java application. The details filled in each field would ultimately be updated in database.

2) Ad-Hoc Testing: Ad-Hoc testing has been performed by us by checking the different scenarios and also checking the code randomly.

3) Condition Testing, Data flow testing, Loop testing, simple loop testing, and Nested loop testing:

The entire code has been tested. Condition Testing, Data flow testing, Loop testing, Simple loop testing, Nested loop testing all have been performed after the coding phase.

4) Smoke testing: Smoke testing is testing the code to find errors and verify that the cost to fix bugs is not more than creating a new application. If the cost is more, then we develop a totally new code instead of rectifying errors in existing code. We have performed this type of testing.

VI. CONCLUSION:

The Effective Results Ranking for Mobile Query by Singing/Humming Using a Hybrid Recommendation Mechanism is the most natural and simple technique to perform music search. An individual does not need to know the title of the song or name of the artist to perform the search. This system generates the textual representation of human voice and also creates the related textual representation of digital musical instruction and compared them for search. System uses users search history to know user's preference. System is able to find users from similar age groups and profession to search similar user and recommends songs from similar user's search history. This improves the rate of success. Such concepts often appear in system design recommendations.

VII. FUTURE SCOPE:

The future work includes study to directly extract melody from standard file format. The system, takes a long time to calculate the edit distance between each string. This causes significant delays when returning the results. A user may discontinue use of this system due to the delay. This problem is also an important research topic in order to reduce the time required to calculate the similarity. All of our experiments are currently performed in a quiet environment. In the real world, this is often not the case and it is expected that devices will be operated in noisy environments. A further area of future research is he filtering of noise to reduce the errors during contour string conversion. The existing system requires manual intervention to establish the corresponding MIDI data for the melodies. An automatic system to perform this function also requires study.

VIII. REFERENCES

[1] Ning-Han Liu "Effective Results Ranking for Mobile Query by Singing/Humming Using a Hybrid Recommendation Mechanism", IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 16, NO. 5, AUGUST 2014

[2]O. Cornelis, M. Lesaffre, D. Moelants, and M. Leman, "Access to ethnic music: Advances and perspectives in content-based music information retrieval," Signal Process., vol. 90, no. 4, pp. 1008–1031, Apr. 2010.

[3] K. Lemström, N. Mikkilä, and V.Mäkinen, "Filteringmethods for content-based retrieval on indexed symbolic music databases," Inf. Retriev.,vol. 13, no. 1, pp. 1– 21, Feb. 2010.

[4] A. J. Ghias, D. C. Logan, and B. C. Smith, "Query by humming-musical information retrieval in an audio database," in Proc. ACM

Multimedia'95, San Francisco, CA, USA, 1995, pp. 216–221.

[5] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham, "Toward the digital music library: Tune retrieval from acoustic input," in Proc. ACM Digital Libraries, 1996, pp. 11–18.

[6] X. Yang, Q. Chen, and X. Wang, "A novel approach based on fault tolerance and recursive segmentation to query by humming," Adv. Comput. Sci. Inf. Technol., vol. 6059, pp. 544–557, 2010.

[7] J. Qin, H. Lin, and X. Liu, "Query by humming systems using melody matching model based on the genetic algorithm," J. Softw., vol. 6, no.12, pp. 2416–2420, Dec. 2011.

[8] J. Li, L. M. Zheng, L. Yang, L. J. Tian, P. Wu, and H. Zhu, "Improved dynamic time warping algorithm the research and application of query by humming," in Proc. IEEE 6th Int. Conf. Natural Computation (ICNC), Aug. 2010, pp. 3349–3353.

[9]H. M. Yu, W. H. Tsai, and H. M. Wang, "A query-by-singing system for retrieving karaoke music," IEEE Trans. Multimedia, vol. 10, no. 8, pp. 1626–1637, 2008.

[10] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating „Word of Mouth", " in Proc. ACM CHI'95 Conf. Human Factors in Computing Systems, 1995, pp. 210–217.

[11] Q. Li and B. M. Kim, "Clustering approach for hybrid recommender system," in Proc. IEEE/WIC Int. Conf. Web Intelligence, 2003, pp.33–38.

[12] Y. H. Cho and J. K. Kim, "Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce,"Expert Syst. Applicat., vol. 26, no. 2, pp. 233–246, 2004.

[13] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Generative models for cold-start recommendations," Proc. SIGIR Workshop Recommender Systems, vol. 6, pp. 253–260, 2001.

[14] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," Commun. ACM, vol. 40, no. 3, pp. 77–87, 1997.

[15] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences," in Proc. 7th Int. Society for Music Information Retrieval (ISMIR), Oct. 2006, pp. 296–301.

[16] Ò. Celma, "Music recommendation and discovery in the long tail," Ph.D. dissertation, Dept. Inf. Commun. Technol., Univ. Pompeu Fabra, Barcelona, Spain, 2008.

[17] N. H. Liu, S. W. Lai, C. Y. Chen, and S. J. Hsieh, "Adaptive music recommendation based on user behavior in time slot," Int. J. Comput. Sci. Netw. Security, vol. 9, no. 2, pp. 219–227, 2009.

[18] S. K. Lee, Y. H. Cho, and S. H. Kim, "Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations," Inf. Sci., vol. 180, no. 11, pp. 2142–2155, Jun. 2010.

[19] "Standard MIDI-File Format Spec. 1.1" Distributed by: The International MIDI association 5316 W.57th St. Los Angeles, CA 90056(213) 649-6434

[20] "YIN, a fundamental frequency estimator for speech and music" Alain de Cheveigne Icram-CNRS, 1 place Igor Stravinsky, 75004 Paris, France.

[21] "Social Information Filtering: Algorithm for Automatic ‚Word of Mouth" upender Shardanand, Pattie Maes MIT Media-Lab 20 Ames Street Rm. 305 Cambridge, MA 02139.

[22] "Core Web programming HTML Forms Sending Data to server-side Programs" 2001-2003 Marty Hall, Larry Brown.

[23] "QBH System using Personal Hybrid Ranking", http://www.ijsres.com/2015/vol-2_issue-10/paper_9.pdf.

Arpita Srivastava , Student, SKNSITS, Lonavala.

Dhvani Pandya, Student, SKNSITS, Lonavala.

Priyanka Teli, Student, SKNSITS, Lonavala.

Yash Sahasrabuddhe, Student, SKNSITS, Lonavala.