Context-Aware Music Recommendation with Serendipity Using Semantic Relations

Mian Wang(✉), Takahiro Kawamura, Yuichi Sei, Hiroyuki Nakagawa, Yasuyuki Tahara, and Akihiko Ohsuga

The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo, Japan
wangmian30@gmail.com, kawamura@ohsuga.is.uec.ac.jp,
{sei,nakagawa,tahara,ohsuga}@uec.ac.jp
http://www.ohsuga.is.uec.ac.jp/

Abstract. A goal for the creation and improvement of music recommendation is to retrieve users' preferences and select the music adapting to the preferences. Although the existing researches achieved a certain degree of success and inspired future researches to get more progress, problem of the cold start recommendation and the limitation to the similar music have been pointed out. Hence we incorporate concept of serendipity using 'renso' alignments over Linked Data to satisfy the users' music playing needs. We first collect music-related data from Last.fm, Yahoo! Local, Twitter and LyricWiki, and then create the 'renso' relation on the Music Linked Data. Our system proposes a way of finding suitable but novel music according to the users' contexts. Finally, preliminary experiments confirm balance of accuracy and serendipity of the music recommendation.

Keywords: Recommendation system · Context awareness · 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 much more important aspect of our 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 music released unstoppable. For solving this problem music recommendation system comes into our view. The music recommendation system 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

W. Kim et al. (Eds.): JIST 2013, LNCS 8388, pp. 17–32, 2014.
DOI: 10.1007/978-3-319-06826-8_2,

been exploring many kinds of context information [1], such as weather [2], emotional state [3], running pace [4], location [5], time [6], social media activities [7] and low-level activities [8]. 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, recent music recommendation systems are 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 [9] 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 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, RDF (Resource Description Framework) 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. For now, we can see various projects using Linked Data to construct their services and datasets.

Meanwhile, the ongoing research on Twitter has found that [10] six mood states (tension, depression, anger, vigor, fatigue, confusion) from the aggregated Twitter content and compute a six-dimensional mood vector for each day in the timeline. Needless to say, music also has an inseparable connect with emotion. Even contextual features [11] influence expressed emotion at different magnitudes, and their effects are compounded by one another. Lyric is a kind of Emotion-Presentation in music composing. It includes composers' implicit thinking. Therefore, we can try to fathom emotions and associated thinking in each song. Since Social Network Services and Music are all connected with Emotion-Presentation to express people's mental state, links between them are reasonable and feasible.

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 illustrated in Fig. 1.

This paper is organized as follows. Section 2 presents the data collection and their triplication to Linked Data which a basis of our music recommendation. Section 3 describes how to connect the Music Linked Open Data (LOD) and recommend the music based on the data. Section 4 describes preliminary experiment and evaluation of our system. Section 5 discusses related researches. Then, finally Sect. 6 concludes this paper with the future work.

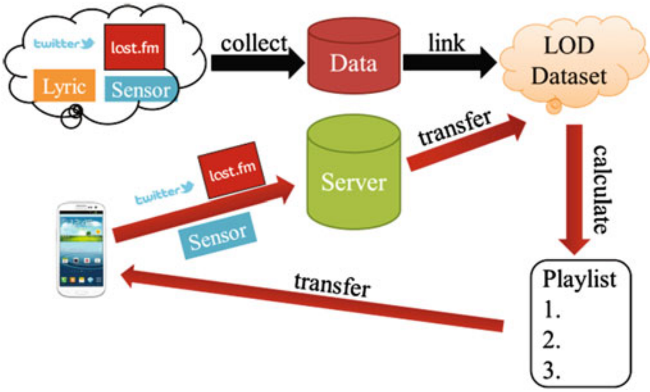

Fig. 1. System approach

2 Data Collection

When talking about music, inevitably involve 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 Last.fm to serve as the base of the music information. Also, we use GPS sensor to collect the user's context information, Twitter to present users' implicit thoughts, and lyrics of English songs and Japanese songs to present composers' implicit thoughts.

2.1 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. 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 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 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.

2.2 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 geographical map and local information that Yahoo! JAPAN provides to the developers. You can take advantage of the rich features of 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 Fig. 2. We mainly use this API to do translation and its request and response are just like in Fig. 3 showed below.

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

2.3 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 name is 584, input each name as 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 Japanese tweets alone.

MeCab [12] is a fast and customizable Japanese morphological analyzer. 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 sentence, and returns parts of speech. MeCab identify sentence into a noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. After morphological analysis, all the parts of speech are collected, we counted times each occurs and total number of them. Then use this simple but effective algorithm TF-IDF.

