Implementing an Intelligent Karaoke to a Social Networking

Alberto Ochoa[1], Fernando Montes[2], Jöns Sánchez[3], Alberto Hernández[4], and Julio Ponce[5]

[1] Inst. de Ing. y Tec., Universidad Autónoma de Ciudad Juárez, México
[2] Departamento de Inteligencia Artificial, Universidad Veracruzana
[3] CIATEC (Centro Conacyt)
[4] Facultad de Ciencias, Universidad Autónoma del Estado de Morelos
[5] Laboratorio de Inteligencia Artificial, Universidad Autónoma de Aguascalientes
alberto.ochoa@uacj.mx

Abstract. In this research is described an Intelligent Karaoke associated to a Social Networking which uses different songs in a database related with a kind of Music (Eurovision songs) which employs the Dublin Core metadata standard for the documents description, the XML standard for describing user profile, which is based on the users profile, and on service and data providers to generate musical recommendations to a group of persons. The main contribution of the work is to provide a recommendation mechanism based on the user of this Social Networking reducing the human effort spent on the profile generation. In addition, this paper presents and discusses some experiments that are based on quantitative and qualitative evaluations.

Key words: Recommendation System, User Profile and Cultural Algorithms.

1 Introduction

Today, the songs can be electronically accessed as soon as they are published on the Web, and these can be used in a Karaoke an artifact invented in Japan by Daisuke Inoue in 1969 which displays the lyrics of a song and the musical track. The main advantage of open music is the minimization of the promotion time. In this context, Digital Libraries (DLs) have emerged as the main repositories of digital documents, links and associated metadata. The Recommender System involves information personalized. The personalization is related to the ways in which contents and services can be tailored to match the specific needs of a user or a community (social networking) [1]. The human-centered demand specification is not an easy task. One experiences this difficulty when trying to find a new song in a good indexing and retrieval system such ESC Radio.

The query formulation is complex and the fine tuning of the user requirements is a time-consuming task. Few users have enough time to spend some hours searching for, eventually new songs. This functionality, the query specification may be reached by the analysis of the user activities, history, information demands, in others. This paper presents a Musical recommendations system of a Karaoke associated to a social networking, the songs recovered are associated with the Karaokes playlist. The main contribution of this work is to provide a recommendation mechanism based on the user reducing the human effort spent on the profile generation. The paper is organized as follows. We start giving an overview of the background literature and concepts, then the recommender system and detail its architecture and techniques. Finally, we present some quantitative and qualitative experiments to evaluate and validate our system and discuss the results and conclusions of our work.

2 Background

The semantic Web technologies promote an efficient and intelligent access to the digital documents on the Web. The standards based on metadata to describe information objects have two main advantages: computational efficiency during the information harvesting process and interoperability among DLs. The first is a consequence of the increasing use of Dublin Core (DC) metadata standard [2]; the latter has been obtained as a result of the OAI initiative (Open Archive Initiative) [3]. DC metadata standard was conceived with the objective of defining a minimal metadata set that could be used to describe the available resources of a DL. This standard defines a set of 15 metadata (Dublin Core Metadata Element Set DCMES) [2].

The main goal of OAI is to create a standard communication way, allowing DLs around the world to interoperate as a federation [4]. The DL metadata harvesting process is accomplished by the OAI-PMH protocol (Open Archives Initiative Protocol for Metadata Harvesting) [5], which define how the metadata transference between two entities, data and service providers, is performed. The data provider acts by searching the metadata in databases and making then available to a service provider, which uses the gathered data to provide specific services.

Considering that a Recommender System concerns with information personalization, it is essential that it copes with user profile. In our work, the user profile is obtained from a social networking similar at used in [6]. According to [7], there are three different methodologies used in Recommender Systems to perform recommendation: (i) content-based, which recommends items classified accordingly to the user profile and early choices; (ii) collaborative filtering, which deals with similarities among users interests; and (iii) hybrid approach, which combines the two to take advantage of their benefits. In our work, the content-based approach is used, once the information about the user is taken from Karaoke users.

