LRC Sousou: A Lyrics Retrieval System

Yong Han[1], Li Min[1], Yu Zou[1], Zhongyuan Han[3,1], Song Li[1], Leilei Kong[1,2], Haoliang Qi[1], Wenhao Qiao[4], Shuo Cui[5], and Hong Deng[1]

[1] School of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin, China
[2] College of Information and Communication Engineering, Harbin Engineering University, Harbin, China
[3] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[4] Beijing Institute of Surveying and Mapping, Beijing, China
[5] China Resources, Shenzhen, China
song12307178126.com

Abstract. Lyrics retrieval is one of the frequently-used retrieval functions of search engines. However, diversified information requirements are neglected in the existing lyrics retrieval systems. A lyrics retrieval system named LRC Sousou, in which erroneous characters are corrected automatically, the mixed queries of Chinese words and Pinyin are supported, and English phonemes queries are also achieved effectively, is introduced in this paper. The technologies of natural language processing, information retrieval and machine learning algorithm are applied to our lyrics retrieval system which enhance the practicability and efficiency of lyrics search, and improve user experience.

Keywords: Lyrics retrieval, information retrieval, learning to rank.

1 Introduction

According to the CNNIC report, by the end of June 2014, the using frequency of network music is up to 77.2%, ranked as the fourth in all network software[1]. Therefore, lyrics retrieval system is an often used function. Lyrics search is asked to retrieve and return the information about the related song (e.g. title, lyric, artist, album, etc.) and then show them to the user after the queries such as the title of the song, a part of the lyric or the artist etc. are submitted. Nevertheless, in practical application, users submit not only standard queries but also nonstandard queries, such as inaccurately remembered lyrics, wrong Chinese spelling, Pinyin used to instead of some forgotten Chinese words, as well as English words which are known only the pronunciation but none of their spelling. In these special cases, traditional information retrieval technologies based on keywords matching are challenged. As a result, users have to change keywords repeatedly to get the satisfactory results. Thus it is useful in improving the user experience by providing diverse, humanized query support and meeting the requirements above.

For this purpose, we implement a lyrics retrieval system named LRC Sousou by using the techniques of natural language processing, information retrieval and machine learning algorithm. The system not only can correct the erroneous characters automatically, support mixed query of Chinese words and Pinyin, but also support the English phonemes retrieval. Specifically, the system implements the function of correcting erroneous characters via the natural language processing technique, realizes the Chinese word and Pinyin mixture retrieval by taking advantage of mixture index. For the sake of homonyms retrieval, English phonemes index and Chinese fuzzy pronunciation index are used. In order to complete the final results ranking, we merge multiple indexes by applying learning to rank algorithm.

2 System Design

There are four main components in the lyrics retrieval system, namely, Crawler and Preprocessing, Build index, Query modification and Retrieval. As is shown in Fig 1.

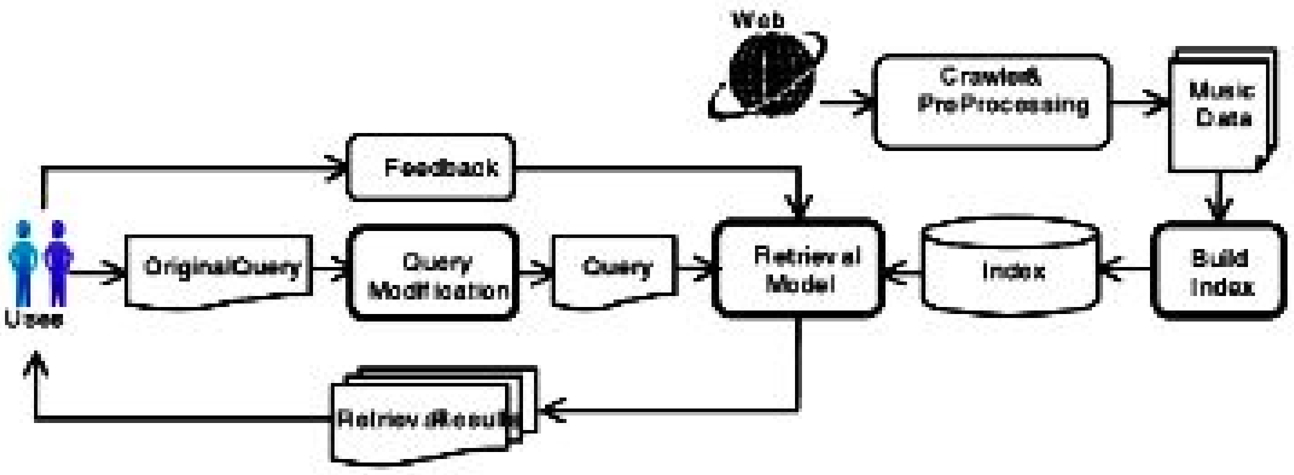

Fig. 1. Framework of the lyrics retrieval system

Crawler and Preprocessing Component crawls the LRC files and the web pages that contain the lyric information from the Internet. We extract information like lyric, artist, album and so on by applying Stanford Named Entity Recognizer (NER)[1].

