Journal of Information Processing Vol.22 No.2 279–288 (Apr. 2014)

[DOI: 10.2197/ipsjjip.22.279]

Regular Paper

Music Recommender Adapting Implicit Context Using 'renso' Relation among Linked Data

Mian Wang[1,a)] Takahiro Kawamura[1,2] Yuichi Sei[1] Hiroyuki Nakagawa[1]
Yasuyuki Tahara[1] Akihiko Ohsuga[1]

Received: July 8, 2013, Accepted: January 8, 2014

Abstract: The existing music recommendation systems rely on user's contexts or content analysis to satisfy the users' music playing needs. They achieved a certain degree of success and inspired future researches to get more progress. However, a cold start problem and the limitation to the similar music have been pointed out. Therefore, this paper proposes a unique recommendation method using a 'renso' alignment among Linked Data, aiming to realize the music recommendation agent in smartphone. We first collect data from Last.fm, Yahoo! Local, Twitter and LyricWiki, and create a large scale of Linked Open Data (LOD), then create the 'renso' relation on the LOD and select the music according to the context. Finally, we confirmed an evaluation result demonstrating its accuracy and serendipity.

Keywords: context awareness, music recommendation, Linked Data

1. Introduction

Since Internet was invented and exploded in the past decades, people have been used to get multimedia information (i.e., video, book and music) from Internet in private time. Especially, the music is getting a much more important aspect of their daily lives. Recently some research indicated that listening to the music is more often than any of the other activities (i.e., watching television, reading books and watching movies). Also as a powerful communication and self-expression approach, the music has been appealing a target of researches.

However, the problem now is to organize and manage millions of ever increasing music. For solving this problem a music recommendation agent comes into our view. The music recommendation agent helps users find music from a large set of music database, and some of which consistently match the user's preference. Context-aware music recommender systems (CAMRSs) have been exploring many kinds of context information [1], such as weather [2], [3], emotional state [4], running pace [5], location [6], time [7], social media activities [8] and low-level activities [9]. A music system that can detect users' context in real-time and play suitable music automatically thus could save time and effort. So far as we know, however, almost all of the existing systems need to accumulate personal information in advance, which is a time-consuming and inconvenient issue (cold start problem).

On the other hand, content-based systems [10] have come up earlier than the collaborative filtering-based systems [11]. It depends on analyzing rhythm, key, chord, tag [12] and so on. It is practical enough for the recommendation of the similar music. But along with music boom, only the similar music cannot meet consumers' appetite.

Thus, we propose a unique method based on the content-based system to avoid the above cold start problem. Also, we considered that people need more music with the diversity they may like. Serendipity [13] appeared as an important evaluation criterion several years ago. Serendipity means a "happy accident" or "pleasant surprise", the accident of finding something good or useful while not specifically searching for it. To this end, our system uses a variety of Linked Open Data (LOD) without analyzing the audio content or key, etc.

Linked Data refers to a style of publishing and interlinking structured data on the Web. The significance of Linked Data is in the fact that the value and usefulness of data increases the more it is interlinked with other data. It builds upon standard Web technologies such as HTTP, Resource Description Framework (RDF) and URIs, but rather than using them to serve web pages for human readers, it extends them to share information in a way that can be read automatically by computers. Sir Tim Berners-Lee, the inventor of the Web, pointed out that efforts for the Semantic Web are going in the right direction with the Linked Data movement. Now there are a lot of valuable data locked up in different organizations which could have been better utilized and further enhanced by joining into a single Web of data. This enables data from different sources to be connected and queried. For now, we can see various projects using Linked Data to construct their services and datasets.

Motivated by these observations, this paper presents a music recommendation method using Linked Data aiming at the avoidance of the cold start problem and the realization of the serendipity in the recommendation. This paper is organized as follows. Section 2 discusses related researches. Section 3 presents the data

[1] University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
[2] Toshiba Corporation, Kawasaki, Kanagawa 212–8582, Japan
[a)] wangmian@gmail.com

collection and their triplication to LOD which is the basis of our music recommendation. Section 4 describes how to connect the LOD and recommend the music based on the LOD. Section 5 describes an experiment and evaluation of our system and a deep analysis of our system's performance. Then, finally Section 6 concludes this paper with the future work.

2. Related Works

Most of the existing music recommendation systems that calculate with the users' preferences present methods to meet long term or short term music needs. Comparing with the users' constant preferences, users' context (such as location, mood and people around) is more changeable and flexible. For example, people who are in the library need quiet and melodious music. The existing commercial music recommendation systems such as Last.fm and Pandora are very popular and excellent, however cannot satisfy these needs very well. Meanwhile, applications for smartphones use rich sensing capabilities to collect context information in real-time and meet the needs better. Recently some researches paid more attention on CAMRSs in order to utilize contextual information and better satisfy users' needs.

Twitter Music [*1] is the brainchild of "We Are Hunted." A smartphone application pulls in music from Rdio, Spotify and iTunes, while using data from your Twitter follower graph to deliver the best possible music for you. It aims to give Twitter a mainstream launch and expand audiences of Spotify and Rdio.

EventMedia [14] is a service that targets a user who wants to relive past experiences or to attend upcoming events. It is a web-based environment that exploits real-time connections to the event and media sources to deliver rich content describing events that are associated with media, and interlinked with the Linked Data cloud. It is a pure service to satisfy daily users and developers.

SuperMusic [15] is a prototype streaming context-aware mobile music service. It is a context-based recommendation combined with collaborative filtering technologies using location and time information. The users have a chance to vote the recommended music and let the system learn their musical preference and opinion concerning song similarity. But this application cannot show the user's current context categories explicitly, many users get confused about the application's "situation" concept. For this reason, this inspired us to search for understandable context categories in the music recommendation.

Wang et al. [16] presented a probabilistic model to integrate the contextual information with the music content analysis to offer a music recommendation for daily activities. Their approach used the contextual information mainly collected with mobile devices.

Foxtrot [17] is a mobile augmented reality application with the audio content-based music recommendation. Rather than textual or visual content, audio content is potentially more emotionally engaging. Furthermore, people are allowed to share sound and music which can be tagged with a particular location. Their friends and other people can listen to shared geo-tagged music everywhere. But it is weak in automatic recommendation without geo-tagged music dataset.