2.4 Lyric

Lyric information is very important for identifying the music, since there is no two songs have the same lyric. Hence, we build an environment to catch up English and Japanese music lyrics. For the English music, we receive the lyric data from LyricWiki. It has very high credibility just like Wikipedia. LyricWiki released its API in 2008. The LyricWiki API lets you search for songs, artists, and albums. 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

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

Fig. 2. Part of industry codes

```
http://placeinfo.olp.yahooapis.jp/V1/get?lat=35.66521320007564&lon=139.7
300114513391&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
      },
```

Fig. 3. Example of request(upper) and response(lower) through Yahoo! Local Search API

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.

2.5 Linked Data Conversion

Once collected, data is organized in different formats. Before converting into Linked Data in RDF, defining schema is indispensable. We aim to make RDF triples providing descriptions of relevance using the Music Ontology [13] and a large SKOS taxonomy [14].

After collecting data and defining schema, we use an online service called LinkData [15] to convert table data into an RDF file. Each data 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 have full of redundancies, hence some post-processing is done to fix the problem.

The detailed structure of our Music LOD is presented in Fig. 4.

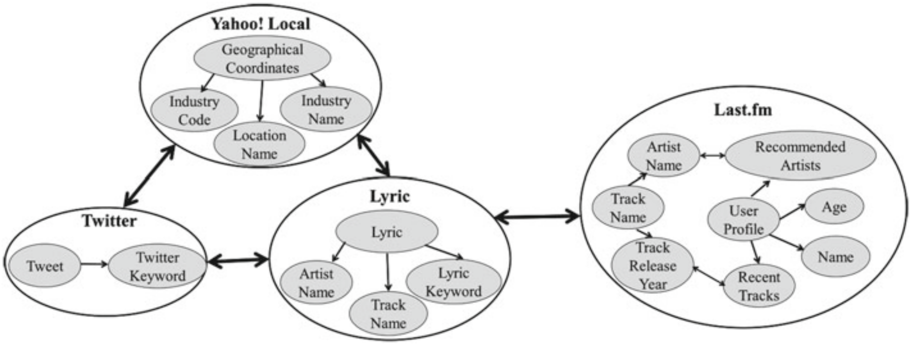

Fig. 4. Data relationship

We collected four datasets altogether: 20,050 triples from Last.fm, 584 triples from Yahoo! Local, 584,000 triples from Twitter and 400,000 triples from Lyric. And then we use over 1,000,000 links which can be divided into 15 types to make relations. Yahoo! Local, Twitter and Lyric connect each other, while Last.fm connects with Lyric directly. The nodes are connected through properties like *mo:track*, *owl:hasValue* and *mo:lyrics*.

3 Proposal of 'renso' Alignment of Music 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 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 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 lyric. Therefore, we propose new relations called 'renso', that mean 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. Thus, 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 $R \subseteq A_1 \times ... \times A_n$.

3.1 Three Alignments

Based on the experiments, we defined three types of the 'renso' alignments between different LOD for the music recommendation.

The first one is the simplest query method. It mainly connects twitter keywords and lyric keywords, and presented in Fig. 5. In 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 return 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 'Bijutsukan' (Art Museum). We then use 'Bijutsukan' (Art Museum) as a keyword to search tweets through Twitter API, and obtain several twitter keywords with strong relevance to 'Bijutsukan' (Art Museum), then count the number of the keywords appeared in each song lyric. 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 'Meganegoshinosora' (Sky over glasses). The same keywords are 'Shashin' (Photograph), 'Yume' (Dream) and 'Megane' (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 the future use. In the recommendation, we can search on the graph of LOD with tracing the 'renso' relations, and find the song using SPARQL querying methods.

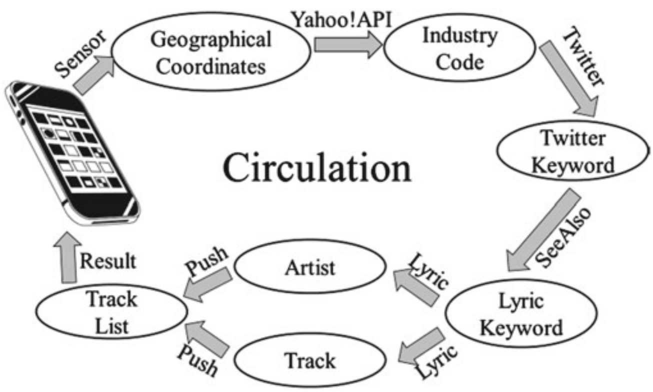

Fig. 5. Alignment 1

Alignment 2 and alignment 3 are on the basis of the above alignment 1, adding user's profile information and the user's listening history with the recommended artist information separately. They go through Yahoo! Local, Twitter, Last.fm and Lyric graphs sequentially.

Definitions of the alignments are as follows.