Build Index Component. In order to meet various retrieval needs, several index strategies, such as word segmentation by Stanford Word Segmenter[2], n-gram, Pinyin and English phonemes based on double metaphone[2], are used to build index respectively on the on the lyric, the title of the song, the artist as well as the album.

Query Modification Component supplies and revises the queries. To conduct the Chinese query modification, we turn the Chinese words into Pinyin and approximate pronunciation firstly, and then use a translation model convert Pinyin to Chinese words. For instance, the original query "潜力值外" is modified to the query

[1] http://nlp.stanford.edu/software/CRF-NER.shtml
[2] http://nlp.stanford.edu/software/segmenter.shtml

"千里之外". Moreover, the English words which have similar pronun-ciation are added into the query. The modified query contains the original query, revised Chinese word, Pinyin, approximate pronunciation Pinyin and English.

Retrieval Model Component. The 68 similarity scores between the parts of modified query and fields are calculated respectively by a special vector space model[3]. To combine these features, RankLR[4], a learning to rank algorithm based on logistic regression, is employed to train a retrieval model with labeled data. Besides, the user feedbacks, such as user click and LRC download, are used to adjust the retrieval model.

3 Demonstration

The LRC Sousou has download and index 160288 LRC files from internet by lucene toolkit[3]. Users can use the accurate retrieval option to input precise query to research the artists, song name, Lyrics, album or all of them. And then, smart retrieval option enables users to submit ambiguous query. Fig. 2 shows the result of an ambiguous query "zhai哪套话sengka的地方". The retrieval results show that the LRC Sousou can find out the songs named "在那桃花盛开的地方" and some other songs partially matching of "桃花", "的地方" and so on. As is shown in Fig.3, the result of a similar pronunciation query "koo kool kat" includes the songs named "cool cool cat" songs and others relevant songs.

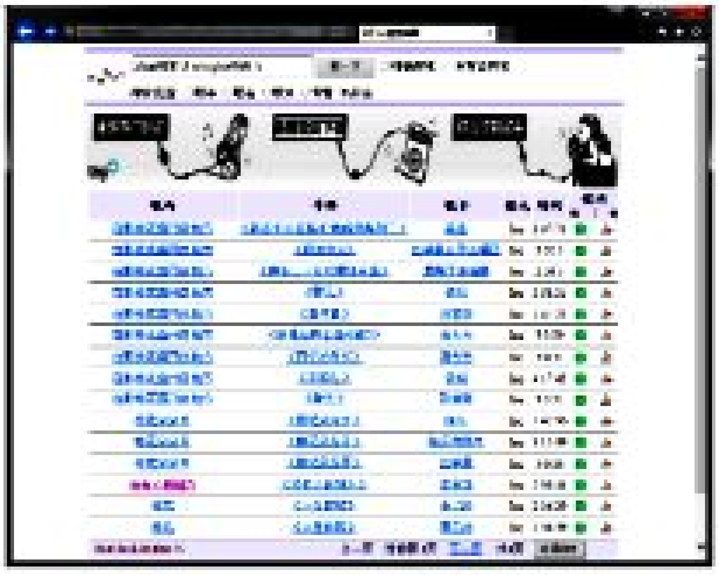

Fig. 2. The example of Chinese ambiguous query

[3] http://lucene.apache.org

shops using map services. The problem is especially severe for people with disabilities, who would prefer shopping in one single region to avoid inconvenience in transportation. While these kind of application will increase time complexity obviously with bad algorithm, which are not suitable for the real online server with such large amount of data.

In this paper, we propose a novel region-based map search method to reduce the expected shopping distance for people with disabilities. Rather than returning isolated POIs, we find dense areas containing multiple POIs as the query result. To improve the algorithm efficiency, we use high order Voronoi diagram to find dense regions containing the query keywords. Our algorithm has a time complexity of $O(n^2)$, which can well satisfy the time requirement of online search.