This recommendation process can be perceived as an information retrieval process, in which users relevant songs should be retrieved and recommended. Thus, to perform recommendations, we can use the classical information retrieval models such as the Boolean Model, the Vector Space Model (VSM) or the Probabilistic Model [8–10]. In this work, the VSM was selected since it provides satisfactory results with a convenient computational effort. In this model, songs and queries are represented by terms vectors. The terms are words or expressions extracted from the documents (lyrics) and from queries that can be used for content identification and representation. Each term has a weight associated to it to provide distinctions among them according to their importance. According to [11] the weight can vary continuously between 0 and 1. Values near to 1 are more important while values near to 0 are irrelevant.

The VSM uses an n-dimensional space to represent the terms, where n corresponds to the number of distinct terms. For each document or query represented the weights represent the vectors coordinates in the corresponding dimension. The VSM principle is based on the inverse correlation between the distance (angle) among term vectors in the space and the similarity between the songs that they represented. To calculate the similarity score, the cosine (Equation 1) can be used. The resultant value indicates the relevance degree between a query (Q) and a document (song) (D), where w represents the weights of the terms contained in Q and D, and t represents the number of terms (size of the vector). This equation provides ranked retrieval output based on decreasing order of the ranked retrieval similarity values [11].

$$Similarity(Q,D) = \frac{\sum_{k=1}^{t} w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2 \cdot \sum_{k=1}^{t} (w_{dk})^2}} \tag{1}$$

The same equation is widely used to compare the similarity among songs, and similarity, in our case, Q represents the user profile and D the documents descriptors (lyrics) that are harvested in the DL (see Section 3.2 for details). The term weighting scheme is very important to guarantee an effective retrieval process.

The results depend crucially of the term weighting system chosen, In addition, the query terms selection is fundamental to obtain a recommendation according to the user necessities. Our research is focused in the query terms selection and weighting. Any person of the social networking that required a musical retrieval may evaluate the process complexity and the difficulty to find the adequate songs. The central idea is to develop an automated retrieval and musical recommendation system where the price for the user is limited to the submission of an already existing preferences query similar at the used on Esc Radio.

3 The Recommender System

Our system focuses on the recommendation of songs from a Karaoke system and its social networking that uses this. The information source to perform recommendations is the database associated with this Karaoke System (Lyrics and Music of each song), while the user profile is obtained from Database Profile

Register subset. However, any DL repository providing DC metadata and supporting the OAI-PMH protocol can be used as a source. An alternative to the user profile generation is under development. This alternative approach is composed by an information retrieval system to gather data from another Music sources. A DL repository stores digital songs or its localization (web or physical), and the respective metadata. A DL data provider allows an agent to harvest documents metadata through the OAI-PMH protocol. Our system handles the songs described with XML in DC standard [12, 13].

3.1 The Recommendation System Architecture

In this section we present the architecture elements of our system and its functionalities (Fig. 1). To start the process, the users must supply their preferences in the XML version to the system. Whenever a user makes its registration in the system and sends his preferences list (1), the XML Songs Document module is activated and the information about the users interests is stored in the local database named User Profile (2). Then the Metadata Harvesting module is activated to update the local database Songs Metadata. This module makes a request to a DL data provider to harvest specific documents metadata. It receives an XML document as response (3) and the XML DC to local DB module is activated (4). This module extracts the relevant metadata to perform the recommendation from the XML document and stores it in the local database named Songs Metadata (5). Once the user profile and the songs metadata are available in the local database, the Recommendation module can be activated (6). The focus is to retrieve lyrics and songs of a DL that the best matches the user profile described through the profile of each user of a social networking related with the Karaoke System.