Considering that there is an infinite set U (RDF URI references), and an infinite 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_1 we shall use U_1, L_1 to denote the URIs, literals that appear in the triples of G_1 respectively. The nodes of G_1 are the values that appear as subjects or objects in the triples of G_1. The main graph G_1 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 in chapter 3 can be expressed in the following pseudo-code. First, we define the following additional functions as proposed in

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
        SEARCH(x, Y)
    end for
    return Result
    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
```

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 numeric value.

Our starting point is geographical *coordinates(x,y)* received from user's smartphone, while ending point is a music playlist. Our mapping pseudo-code is defined as Algorithm 2 and Algorithm 3. It describes three alignments' implementation procedures. The three alignments are designed to use less profile information in order to avoid cold start problem. Meanwhile, be independent of user profile lead to high serendipity. What we try hard to do is enhance accuracy.

4 Music Recommendation Experiment

In this section, we describe the results of our recommendation system. Adequacy of the 'renso' relation highly depends on the user's feeling, and has difficulty 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 significant promise from both perspectives.

4.1 Dataset

All the data were collected for two weeks; their details are presented in Table 1. We cascade all the data and let the prior one to determine the subsequent one.

Based on the industry codes, we defined 584 location keywords. By using these location keywords, we employ Twitter as a platform to search tweets. Every location keyword corresponds to 1,000 tweets. Then we utilize MeCab to extract 50 twitter keywords by analyzing every 1,000 tweets. Also, Last.fm stores many aspects of information. All the data are collected through its API methods. We set Japan as the main place for experiments, request Top 500 artists in Japan and their Top 20 tracks. Afterwards, I use my Last.fm account to acquire any personal information for alignment 2 and alignment 3. Every track corresponds to one lyric data, and 10,000 in total. As well as the tweets, we processed lyric data with MeCab and extracted 40 lyric keywords per song.

Table 1. Data details

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

4.2 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,

Algorithm 2. Three Alignments Part 1