2 Related Work

Various Geographic Information Systems (GIS) have been developed to provide diversified Location Based Services (LBS) nowadays, include accessibility information service [2]. However, existing works mainly focus on finding Points-of-Interest (POIs) to for a user query by considering both the keyword similarity and geographic distance. With huge amount of data stored in GIS systems, many methods are used to speed up the searching process, such as R-tree and so on [3].

Although it is generally recognized that Regions-of-Interests (ROIs) can better satisfy users search need in many cases that POIs, few systems for ROI-based map service exist because calculating the relevance of an ROI is often too expensive to be applied in practical online service. There are existing researches for retrieving target ROIs from a given candidate ROI set for a user query such as [4]. The results of these methods are sub-optimal and the service quality is highly dependent on the predefined candidate ROI set.

3 Map Region Search

While existing map search mainly focuses on recommending isolated stores for a query, our purpose is to find relevant ROIs to reduce the expected shopping distance for users. We will first define the relevant terms and discuss the measurement for the expected shopping distance.

We assume that most people will finish a shopping task by visiting no more than k stores and we define a set with k adjoining POIs as a candidate *region* for user queries. It is natural to choose the variance of the distances between k stores in this region as a measurement for the *shopping distance*. A region with low variance is the one densely populated k stores, whose expected shopping distance will generally be much shorter than the one with higher variance. Thus the map region search problem can be defined as follows: given a set of n POIs each labeled with multiple query keywords, we want to find the region (which is a set of k POIs) with the minimum variance.

This problem is non-convex in the discrete solution space, meaning no global optimal solution can be found for it generally. The exhaustive search in the solution space would be prohibitively expensive for large data sets (the complexity is $O(n^k)$), which is not suitable for online search services. A naive k-NN search aiming to reduce the search space is to examine the regions generated by a single POI with its $(k-1)$ nearest neighbors. This method can reduce the searching complexity into $O(n)$, but the result is sub-optimal, as shown in the example in Fig. 1:

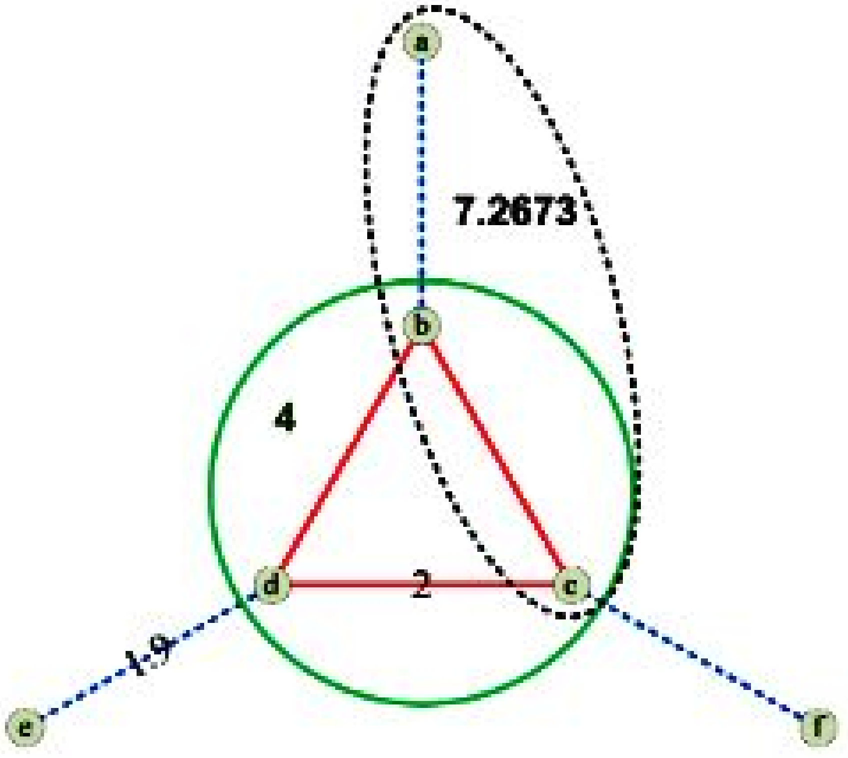

Fig. 1. An example which k-NN method lose the optimal solution

We want to find the optimal region containing 3 stores from 6 stores in Fig 1. The distance denoted by the blue line between stores is 1.9, while red line is 2. k-NN method can only retrieve the purple regions with a variance of 7.2673, which is similar as the set $\{a, b, c\}$, but the optimal region is the green one with a variance of 4, which is the set $\{b, c, d\}$.