Ziegler et al. [18] used the common item-based collaborative filtering algorithm to generate recommendation lists. They presented a topic diversification method to increase the diversity of a top-N list of recommendation. Besides, considering not only accuracy, they introduced a metric named intra-list similarity to capture user satisfaction and evaluate their method.

Schickel-Zuber et al. [19] presented an ontology to capture user's preference. On the basis of this ontology, they came up with a technique called Ontology Structure Similarity to derive a similarity metric. On the other hand, a recommendation system named HAAPL was designed to gain a very high accuracy and novelty.

TANGENT [20] is a recommendation algorithm considering serendipity. It is a "surprise me" query that gives a user a recommendation which is related but not regular. TANGENT treats the recommendation as a node selection on a graph, its goal is to select nodes that not only connect to nodes with high scores but also connect to unrelated ones. That method is different from our method, ours can select nodes just unrelated but surprise people.

3. Data Collection

Talking about music inevitably involves many new Web Services (i.e., Last.fm, Pandora and so on). In these services, the most valuable data are public music information and personal user information. We choose one to serve as the base of the music information. Also, in order to collect the user's context information, sensor data are needed for our system. In this part, we choose a GPS sensor to collect a GPS coordinate. Additionally, we need data to present users' implicit thoughts, and considered that Social Network is the proper source. Twitter is an online social networking service and micro-blogging service that enables its users to send and read text-based messages. That's why we choose it. Moreover, we also collect lyrics of English songs and Japanese songs from lyric websites. Thus, we collected four kinds of data such as the music information, location information, tweet and lyrics from different sources.

3.1 Last.fm

Last.fm is a music-based social networking company using a music recommender system called "Audioscrobbler." It offers events, wiki-created artist profiles and discographies and community forums. The Last.fm APIs were released several years ago, which give users the ability to build programs using Last.fm data, not only on the web, but also desktop and mobile devices. By using its RESTful API, developers can read and write access to the full slate of last.fm music data resources - albums, artists, playlists, events, users and more. We use this mainly to gather artist, track and user information.

To keep the coherence of all the dates, we chose six methods listed below.

- artist.getTopTracks: Get the top tracks by an artist on Last.fm, ordered by popularity.
- geo.getTopArtists: Get the most popular artists on Last.fm by country. It was restricted to Japan for this research.
- track.getInfo: Get the metadata for a track on Last.fm using

[*1] https://music.twitter.com/

Table 1 Part of industry codes.

業種コード 1 Industry Code 1	業種コード 2 Industry Code 2	業種コード 3 Industry Code 3	業種名 1 Industry Name 1	業種名 2 Industry Name 2	業種名 3 Industry Name 3
1	101	101001	グルメ Gourmet	和食 Japanese Food	懐石料理 Kaiseki Cuisine
1	101	101002	グルメ Gourmet	和食 Japanese Food	会席料理 Banquet
1	101	101003	グルメ Gourmet	和食 Japanese Food	割ぽう Japanese-style Cooking
1	101	101004	グルメ Gourmet	和食 Japanese Food	料亭 High-class Restaurant
1	101	101005	グルメ Gourmet	和食 Japanese Food	小料理 Snack
1	101	101006	グルメ Gourmet	和食 Japanese Food	精進料理 Vegetarian Cooking
1	101	101007	グルメ Gourmet	和食 Japanese Food	京料理 Kyoto Cuisine
1	101	101008	グルメ Gourmet	和食 Japanese Food	豆腐料理 Tofu Dish
1	101	101009	グルメ Gourmet	和食 Japanese Food	ゆば料理 Yuba Cuisine

the artist/track name or a musicbrainz id. We acquired artist/track name and track release year through this method.

- user.getInfo: Get information about a user profile. Used to catch a user's age.
- user.getRecentTracks: Get a list of the recent tracks listened to by this user.
- user.getRecommendedArtists: Get Last.fm artist recommendations for a user.

3.2 Yahoo! Local

Yahoo! is widely known for its web portal, search engine, and related services. And several years ago, it released Open Local Platform API to developers in Japan region. Yahoo! Open Local Platform (YOLP) is the API & SDK of the geographical map and the local information that Yahoo! JAPAN provides to the developers. You can take advantage of the rich features of the multifold map display, store and facility search, geocoding, route search, and land elevation acquisition in the development of web page and applications for smartphones.

We use Yahoo! Place Info API that one of the most useful APIs for searching region-based store information, events, reviews and other information (POI). This API can translate geographical coordinates into industry code. Part of the industry codes is presented in **Table 1**. We mainly use this API to do the translation and its request and response are just like showed below in **Table 2**.

Table 2 Example of request (upper) and response (lower) through Yahoo! Local Search API.

```
http://placeinfo.olp.yahooapis.jp/V1/get?lat=35.66521320007564&lon=139.7300114513391&appid=＜あなたのアプリケーション ID＞&output=json
                                                    Your Application ID
```

```
{
  "ResultSet": {
    "Result": [
      {                                  Around Tokyo MidTown
        "Combined": "外苑東通りの東京ミッドタウン付近",
        "Uid": "29586a2bea6bcea0f0934695959d389404b04269",
        "Score": 87.9099807710434,
        "Label": "東京ミッドタウン",    Tokyo MidTown
        "Name": "東京ミッドタウン",    Tokyo MidTown
        "Where": "外苑東通り",    Meijijingu Gaien
        "Category": "ショッピングセンター・モール、複合商業施設"
                                  Shopping Center, Mall, Commercial Institution
      },
```

Table 3 Example of MeCab analysis.

```
>この研究は面白い          This research is interesting
この    This    連体詞,*,*,*,*,*,この,コノ,コノ
研究    research    名詞,サ変接続,*,*,*,*,研究,ケンキュウ,ケンキュー
は      is  助詞,係助詞,*,*,*,*,は,ハ,ワ
面白い  interesting 形容詞,自立,*,*,形容詞・アウオ段,基本形,面白い,
        オモシロイ,オモシロイ
EOS
```

By using this service, the developers can catch the location information more particular than the industry code and name, specific to a particular location.