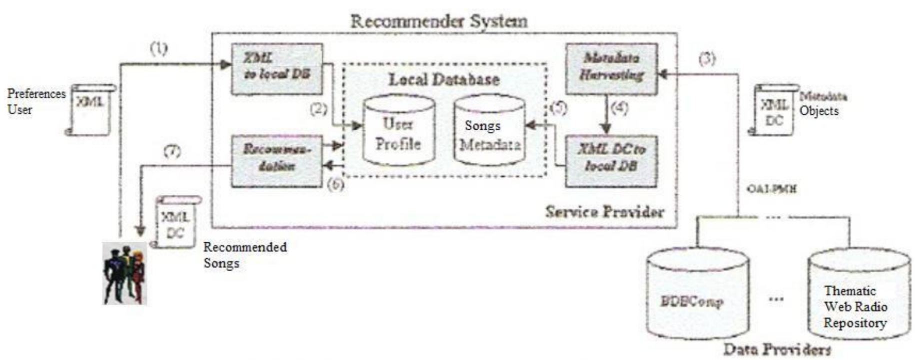

Fig. 1. The recommender system architecture

3.2 The Recommendation Model

As stated before, the recommendation is based on the VSM model. The query vector is built with the term parsed from the title, keywords, singer or band, album and date. The parser ignores stop-words [14] (a list of common or general terms that are not used in the information retrieval process, e.g., prepositions, conjunctions and articles). The parser considers each term as a single word. On the other hand, the terms are taken integrally, as single expressions.

The query vector terms weights are build up according to the Equation 2. This equation considers the type of term (keyword or title), the language and the year of the first air data. Keyword terms are considered more important that the titles of the songs and have more reading proficiency are more valorized (higher weight), and the terms obtained from the most recent album from an artist or band including cameos and contributions with another singers or bands are assigned a more important weight than the less recent ones.

$$W_t = W_{KeywordOrTitle} * W_{Language} * W_{Year} \tag{2}$$

The weights $W_{KeywordOrTitle}, W_{Language}, W_{Year}$ are calculated with Equation 3.

$$W_i = 1 - (i-1)\left(\frac{1 - W_{min}}{n-1}\right) \tag{3}$$

In this equation W_i varies according to the type of weight we want to compute. To illustrate this in the experimental evaluation (Section 4), for $W_{KeywordOrTitles} W_{min}$ was 0.95, and I is 1 if the language-skill.level is "good", 2 for "reasonable" and 3 for "few". For $W_{Years} W_{min}$ was 0.55 and i vary from 1 to n, where n is the interval of years considered, begin 1 the highest and n the lowest. In the experimental evaluation it was considered the interval of songs between 2008 and 2002. However, if the interval is omitted, it will be considered as between the present year and the less recent year (the smallest between artist:first-album and artist:last-album).

If W_{min} is not informed, the default value will be used (presented in Equation 4). In the situation, Equation 3 is reduced to Equation 5.

$$W_{mindefault} = \frac{1}{n} \tag{4}$$

$$W_i = \frac{n - 1 + 1}{n} \tag{5}$$

Once the query vector is build, the songs vector terms and the respective weights must be defined. The adopted approach was $(tf * idf)$, i.e., the product of the term frequency and the inverse document frequency [11]. This approach allows automatic term weights assignment for the songs retrieval. The term frequency (tf) corresponds to the number of occurrences of a term in a document. The *inverse document frequency* (idf) is a factor that varies inversely with the number of the songs n to which a term is assigned in a collection of N songs (typically computed as $\log(N/n)$).

The best terms for content identification are those able to distinguish individuals ones from the remainder of the collection [11]. Thus, the best terms correspond to the ones with high term frequencies (tf) and low overall collection frequencies (high idf). To compute $tf * idf$, the system uses the DC metadata *dc:title* and *dc:description* to represent the songs content. Moreover, as your system deals with different languages, the total number of songs will vary accordingly. After building the query and songs vectors, the system is able to compute the similarities values among the songs and the query according to Equation 1.

4 Experimental Evaluation

In order to evaluate the musical recommender system, we have asked for preferences from a social networking entailed to different musical interest terms of different genres as English Pop or Electronic Dance. As response, a group of 47 people send us their list of preferences, whose information was loaded in the Songs Metadata related with the Karaoke System local database. The songs Metadata local database was loaded in the User Profile local database related with the Social Networking. This database stored up to January 2009, totalizing 1007 songs from 267 singers or bands including in 87 albums.

After 20 recommendations were generated by the system for each Karaokes performance, considering individuals profile of the user and the genres preferences. This information was obtained using the users data base related with the Social Networking.

Two evaluations were performed. The first was based on the hypothesis that the best songs to describe the profile of a user should be those produced by them. Since we had information about the songs by each user, we can match the items recommended to those. This evaluation was accomplished by the recall and precision metrics that is a standard evaluation strategy for information retrieval systems [8, 10]. The recall is used to measure the percentage of relevant songs retrieved in relation to the amount that should have been retrieved. In the case of songs categorization, the recall metric is used to measure the percentage of songs that are correct classified in relation to the number of songs that should be classified. *Precision* is used to measure the percentage of songs correctly recovered, i.e., the number of songs correctly retrieved divided by the number of songs retrieved.