In this paper, we use high order Voronoi diagram to improve map region search. Voronoi diagram has been thoroughly studied [5] and widely applied in 2-D space data analysis [6]. Given n points in the 2-D space, a k-order Voronoi diagram partitions the 2-D space into mutually disjoint Voronoi regions. Each Voronoi region is controlled by k points, meaning that for every point in this region, the distances between this point and the k control points cannot be longer than the corresponding distances between this point and other $(n-k)$ points. A case of a 3-order Voronoi is given in Fig 2, in which the blue point is in the Voronoi region controlled by $\{q_3, q_4, q_5\}$; so the distances between this point and each point in $\{q_3, q_4, q_5\}$ is shorter than the corresponding distances between it and $\{q_1, q_2, q_6, q_7, q_8\}$.

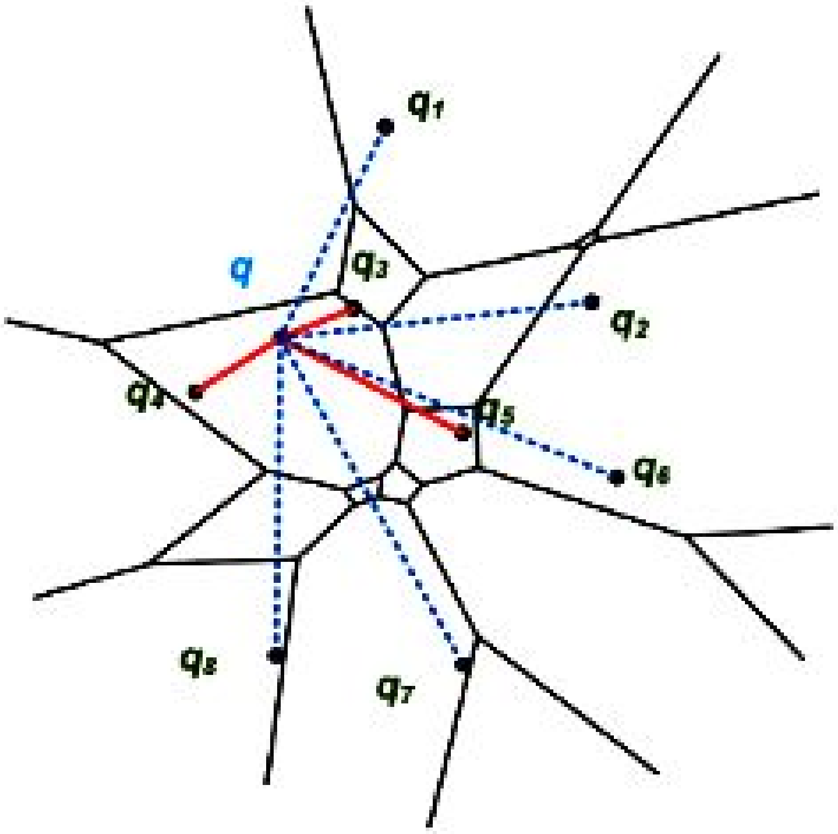

Fig. 2. An example of 3-order Voronoi diagram

It can be easily proven that the optimal k POIs with the minimum variance should control a k-order Voronoi region. Thus our proposed k-Voronoi map region search method contains 2 parts: (1) find out all k-order Voronoi regions as candidate solutions, and then (2) select the best one from them as the query answer. It can be proven that our method always find the optimal solution. In addition, existing works have shown that there are no more than $O(k(n-k))$ Voronoi regions, and the time complexity to find out all of them is $O(n^2)$[6], which ensures that our proposed method needs only $O(n^2)$ time to answer a query, which can be applied in online search services.

4 Experiments and Results

Data used in our experiment is downloaded from dianping.com, which contains location information of 13539 stores in Hangzhou, labeled with 100 distinct keywords. Fig.3 shows an example of the map with 4 different types of stores.

We use this data set to test the performance of the k-NN and the k-Voronoi map search method. In this paper, k is set to 10.

Firstly, we run both of the 2 methods on the No.1 keyword to recommend the top-100 regions separately, and the result is shown in Fig.4. We can see that the variances of regions found by k-Voronoi method are significantly lower than those found by k-NN method.