3.3 Twitter

User-generated content on Twitter (produced at an enormous rate of 340 million tweets per day) provides a rich source for gleaning people's thoughts, which is necessary for deeper understanding of people's behaviors and actions.

Therefore, we collected a quantity of tweets relevant to industry names from Yahoo! Local Search API. While the sum of industry is 584, input each name as a keyword, through the Twitter API to acquire the related data. But a problem we have to solve is the language classification. People all around the world are using Twitter, and they send messages in English and many other languages. So in this paper we focused on Japanese tweets only.

MeCab is a fast and customizable Japanese morphological analyzer. It is designed for generic purpose and applied to a variety of NLP tasks, such as Kana-Kanji conversion. It provides parameter estimation functionalities based on CRFs and HMM. MeCab used in our system has two tasks. One is morphological analysis, and the other is keyword extraction using a TF-IDF algorithm. In morphological analysis, it analyzes all the tweets sentence by

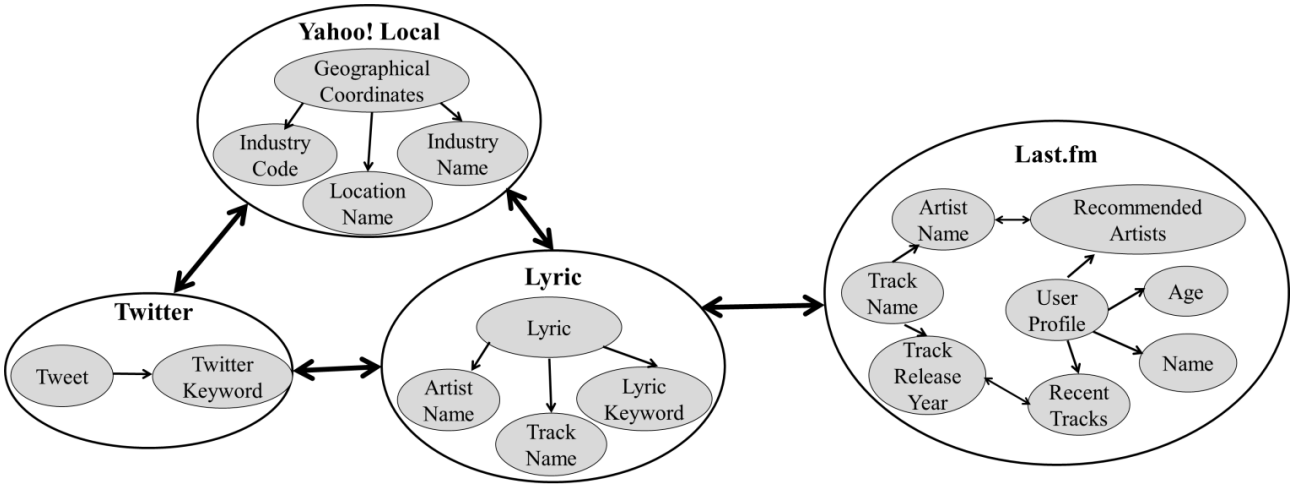

Fig. 1 Data relationship.

sentence, and returns parts of a speech. Here is an example in **Table 3**. MeCab identifies a sentence into a noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. After the morphological analysis, all parts of the speech are collected, we count times each occurs and their total number. Then use this simple but effective algorithm TF-IDF.

3.4 LyricWiki

Lyric information is very important for identifying the music, since there are no two songs that have the same lyrics. Hence, we build an environment to catch up English and Japanese music lyrics. For the English music, we receive the lyric data from LyricWiki. LyricWiki is a community-powered online Lyrics database containing over 1,800,000 lyric pages. Users on the site can view, edit, and discuss the lyrics of songs, which are also available for purchase from links on the site. It has a very high credibility just like Wikipedia. LyricWiki released its API in 2008. The LyricWiki API lets you search for songs, artists, and albums. You can get the lyrics for a specific song; get an artist's discography; or get a track listing for a specific album. You can also post new information into the LyricWiki archive like a new artist, an album, and song lyrics. While in Japanese music, things are more complex. It is impossible to fetch lyric data from the website directly. After a huge amount of effort in intelligence gathering, we found an intermediate tool called Lyrics Master, which is a lyric software made by Kenichi Maehashi. It provides lyric search and download services from over 20 lyric websites. Not only Japanese pop music, but also western and other genre music is supported.

Lyrics Master has an external call function through which anyone can write a bash script or program by passing parameters to search and download lyrics. It is necessary to pay attention to the request rate because lyrics websites all set up a request limit.

All the four kinds of data were collected and their relations are presented in **Fig. 1**.

3.5 Linked Data Conversion

Once collected, data is organized in different formats. Before converting into Linked Data in RDF, defining a schema is indispensable. We aim to make RDF triples by providing descriptions of relevance using the Music Ontology [21] and a large SKOS taxonomy [22]. The detailed structure is presented in Fig. 1. From last.fm, the user ID, recent tracks, the user profile info, the artist name, the track name, and the recommended artist are gathered and arranged with Music Ontology. While from Yahoo!, we grab geographical coordinates in real-time, link industry code and genre keyword with the SKOS taxonomy. Twitter keyword and genre keyword extracted from the tweets are associated with the SKOS taxonomy. At last, the lyric information containing the artist name, the track name, the lyric keyword is defined in the Music Ontology.

After collecting data and defining a schema, we use an online service [*2] to convert table data into an RDF file. This service is called LinkData, which is very easy for beginners and experts to use, aiming at the dissemination of open data by offering an online service for anyone to create Linked Open Data and publish it. The main difference with any other RDF-Converter is a simple template-based system combined with a data hosting service. Each dataset is converted to an RDF file and uploaded to the LinkData site. It is set to public and anyone can access and use them. By using this service, we generated several RDF/Turtle files. But these files are full of redundancies, hence some post-processing is done to fix the problem.

4. Proposal of 'renso' Alignment of LOD

In the previous section, we created the four kinds of LOD. However, they are almost isolated and the links among different datasets are few, although we created the '`owl:sameAs`' links if the nodes between the datasets are identical. This is a well-known instance matching issue in the area of ontology alignment.