As the profiles can be seen as classes and the songs as items to be classified in these profiles, we can verify the amount of items from the author that are correctly identified (i.e. classified) by the user profile. As we have many users (i.e., many classes), it is necessary to combine the results. The *macroaverage* presented in Equation 6 was designed by D. Lewis [15] to perform this specific combination ("the unweighted mean of effectiveness across all categories"), and was applied by him in the evaluation of classification algorithms and techniques.

$$macroaverage = \frac{\sum_{i=1}^{n} X_i}{n} \tag{6}$$

In this formula, X_i is the recall or the precision, depending on the metric we want to evaluate, of each individual class (user in our case) and n is the number of classes (users). Thus, the *macroaverage recall* is the arithmetic average of the recalls obtained for each individual, and the *macroaverage precision* is the arithmetic average of the precisions obtained for each individual.

Given that the users are not interested in its own preferred songs as recommendations, we performed another evaluation that takes into account only the items from other users. Then, 15 recommendations were presented to each individual ranked on the relative grade of relevance generated by the system. In this rank, the songs with the highest grade of similarity with the user profile were set as 100% relevant and the others were adjusted to a value relative to it. In this case, each author was requested to evaluate the recommendations generated to them assigning one of the following concepts (following the bipolar five-point Lickert scale); “Inadequate”, “Bad”, “Average”, “Good”, and “Excellent”, and were also asked to comment the results. The following sections present the results obtained.

5 Analysis of Experiments

The first experiment was designed to evaluate the capability of the system to correctly identify the user profile (i.e., to represent its preferences), since we believe that the best songs to describe the user profile are those selected by themselves, as stated before. To perform such evaluation, we identified the songs of each user had at Karaoke System. After that, we employed the *recall* metric to evaluate the number of articles recovered for each author and combined then with the *microaverage* equation explained before.

We have found a macroaverage recall of 43.25%. It is important to state that each user received 20 recommendations. This is an acceptable value as the query construction was made automatically without human intervention. It happened to be lower than it should be if we have used more songs, maybe access to ESC Radio, but the problem is the limited songs for singer or band. Other important consideration is that the recommendation ranking was generated with a depreciation degree that was dependent on the promotion year and on the user language, as explained in the previous section. As the time-slice considered corresponds to a small part of the full period stored in the database related with the Karaoke System, not all songs are good recommendations since the preferences changes along the time, similar at propose in [16].

Figure 2 presents the results of the second experiment, which was based on the users’ qualitative evaluation of the recommended songs. On this experiment each user received 15 recommendations and evaluated them according to one of the following concepts: “inadequate”, “bad”, “average”, “good”, and “excellent”.

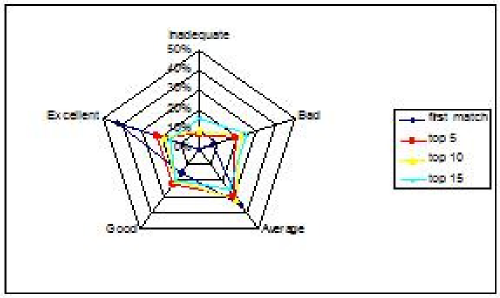

Fig. 2. Users' evaluations of the recommendations

The results were grouped into the categories "first match", "top 5", "top 10", and "top 15", and are presented in Figure 2.

Analyzing three results, it is possible to observe that, if we only consider the first song recommended (the "first match"), the number of items qualified as "excellent" in greater than the others (i.e., 42.86%) and none of them were classified as "inadequate". This strengthens the capability of the system on performing recommendations adjusted to the present users genre preferences interests. We have also grouped the concepts "good" and "excellent" into a category named "positive recommendation" and the concepts "bad" and "inadequate" into a "negative recommendation" group, so we could obtain a better visualization and comprehension of the results (Fig. 3).

