A PROPOSAL OF INFORMATION RETRIEVAL METHOD BASED ON TPO METADATA

Ismail Arai

Graduate school of Information Science,
Nara Institute of Science and Technology
8916-5 Takayama Ikoma, Nara, 630-0192, Japan
E-mail : ismail-a@is.naist.jp

Kazutoshi Fujikawa, Hideki Sunahara

Information Technology Center,
Nara Institute of Science and Technology
8916-5 Takayama Ikoma, Nara, 630-0192, Japan
E-mail : fujikawa@itc.naist.jp, suna@wide.ad.jp

ABSTRACT

To search the contents through WWW in consideration of user's circumstance condition, search engines should make use of some information such as current time, user's location, user's schedule and so on. We propose an information retrieval method that effectively uses both user's and content's metadata based on TPO (Time, Position, Occasion). We implement a prototype of information retrieval system based on the proposed method, which consists of query generation part, contents matching part, and matched results scoring part. In our query generation part, we reduce user's overload of choosing metadata. And a user becomes possible to retry sorting the matching results.

1. INTRODUCTION

A search engine, which provides information that depends on user's circumstance, is desired. For example, a user wants to get weather forecast in business trip, or find a preferable restaurant near user's location. To acquire the information related to user's circumstance, we need to issue a query that includes user's circumstantial information exactly. Then, a search engine should understand such query. Also, some supplementary data are desired because a query in a text form includes little circumstantial information.

A sequence of an ordinary information retrieval is shown as follows:

1. A user sends a query, which includes user's circumstantial information as keywords to a search engine.

2. The search engine compares user's query with contents. (Matching)

3. The search engine sorts the results of the matched contents. (Scoring)

4. The search engine returns results to the user.

Usually, users make a query for a search engine to acquire information reflecting his circumstance. When a query depends on only text input, it is difficult to take current information such as current time and current location. Thus, their information should be appended to a query as metadata.

W3C (World Wide Web Consortium) [1] standardizes RDF (Resource Description Format) [2] which is a description format of web documents. Semantic Web [3] working group discusses structure web documents. In Semantic Web, contents have a lot of metadata, which are effective for information retrieval. However, a user suffers from choosing a huge amount of metadata for information retrieval.

Context-Aware system enables users to make query including their circumstantial information automatically [4, 5, 6, 7]. In the future, methods that make query automatically from user's context are essential to realize an information retrieval adapting to user's circumstance. On the other hand, to choose suitable metadata automatically, data mining technique and natural language processing are promising. However, data mining technique casts anchor at applying to web usage mining [8]. There is no information retrieval method to embed their data mining technology element as a component of system flexibly.

In the contents matching part, most systems are using text pattern matching. It is airy ideas that a search engine processes pattern matching to all contents. The matching method that reflects user's query including strict metadata is desired.

In the matched results scoring part, most search engine having a huge amount of contents e.g. Google [9] use static scores such as Pagerank [10]. When a search engine gets matching results, it can only to sort results by static scores. When the search engine sorts results by dynamic scores, the load of the search engine becomes higher.

However, sometimes a user wants search engine to calculate the score dynamically when the user desires to match exactly a user's query with search results. For example, when user's query includes the exact location information such as GPS (Global Positioning System) data, the user wishes the score of a result were getting higher if user's location is closed to content's location. If a search engine uses the

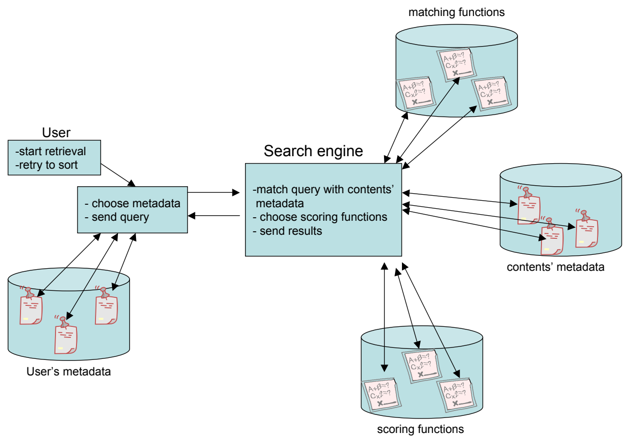


Fig. 1. A model of proposed information retrieval

static score, it cannot realize such requirement.

2. REQUIREMENTS FOR INFORMATION RETRIEVAL

According to ordinary sequence of information retrieval, we propose an information retrieval model based on TPO metadata. Fig. 1 shows a system environment. The proposed system consists of three parts, query generation part, matching part, and scoring part.

2.1. Query generation mechanism

A query consists of plain text keywords and metadata. The plain text keywords are taken for full-text search; Metadata are supplemental data for user's circumstantial information that is difficult to describe in the form of plain text. We suppose user's circumstantial information is written and stored by metadata in advance. For example, location information is sensed from GPS, time information is gotten from a clock, user's schedule data, and so on. Their circumstantial information is systematized based on TPO (Time, Position, Occasion). TPO is a description model, which reflects human behaviors.

To retrieve information by several search engines in WWW, we have to choose adaptive user's metadata unlike context aware system. But, the cost of choosing user's metadata, which the search engine requires, gets higher. When a lot of metadata are stored into user's storage, a mechanism automatically selecting metadata is desired.