Ontology alignment [23] means determining corresponding nodes between ontologies. We can define different types of (inter-ontology) relationships among their nodes. Such relationships will be called alignments. There are several matching methods like predicating about the similarity of ontology terms, and mappings of logical axioms, typically expressing the logical

[*2] http://linkdata.org/

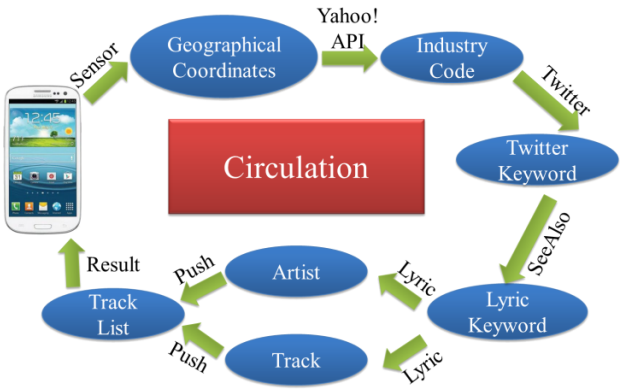

Fig. 2 Process of alignment 1.

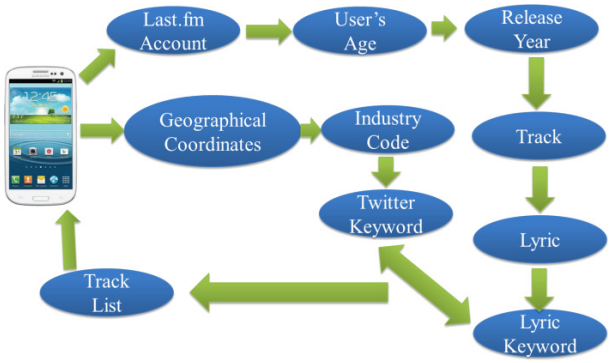

Fig. 3 Process of alignment 2.

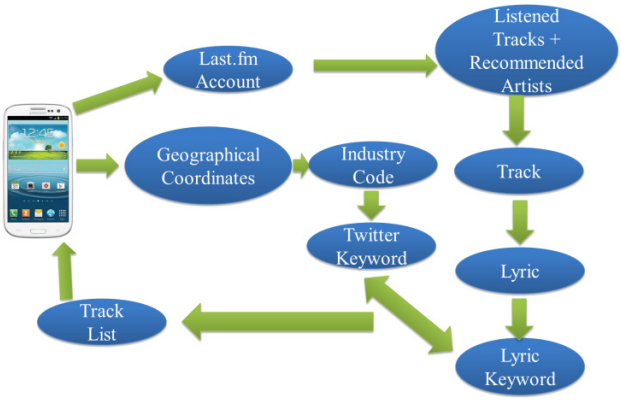

Fig. 4 Process of alignment 3.

equivalence or inclusion among ontology terms.

Although we use the basic instance matching methods such as the string similarity using N-gram and the semantic similarity using Japanese Word Net, the expression of a term may vary especially in Twitter, and the semantics are richer than the Word Net especially in the lyrics. Therefore, we propose a new relation 'renso,' that means n hops of implicit associations, to create more connections among the LOD. For example, we can say there is a 'renso' relation between "Cherry blossoms" and "Graduation" in Japan. Or, a Japanese proverb says "if the wind blows the bucket makers prosper," which means any event can bring about an effect in an unexpected way. So by tracing these 'renso' links in the LOD we intend to recommend the music with the serendipity. More precisely, given two data sources A_1 and A_n as input, the goal of 'renso' alignment is to compute the set $M = \{(a_1, a_n) | (a_1, a_n) \in A_1 \times A_n,\ (a_1, a_n) \in R_a\}$. Here, $(a_1, a_n) \in R_a$ means a pair of distinct instances a_1 and a_n has a 'renso' relation if they are part of (any kind of) n-ary relation $R \subseteq A_1 \times \ldots \times A_n$.

Based on the experiments, we defined three types of the 'renso' alignments between different LOD for the music recommendation. In every alignment the number of hops is static. In alignment 1, n is equal to 4, while in alignment 2 it is equal to 9 and in alignment 3 it is equal to 8.

The first one is the simplest query method. It mainly connects twitter keywords and lyric keywords, and presented in **Fig. 2**. In the case that the user who has a smartphone in hand is using our system, firstly the GPS sensor in the smartphone locates the user's situation and returns a pair of coordinate value. We collect this coordinate and convert it into the industry code through Yahoo! Place Info API, then get a return value like "美術館" (Art Museum), this is the first hop. We then use "美術館" (Art Museum) as a keyword to search tweets through Twitter API, and obtain several twitter keywords with a strong relevance to "美術館" (Art Museum), which is the second hop, then count the number of the keywords that appeared in each song lyric, that is the third hop. Finally we sort the songs with the number, and get No.1 song which in the example has three same keywords with the Twitter keywords called "眼鏡越しの空" (Sky over glasses), this is the last hop. The same keywords are "写真" (Photograph), "夢" (Dream) and "眼鏡" (Glass). As a result, the 'renso' relation between the location of the user and the resulted song has been created. These relations can be found and created on the fly and stored in the LOD for future use. In the recommendation, we can search on the graph of LOD by tracing the 'renso' relations, and find the song using SPARQL querying methods.

Alignment 2 is on the basis of the above alignment 1, adding a user's profile information. When querying, we set the user's profile as a filter. Alignment 2 is presented in **Fig. 3**. We assume the user was born around 90's and now more than 20 years old. After getting that the user is in a cramming school through YOLP, a list of songs with strong relevance is generated in intermediate stage. Considering the user's age, songs released from year $2013 - 20 = 1993$ to year 2013 are selected and sorted with the number of the keywords that appeared in each song lyric. Finally we get a song named "sympathizer" released in 2009. The same keywords are "希望" (Hope), "魂" (Soul) and "先" (Destination). Also the 'renso' relation between the location of the user and the resulted song has been created.