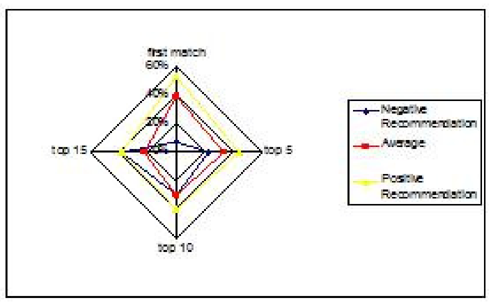

Fig. 3. Grouped users' evaluation

We could perceive that the positive recommendations, considering only the "first match", are superior (57.14%) in relation to the negative ones (7.14%). The same behavior can be perceived in the "top 5" and "top 10" categories, the recommendations had a negative evaluation only in the "top 15" category, and that probably happened because as the number of recommendations grows, the number of correct recommendations falls. It is clear that the automated procedure here adopted is adequate for an alert recommender system. Our proposal is to add to the Karaoke System an automated alert system that periodically sends to the user a list of the most relevant songs recently listen on ESC Radio during seven or more weeks.

Further, in our tests the users that have changed their search in the last three months have negatively qualified the recommendations. In the next experiments a variable time threshold and different depreciation values will be employed and the temporal component will be exhaustively analyzed.

6 Conclusions

This paper presented a Musical Recommender System to users of a Karaoke System related with the lyrics of plethora of songs. In current days, in which the recovery of relevant digital information on the Web is a complex task, such systems are of great value to minimize the problems associated to the information overload phenomena, minimizing the time spent to access the right information.

The main contribution of this research consists on the heavy utilization of automated Music Recommendation and in the use of a Digital Library (DL) metadata to create the recommendations. The system was evaluated with BDB-Comp, but it is designed to work wit the open digital library protocol OAI-PMH, then it may be easily extended to work with any DL that supports this mechanism. The same occurs with the lyrics format related with the song, but it can be extended to support other formats or to analyze information about the user stored on tools like ESC Radio. Alternatively the operational prototype offers the possibility to the user to load the lyrics via an electronic form.

The developed system will have many applications. One of them is the recommendation of songs to support a Social Networking related with a kind of music to relax activities. Thus, the user could log into a specific distance or electronic Karaoke System environment supported by this system and receive recommendations of songs containing actualized relevant material to complement its current musical selection.

References

1. Callahan, Jamie et al.: Personalization and Recommender Systems in Digital Libraries. Joint NSF-EU DELOS Working Group Report, May (2003).
2. Dublin Core Metadata Initiative: http://dublincore.org, Sept (2005).
3. OAI: Open Archives Initiative: http://openarchive.org, Oct (2005).
4. Sompel, H.; V. de Lagoze, C.: The Santa Fe Convention of the Open Archives Initiative D-Lib Magazine, [S.1.], v.6, n.2, Feb (2000).
5. OAI-PMH. The Open Archives Initiative Protocol for Metadata Harvesting: http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm. Nov. (2005).
6. Laustanou, K.: MySpace Music (2007)
7. Huang, Z. et al.: A Graph-based Recommender System for Digital Library. In JCDL02 Portland, Oregon (2002).

8. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Workingham, UK (1999).
9. Grossman, David A.: Information retrieval: algorithms and heuristics. 2nd ed. Dordrecht: Springer, 332 p. (2004).
10. Salton, Gerard; Macgill, Michael J.: Introduction to Modern Information Retrieval. New York. McGraw-Hill. 448p. (1983).
11. Salton, Gerard; Buckley, Christopher: Term-Weighting Approaches in Automatic Text Retrieval, Information Processing and Management an International Journal, v.24, Issue 5, 513-523 (1988).
12. DC-OAI: A XML schema for validating Unqualified Dublin Core metadata associated with the reserved oai_dc metadataPrefix: http://www.openarchives.org/OAI/2.0/oai_dc.xsd, 2005.
13. LPMP-CNPq. Padronização XML: Curriculum Vitae: http://lmpl.cnpq.br Mar. (2005).
14. CLEF and Multilingual information retrieval: http://www.unine.ch/info/clef/, Institut interfacultaire d'informatique, University of Neuchatel (2005).
15. Lewis, D.D.: Evaluating text categorization. In Proceedings of Speech and Natural Language Workshop. Defense Advanced Research Projects Agency, Morgan Kaufmann. (1991) 312-318.
16. Ochoa, A., et al.: Musical Recommendation on Thematic Web Radio. In Journal of Computers, Oulu, Finland (2009) By publishing.