We also run the two methods to recommend the top-10 regions on all keywords. For each keyword, we calculate the variance in each region and then

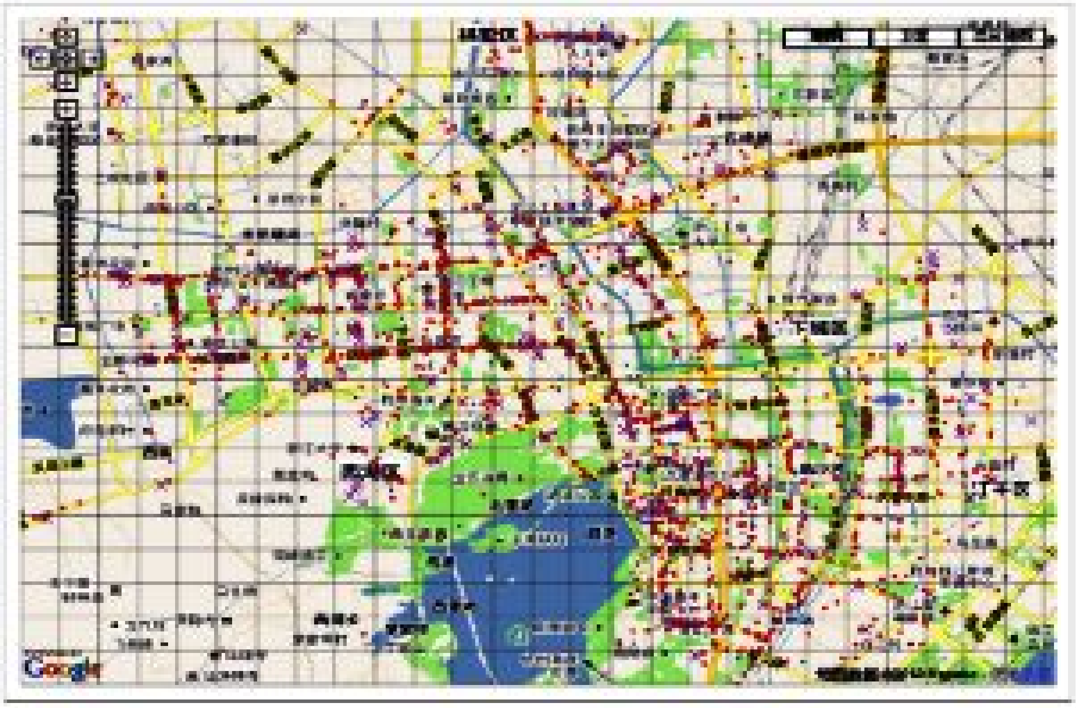

Fig. 3. Hangzhou map with 3 different types of locations

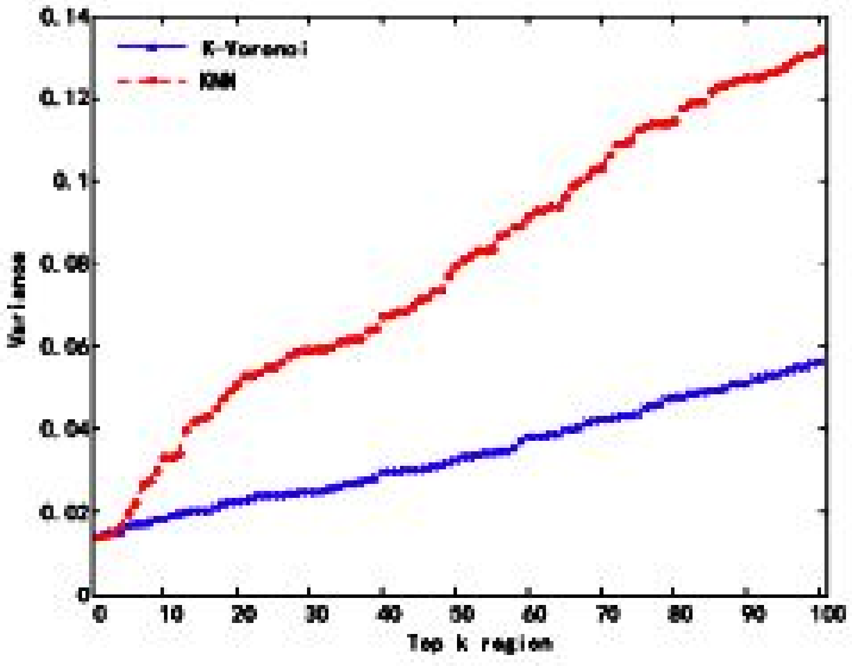

Fig. 4. Result of top-100 regions recommendation on No.1 keyword

calculate the average value of the 10 variances corresponding to the top-10 regions found by the two methods. Finally, we calculated the ratio of the two average value represented by the bars (one for each keyword) shown in Fig.5. All ratios are less than 1 and the average value of them is 0.8035 (represented by the red line in Fig.5), which indicates that our method has 20% gain over the *k*-NN method.