Similar to alignment 2, alignment 3 uses the user's listening history and the recommended artist information as a filter, presented in **Fig. 4**. We assume the user of our system prefers J-POP, J-ROCK music; favorite artists are "YUI," "HIGH and MIGHTY COLOR" and so on. GPS sensor and Yahoo! API show that he/she is in "野球場" (Baseball Ground), keyword matching returns a list, and Last.fm recommends him/her several artists he/she may like. Through this cross matching, the most suitable song filtered is "君の名は希望" (Your name is HOPE) with keywords "希望" (Hope) "想像" (Imagine) and "雨" (Rain).

Definitions of the alignments are as follows. Considering that there is an infinite set U (RDF URI references), and an infinite

Algorithm 1 Define Some Functions

```
function SEARCH(x, G1)                          ▷ Search x in graph G1
    for all g ∈ G1 do
        if g = x then
            Push g into array Result
        end if
    end for
return Result
end function

function MATCH(X, Y)                            ▷ Match between X and Y
    for all x ∈ X do
        Result ← Result ∪ SEARCH(x, Y)
    end for
    Calculate times each element appeared
    return Result
end function

function SORT(M)                                ▷ Sort by numeric value
    for i ← 1, length(M) do
        for j ← length(M), i + 1 do
            if thenM_j < M_{j−1}
                Exchange M_j and M_{j−1}
            end if
        end for
    end for
end function
```

set L (literals). A triple $(s, p, o) \in (U \cup L) \times U \times (U \cup L)$ is called an RDF triple (s is called the subject, p the predicate and o the object). An RDF Knowledge Base (KB) K, or equivalently an RDF graph G, is a set of RDF triples.

For an RDF Graph G, we shall use U, L to denote the URIs, literals that appear in the triples of G respectively. The nodes of G are the values that appear as subjects or objects in the triples of G. The main graph G contains four datasets collected separately: F is from Last.fm, Y is from Yahoo Local, T is from Twitter and L is from Lyric.

The algorithm described above can be expressed in the following pseudo-code. First, we define the following additional functions as proposed in Algorithm 1.

Function Search is a function to search a value in a graph. While function Match is to find common elements between two graphs, function Sort is to sort an array by the numeric value.

Our starting point is geographical *coordinates(x,y)* received from a user's smartphone, while the ending point is a music playlist. Our mapping pseudo-code is defined as Algorithm 2. It describes three alignments' implementation procedures. The three alignments are designed to use less profile information in order to avoid the cold start problem. Meanwhile, being independent of a user profile leads to high serendipity. What we try hard to do is enhance accuracy.

5. Music Recommendation Experiment

In this section we describe the results of our recommendation system illustrated in **Fig. 5**. Adequacy of the 'renso' relation highly depends on the user's feeling, and is difficult to be measured in an objective way. Instead, we evaluate the accuracy of the music recommendation according to the purpose of this paper. We have conducted an evaluation of accuracy and serendipity, and the results demonstrate a significant promise from both perspectives.

5.1 Dataset

All the data were collected for two weeks; their details are presented in **Table 4**. First on is the industry code which contains three levels from top to down. The first level is divided into "グルメ" (Gourmet), "ショッピング" (Shopping), "レジャー" (Leisure) and "暮らし" (Life). Each has sublevels. The YOLP sets out 584 separate codes altogether, covering a wide range of industries and fields.

Based on the industry codes, we defined 584 location keywords. By using these location keywords, we employed Twitter

Algorithm 2 Three Alignments

```
procedure BASE
    coordinates(x, y) ← GPSsensor
    category ← SEARCH(coordinates(x, y), Yahoo!API)
    categorykeyword ← SEARCH(category, Y)
    TwitterKeyword ← SEARCH(categorykeyword, T)
    for all twitterkeyword ∈ TwitterKeyword do
        Track(TK) ← Track(TK) ∪ SEARCH(twitterkeyword, L)
    end for
    return Track(TK)
end procedure

main Alignment 1
Start
    procedure BASE
    SORT(Track) by LK
    for count ← 0, 2 ∈ Track do
        Track(Info) ← Track(Info) ∪ SEARCH(count, F)
    end for
    return Track(Info)
End

main Alignment 2
Start
    User(age) ← SEARCH(user, Last.fm)
    Now(year) ← getYear
    for year ← Now(year)−Age(user), Now(year) do
        YearTrack ← YearTrack ∪ SEARCH(year, F)
    end for
    for all yeartrack ∈ YearTrack do
        Track(Lyric) ← Track(Lyric) ∪ SEARCH(yeartrack, L)
    end for
    for all tracklyric ∈ Track(Lyric) do
        Track(LK) ← Track(LK) ∪ SEARCH(tracklyric, L)
    end for
    procedure BASE
    Count ← MATCH(Track(TK), Track(LK))
    SORT(Track) by Count
    for count ← 0, 2 ∈ Track do
        Track(Info) ← Track(Info) ∪ SEARCH(count, F)
    end for
    return Track(Info)
End

main Alignment 3
Start
    User(info) ← SEARCH(user, Last.fm)
    User(SelectedTrack) ← User(ListenedTrack) + User(RecommenedTrack)
    for all selectedtrack ∈ User(SelectedTrack) do
        Track(Lyric) ← Track(Lyric) ∪ SEARCH(selectedtrack, L)
    end for
    for all tracklyric ∈ Track(Lyric) do
        Track(LK) ← Track(LK) ∪ SEARCH(tracklyric, L)
    end for
    procedure BASE
    Count ← MATCH(Track(TK), Track(LK))
    SORT(Track) by Count
    for count ← 0, 2 ∈ Track do
        Track(Info) ← Track(Info) ∪ SEARCH(count, F)
    end for
    return Track(Info)
End
```

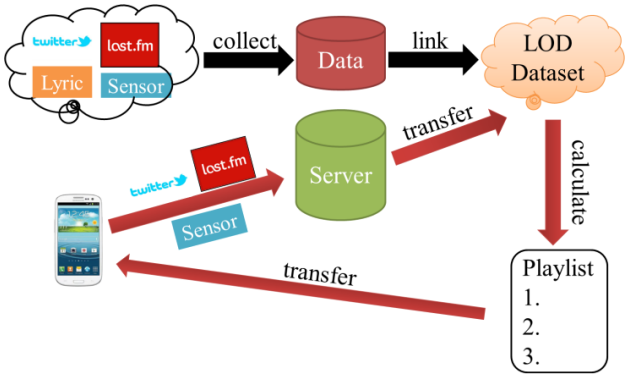

Fig. 5 System approach.