```
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
        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
        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
        Search(year, F)
    end for
    return YearTrack
    for all yeartrack ∈ YearTrack do
        Search(yeartrack, L)
    end for
    return Track(Lyric)
    for all tracklyric ∈ Track(Lyric) do
        Search(tracklyric, L)
    end for
    return Track(LK)
    procedure Base
    Count ← Match(Track(TK), Track(LK))
    Sort(Track) by Count
    for count ← 0, 2 ∈ Track do
        Search(count, F)
    end for
    return Track(Info)
End
```

Algorithm 3. Three Alignments Part 2

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

the others are not. The question are contains ten items which is formed in 3 parts: situation description, question 1 and question 2. Situation description is written in this pattern: "When you are in (location name) right now, our system recommend a track which has no direct connection with (location name) but contain (keyword) and (keyword) in its lyric named (song name)". Question 1 "Do you think this recommendation is unexpected but fun?" is asking for the serendipity, while question 2 "Do you think this recommendation is fit for this place?" is for the accuracy. We wrote the question are with forms in Google Drive and set it public for test users here[1].

The result of the experiment is presented in Fig. 6. We can see that all three alignments achieved about 60 % satisfaction in the accuracy and the serendipity. Among them, alignment 3 presented the highest value; both evaluating indicators are above 65 %. While in Table 2 that shows the user's favorite alignment is alignment 1. Overall, although alignment 3 shows the best performance, the test users preferred the one uses the most simplest but straight recommendation method.

Table 2. The result of favorite alignment

Alignment 1	Alignment 2	Alignment 3
45 %	25 %	30 %

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

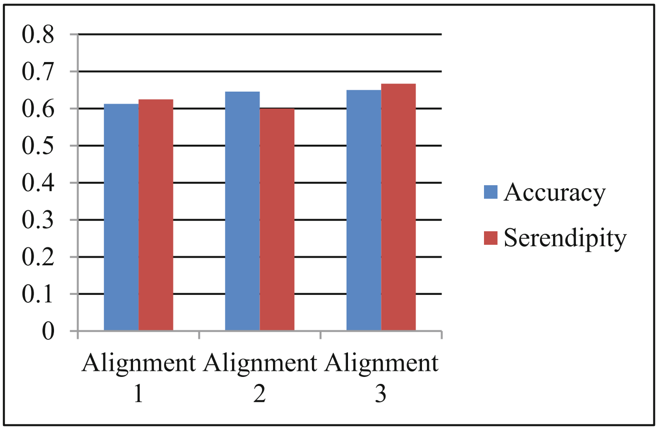

Fig. 6. The result of experiment

4.3 Lesson Learnt

Through this experiment, we found two key points. First, we confirmed there is no cold start problem in our recommendation method, since our base is the large set of Music LOD, where a point is about the data relevance. Twitter is a platform 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 the emotion extraction is very difficult, not to mention implicit thoughts. Even so, along with the increase of the relation path's length in the LOD, we could find more implicit connections to reach higher accuracy and serendipity.

Also, unlike the conventional content-based method, we confirmed our system is easier to archive the serendipity in the recommendation. However, meeting both the accuracy and the serendipity at a time is an arduous task. This research is for getting the users' implicit context, but the emotion is more complicated and changeable than any other. For example, in our university there are so many students who love sci-fi and robot animation, and are especially crazy about 'Evangelion' (Japanese TV Animation Program). Thus, 'University of Electro-Communications' comes along with 'Evangelion' frequently on Twitter. Then, we recommend 'Evangelion' theme song for the user who is in our university. It is full of fun, but the accuracy is subtle. In other words, it is difficult enough for satisfying everyone. As a summary of our evaluation, only one of two criteria, accuracy and serendipity can be satisfied in most of the cases.

5 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 constant the users' preferences, users' context (such as location, mood and people around) is 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 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 [16] 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 [17] is a service targets 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 [18] 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.

Foxtrot [19] 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.

6 Conclusions

In this paper, we presented a music recommendation method using the 'renso' alignment that connects various kinds of LOD, aiming at 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 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 the other media like 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 24300005C2350003-9C25730038 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: workshop keynote abstract. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 865–866. ACM, Lyon (2012)
2. Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S., Seada, K.: Fusing mobile, sensor, and social data to fully enable context-aware computing. In: Proceedings of the 11th Workshop on Mobile Computing Systems & Applications, pp. 60–65. ACM, New York (2010)
3. Komori, M., Matsumura, N., Miura, A., Nagaoka, C.: Relationships between periodic behaviors in micro-blogging and the users' baseline mood, pp. 405–410. IEEE Computer Society (2012)
4. Elliott, G.T., Tomlinson, B.: PersonalSoundtrack: context-aware playlists that adapt to user pace. In: CHI '06 Extended Abstracts on Human Factors in Computing Systems, pp. 736–741. ACM, New York (2006)
5. Kaminskas, M., Ricci, F.: Location-adapted music recommendation using tags. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 183–194. Springer, Heidelberg (2011)
6. Cebrián, T., Planagumà, M., Villegas, P., Amatriain, X.: Music recommendations with temporal context awareness. pp. 349–352. ACM, New York (2010)
7. Bu, J., Tan, S., Chen, C., Wang, C., Wu, H., Zhang, L., He, X.: Music recommendation by unified hypergraph: combining social media information and music content. In: Proceedings of the International Conference on Multimedia. pp. 391–400. ACM, New York (2010)
8. North, A.C., Hargreaves, D.J., Hargreaves, J.J.: Uses of music in everyday life. Music Percept. Interdiscip. J. **22**(1), 41–77 (2004)
9. Zhang, Y.C., Séaghdha, D., Quercia, D., Jambor, T.: Auralist: introducing serendipity into music recommendation. In: Proceedings of the 5th ACM International Conference on Web Search and Data Mining, pp. 13–22. ACM, New York (2012)
10. Johan, B., Alberto, P., Huina, M.: Modeling public mood and emotion: twitter sentiment and socio-economic phenomena. CoRR. abs/0911.1583 (2009)
11. Scherer, K.A., Zentner, M.R.: Emotional effects of music: production rules. In: Juslin, P.N., Sloboda, J.A. (eds.) Music and Emotion: Theory and Research, pp. 361–392. Oxford University Press, New York (2001)

12. MeCab: Yet Another Japanese Dependency Structure Analyzer. https://code.google.com/p/mecab/
13. Music Ontology. http://musicontology.com/
14. SKOS Simple Knowledge Organization System. http://www.w3.org/2004/02/skos/
15. LinkData. http://linkdata.org/
16. Twitter Music. https://music.twitter.com/
17. Khrouf, H., Milicic, V., Troncy, R.: EventMedia live: exploring events connections in real-time to enhance content. In: ISWC 2012, Semantic Web Challenge at 11th International Semantic Web Conference (2012)
18. Lehtiniemi, A.: Evaluating superMusic: streaming context-aware mobile music service. In: Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, pp. 314–321. ACM, New York (2008)
19. Ankolekar, A., Sandholm, T.: Foxtrot: a soundtrack for where you are. In: Proceedings of Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications, pp. 26–31. ACM, New York (2011)

http://www.springer.com/978-3-319-06825-1

Semantic Technology
Third Joint International Conference, JIST 2013, Seoul, South Korea, November 28--30, 2013, Revised Selected Papers
Kim, W.; Ding, Y.; Kim, H.-G. (Eds.)
2014, XII, 462 p. 191 illus., Softcover
ISBN: 978-3-319-06825-1