Table 4 Data details.

Category	Industry Code	Tweet	Lyric	Last.fm
Triples	584	584,000	400,000	20,050

Table 5 Example of questionnaire item.

Description	Question 1	Question 2
When you are in (location name) right now, our system recommends a track which has no direct connection with the library but contains (keyword) and (keyword) in its lyric named (song name).	Do you think this recommendation is unexpected but fun?	Do you think this recommendation is fit for this place?

as a platform to search tweets. Every location keyword corresponds to 1,000 tweets. Then we utilized MeCab to extract 50 twitter keywords by analyzing every 1,000 tweets. Also, Last.fm stores many aspects of information. All the data were collected through its API methods. We set Japan as the main place for experiments, requested Top 500 artists in Japan and their Top 20 tracks. Afterwards, I used my Last.fm account to acquire any personal information for alignment 2 and alignment 3. Every track corresponds to one lyric data. As well as the tweets, we processed the lyric data with MeCab and extracted 40 lyric keywords per song.

5.2 Experiment Setting

We conducted an evaluation to identify the accuracy and the serendipity. We invited 20 test users to participate in this experiment. Six of them are our college students, the others are working adults. Also, three of them are Japanese, the others are not. We designed a questionnaire and the example is presented in **Table 5**. Question 1 is asking for the serendipity, while question 2 is for the accuracy. To start with, we explained these two indicators to the test users. In the case of "serendipity", we defined it as "a track that is interesting from a different point of view regardless of familiarity" and gave the test users an 'Evangelion' example. We explained "accuracy" by giving the test users another example "When you are in a club, dance music is fit for this place". We wrote 10 items with form in Google Drive. It was set to public for everyone here [*3].

The result of the experiment is presented in **Fig. 6**. We can see that all three alignments achieved about 60% in the accuracy and the serendipity. Among them, alignment 3 presents the highest value; both evaluating indicators are above 65%. While in **Fig. 7** that shows the users' favorite alignment, it is alignment 1. Overall, although alignment 3 shows the best performance, the test users prefer the one that uses the most simplest but straight recommendation method.

[*3] https://docs.google.com/forms/d/1VGHP6OuSJqBxIo3F7pvSJv5LNqZvtZbVHXmI24982fQ/viewform

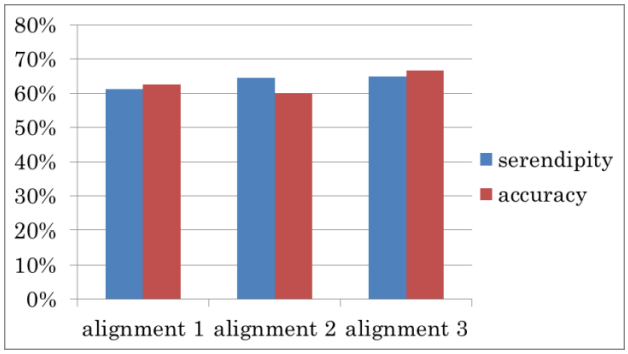

Fig. 6 The result of experiment.

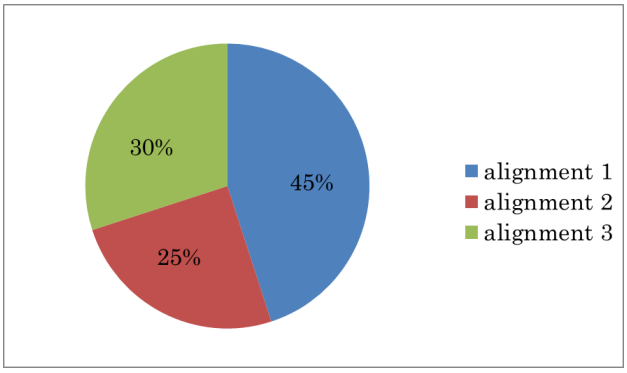

Fig. 7 The result of favorite alignment.

5.3 Lesson Learnt

Through this experiment, we found three key points. First one is about the data relevance. Twitter is a platform that the people use when they want to express their emotions, while the lyric is a transfer mediator for the people from who composed it. They all include implicit thoughts. But implicit thoughts analysis is very difficult just like the emotion extraction. Thus, there is no guarantee of the accuracy. Even so, along with the increase of the relation path's length in the LOD, we could find more implicit connections to reach a higher accuracy and serendipity. Also, we confirmed there is no cold start problem, since our base is the large set of LOD.

On the other hand, meeting both the accuracy and the serendipity at a time is an arduous task. As it is stated in Ref. [24], the more we focus on increasing the accuracy by yielding a catalog coverage, the less serendipity happens. Even a "perfect" content-based recommender could not find surprising items. Our research is for getting the users' implicit context, but the emotion is more complicated and changeable than any other. As a summary of our evaluation, only one of two criteria can be satisfied in most of the cases. However, unlike the conventional content-based method, we confirmed our system finds it easier to achieve the serendipity in the recommendation. For a small instance, in our university there are so many students who love SF and robot animation, and are especially crazy about "Evangelion" (a Japanese media franchise). Consequently, "University of Electro-Communications" comes along with "Evangelion" frequently on Twitter. For this reason, we recommend "Evangelion" theme song for the user who is in our university. It is full of fun, but the accuracy is subtle. Or to put it another way, it is difficult enough to satisfy everyone.

The last point is about statistically-significant differences. We made an additional questionnaire after the main experiment. It contains questions about locations and music genres for a deeper analysis and future work.

From the viewpoint of the location, we found users in different types of locations are satisfied with different types of music. Yahoo! Local API classifies locations into four types: Gourmet, Shopping, Leisure and Life. And we also classified the music into 8 types: Pop, Rock, Jazz, Classic, Hip-Hop, Folk, New Age and Anime. According to the experiment, when people in locations belong to the category of Gourmet, 66.7% of them prefer listening to Classic music. While in the other three types of locations, 100%, 37.5% and 50% of them prefer Pop music, Anime music and Classic music respectively. For example, people in Japanese restaurants have been satisfied with Classic music or no music. On the other hand, in French restaurants they have been satisfied with Jazz music and Classic music.

From the viewpoint of the genre of the music, we can find that people appreciates quiet and familiar music more, no matter where they are. In our experiment, Rock music, Hip-Hop music, Folk music and New Age music presented a low satisfaction as 31.0%. The term "satisfaction" here is different from the accuracy and the serendipity, simply representing users' preference on the recommended track. The other four genres of music each have their different atmosphere: Classic music means relax and slow speed, Pop music means passion and motion, Jazz music means elegance and romance, Anime music means enjoyment and relax. From the experimental results, we received about the people's preferences in various situations, the above overall trend is reflected clearly.

Through this additional questionnaire, we got rather useful feedbacks for developing a more practical system and a more valuable algorithm. Using genre information to build a subsequent version could be our next work.

5.4 Performance Evaluation

For examining how the degree of 'renso' affects the performance of the proposed method by changing the number of tweets, we measured time consumed in case of 100, 1,000 and 10,000 tweets in alignment 1.

For this comparison, we devided the 'renso' recommendation process into separate parts among which 'collect tweets,' 'MeCab analysis tweets' (using MeCab to do Japanese morphological analysis and keyword extraction) and 'twitter keyword renso' (searching for twitter keywords and lyric keywords with strong relevance to a specific situation, and sorting the keywords on both collections with the number) are the three parts that affect the process speed. The details are presented in **Table 6**.

Through this experiment, we found that the one which affects the performance most is 'Collect tweets', because network communication with Twitter Server consumed too much time. 'MeCab analysis tweets' is a procedure without network communication, so it did not cost much time. At last, since the number of tweets did not affect the number of twitter keyword, the processing time of 'Twitter keyword renso' was in 0.1 second in all cases.

Table 6 Comparison of the time consumption.

	100 Tweets	1,000 Tweets	10,000 Tweets
Collect tweets	1.936s	22.157s	278.370s
MeCab analysis tweets	0.240s	0.570s	3.650s
Twitter keyword renso	0.074s	0.080s	0.076s

Table 7 Comparison of accuracy and serendipity.

	1,000 Tweets Dataset	100 Tweets Dataset
Alignment 1 (Serendipity)	0.875	0.563
Alignment 1 (Accuracy)	0.438	0.625
Alignment 2 (Serendipity)	0.667	0.417
Alignment 2 (Accuracy)	0.500	0.667
Alignment 3 (Serendipity)	0.583	0.750
Alignment 3 (Accuracy)	0.500	1.000

In our system, all the tweets were collected before the experiment except users' recent tweets, and the quantity of recent tweets is limited to 100. On the basis of the result above, the collection procedure costs about 2 seconds. Thus, it can be concluded that the number of tweets affects the performance, but in our system the effect is very little.

Furthermore, for examining how the degree of 'renso' affects the accuracy and the serendipity of our proposed system, we prepared another dataset. This dataset contains 100 tweets every location keyword, while the original one contains 1,000 tweets. We did this experiment on the condition of not telling our test users the difference of these two datasets. For this comparison, we chose four people from former 20 test users. They were asked to reevaluate the 3 alignments by this new dataset and recommended tracks, especially items evaluated not accurate or with no serendipity in the former experiment. All the test users' feedbacks indicated that the two datasets have distinct differences. As a result, in comparison with the 100 tweets dataset, the 1,000 tweets dataset has a lower accuracy and a higher serendipity. We thought the reason is that more tweets can bring in more implicit relations which lead to the higher serendipity. Since the accuracy and the serendipity cannot both reach high scores as mentioned before, this resulted in the lower accuracy of the 1,000 tweets dataset. Only the serendipity in alignment 3 seemed to be an exception. The details are presented in **Table 7**.

6. Conclusion

In this paper we presented a music recommendation method

using the 'renso' alignment that connects various kinds of LOD, aiming at the avoidance of the cold start problem and the realization of the serendipity in the recommendation.

We collected four datasets: 20,050 triples from Last.fm, 584 triples from Yahoo! Local, 584,000 triples from Twitter and 400,000 triples from Lyric. They are public in LinkData and developers who want to carry out the related research can reuse them. Based on the datasets and three different alignments, we found that there really is a relation between lyric and tweet in expressing people's implicit thoughts.

As the future work, we plan to include more data sources and design more alignments. Not only from Twitter and Lyric, we will discover the people's implicit thoughts from other online resources, and design a more reasonable and effective alignment. Then, we will develop the music recommendation agent for the smartphone in the near future. Furthermore, this recommendation method using the user's implicit thought, that is, the 'renso' alignment could be used for another media like a movie, fashion items and so forth. So we will consider the adaptation of this method to other domains.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 24300005, 23500039, 25730038 from Graduate School of Information Systems at University of Electro-Communications. We would like to thank Professor Shinichi Honiden in National Institute of Informatics/University of Tokyo and his group for offering a place for discussing, studying and providing instructions.

References

[1] Ricci, F.: Context-Aware Music Recommender Systems, *AdMIRe'12 Workshop*, pp.865–866 (2012).
[2] Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S. and Seada, K.: Fusing Mobile, Sensor, and Social Data To Fully Enable Context-Aware Computing, *HOTMOBILE 2010*, pp.60–65 (2010).
[3] Reynolds, G., Barry, D., Burke, T. and Coyle, E.: Interacting with large music collections: Towards the use of environmental metadata, *IEEE International Conference on Multimedia and Expo*, pp.989–992 (2008).
[4] Komori, M., Matsumura, N., Miura, A. and Nagaoka, C.: Relationships between periodic behaviors in micro-blogging and the users' baseline mood, *SNPD 2012*, pp.405–410 (2012).
[5] Elliott, G.T. and Tomlinson, B.: PersonalSoundtrack: context-aware playlists that adapt to user pace, *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pp.736–741 (2006).
[6] Kaminskas, M. and Ricci, F.: Location-Adapted Music Recommendation Using Tags, *UMAP 2011 LNCS*, Vol.6787, pp.183–194 (2011).
[7] Cebrian, T., Planaguma, M., Villegas, P. and Amatriain, X.: Music Recommendations with Temporal Context Awareness, *RecSys'10*, pp.349–352 (2010).
[8] Bu, J., Tan, S., Chen, C., Wang, C., Wu, H., Zhang, L. and He, X.: Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content, *MM'10*, pp.391–400 (2012).
[9] North, A.C., Hargreaves, D.J. and Hargreaves, J.J.: Uses of Music in Everyday Life, *Music Perception: An Interdisciplinary Journal*, Vol.22, No.1, pp.41–77 (2004).
[10] Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C. and Slaney, M.: Content-Based Music Information Retrieval: Current Directions and Future Challenges, *Proc. IEEE*, Vol.96, No.4, pp.668–696 (2008).
[11] Aizenberg, N., Koren, Y. and Somekh, O.: Build Your Own Music Recommender by Modeling Internet Radio Streams, *WWW 2012*, pp.1–10 (2012).
[12] Wang, J.-C., Shih, Y.-C., Wu, M.-S., Wang, H.-M. and Jeng, S.-K.: Colorizing Tags in Tag Cloud: A Novel Query-by-Tag Music Search System, *MM'11*, pp.293–302 (2011).
[13] Zhang, Y.C., Seaghdha, D.O., Quercia, D. and Jambor, T.: Auralist: Introducing Serendipity into Music Recommendation, *WSDM'12*, pp.13–22 (2012).
[14] Khrouf, H., Milicic, V. and Troncy, R.: EventMedia Live: Exploring Events Connections in Real-Time to Enhance Content, *ISWC 2012* (2012).
[15] Lehtiniemi, A.: Evaluating SuperMusic: Streaming context-aware mobile music service, *Proc. 2008 International Conference on Advances in Computer Entertainment Technology*, pp.314–321 (2008).
[16] Wang, X., Rosenblum, D. and Wang, Y.: Context-Aware Mobile Music Recommendation for Daily Activities, *MM'12*, pp.99–108 (2012).
[17] Ankolekar, A. and Sandholm, T.: Foxtrot: A Soundtrack for Where You Are, *IWS'11*, pp.26–31 (2011).
[18] Ziegler, C.-N., McNee, S.M., Konstan, J.A. and Lausen, G.: Improving Recommendation Lists Through Topic Diversification, *WWW 2005*, pp.22–32 (2005).
[19] Schickel-Zuber, V. and Faltings, B.: Inferring User's Preferences using Ontologies, *AAAI'06*, pp.1413–1418 (2006).
[20] Onuma, K., Tong, H. and Faloutsos, C.: TANGENT: A novel, 'Surprise me', recommendation algorithm, *KDD'09*, pp.657–666 (2009).
[21] Music Ontology, available from ⟨http://musicontology.com/⟩.
[22] SKOS Simple Knowledge Organization System, available from ⟨http://www.w3.org/2004/02/skos/⟩.
[23] Ontology Alignment Evaluation Initiative, available from ⟨http://oaei.ontologymatching.org/⟩ (2008).
[24] Ge, M., Battenfeld, C.D. and Jannach, D.: Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity, *Proc. 4th ACM Conference on Recommender Systems, RecSys'10*, pp.257–260 (2010).

Mian Wang is a Master's student at University of Electro-Communications. He received a B.S. degree in Information Science and Technology from Northwest University, China in 2008. His research interests include music analysis, semantic web and open data.

Takahiro Kawamura is a Senior Research Scientist at Corporate Research & Development Center, Toshiba Corp., and also an Associate Professor in the Graduate School of Information Systems, University of Electro-Communications, a Lecturer in the Graduate School of Engineering, Osaka University, Japan. He received his Ph.D. degree in Computer Science from Waseda University in 2001. From 2001 to 2002 he was a Visiting Scholar in Robotics Institute, Carnegie Mellon University. His current research interests include semantic web, open data, and software agent.

Yuichi Sei received his Ph.D. degree in Information Science and Technology, from the University of Tokyo, Japan, in 2009. From 2009 to 2012 he was working at Mitsubishi Research Institute. Since 2013, he has been an Assistant Professor at University of Electro-Communications. His research interests include pervasive computing, security, and privacy-preserving data mining.

Hiroyuki Nakagawa is an Associated Professor with the Graduate School of Information Science and Technology, Osaka University. He received his B.S. degree in Computer Science from Osaka University in 1997, his M.S. degree in Computer Science from the University of Tokyo in 2007, and his Ph.D. degree in Computer Science from Waseda University in 2013. He worked in Kajima Corporation from 1997 to 2008. He was an Assistant Professor in University of Electro-Communications from 2008 to 2013. His research interests include requirements engineering, self-adaptive systems, and agent technology. He is a member of IEEE Computer Society, IPSJ, and IEICE.

Yasuyuki Tahara is an Associate Professor in University of Electro-Communications (UEC). He received his B.Sc. and M.Sc. in Mathematics from the University of Tokyo, Japan, and his Ph.D. in Information and Computer Science from Waseda University, Japan, in 1989, 1991, and 2003, respectively. He joined Toshiba Corporation in 1991. He was a Visiting Researcher in City University London, UK, from 1995 to 1996, and in Imperial College London, UK, from 1996 to 1997. He left Toshiba Corporation and joined National Institute of Informatics (NII) in 2003. He left NII and joined UEC in 2008. His research interests include formal verification of software and requirements engineering. Professor Tahara is a member of IPSJ and Japan Society for Software Science and Technology.

Akihiko Ohsuga received a B.S. degree in Mathematics from Sophia University in 1981 and a Ph.D. degree in Computer Science from Waseda University in 1995. From 1981 to 2007 he worked with Toshiba Corporation. Since April 2007, he has been a Professor in the Graduate School of Information Systems, University of Electro-Communications. He is a member of the IEEE Computer Society, IPSJ, IEICE, The Japanese Society for Artificial Intelligence, and Japan Society for Software Science and Technology. He received the 1986 Paper Award from IPSJ.