There are two methods of choosing metadata automatically. One method is that the search engine sends a user a required user's metadata list. When a user connects a search engine, the search engine tells the required metadata to the user, and then the user chooses user's metadata according to it. Another method is that a user utilizes a data mining engine for behavior prediction of a user. A data mining engine predicts user's requirements and suggests the suitable user's metadata to the user. To realize the suitable user's metadata choosing mechanism, we develop a query generation mechanism.

Also, a query has scoring parameter because it is required that scoring parameter should be chosen automatically.

2.2. Content retrieving mechanism

The search engine compares user's metadata with contents' metadata to judge whether both of metadata are matching or not. Also, we suppose a content creator describes contents' metadata. The quantity of matching functions should be sufficient to compare any user's metadata and contents' metadata. Matching functions compare values of user's metadata and contents' metadata. If the value is written in a text format, a matching function works as text pattern matching function. If the value is written in a numerical value, a matching function does as arithmetic formula. The search engine is desired to select a proper matching function for the query.

The contents, which are fitting for user's purpose, should be listed upper in the matching results. Therefore, the search engine sorts contents by the score. A score is given by total results of scoring which are calculated by scoring functions with values of each metadata.

The search engine has a weight value for each TPO property. It allows a user to adjust scoring results without sending metadata for matching part. Ordinary search engines have "search engine spam" problem because of using static scoring method. Averse to that, the proposed method moderates such spam problem by adjusting scoring results.

3. A DESIGN OF M3 SEARCH ENGINE

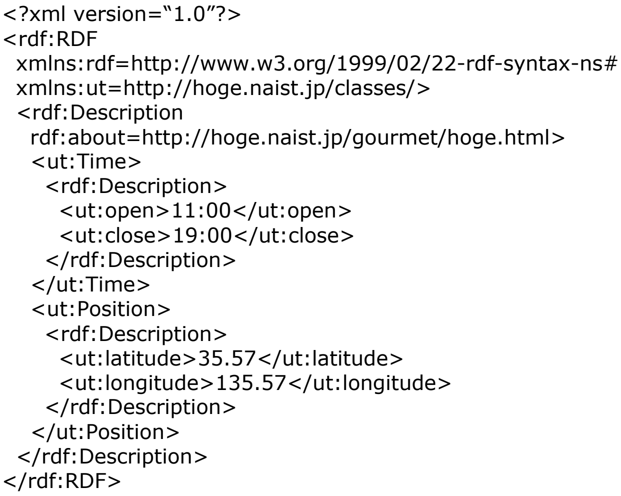
According to the requirement, we design M3 (Make the best use of Mutual Metadata) search engine.

3.1. Query generation mechanism

A sequence of query generation mechanism is as follows.

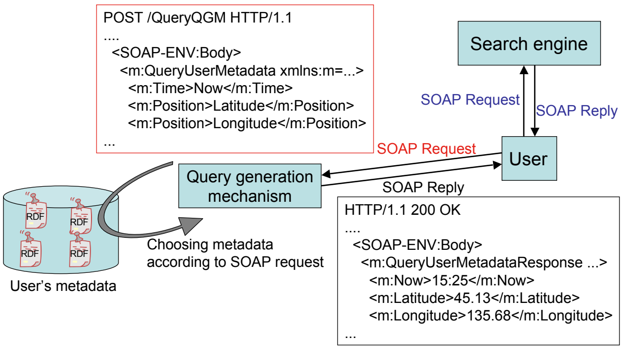
1. A user connects to a search engine, and he/she receives a text box and syntax of user's metadata which the search engine requires.

2. The user sends keywords written in a syntax of user's metadata to query generation mechanism; The user receives a query written in XML format from a query generation mechanism.



```
<?xml version=“1.0”?>
<rdf:RDF
 xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
 xmlns:ut=http://hoge.naist.jp/classes/>
 <rdf:Description
  rdf:about=http://hoge.naist.jp/gourmet/hoge.html>
  <ut:Time>
   <rdf:Description>
    <ut:open>11:00</ut:open>
    <ut:close>19:00</ut:close>
   </rdf:Description>
  </ut:Time>
  <ut:Position>
   <rdf:Description>
    <ut:latitude>35.57</ut:latitude>
    <ut:longitude>135.57</ut:longitude>
   </rdf:Description>
  </ut:Position>
 </rdf:Description>
</rdf:RDF>
```

Fig. 2. User's metadata



```
POST /QueryQGM HTTP/1.1
....
 <SOAP-ENV:Body>
  <m:QueryUserMetadata xmlns:m=...>
   <m:Time>Now</m:Time>
   <m:Position>Latitude</m:Position>
   <m:Position>Longitude</m:Position>
...
```

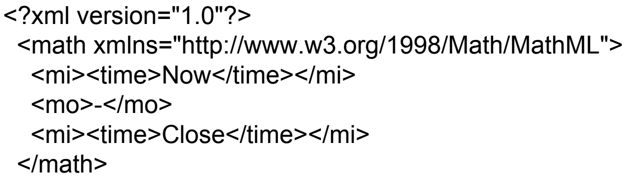
```
HTTP/1.1 200 OK
....
 <SOAP-ENV:Body>
  <m:QueryUserMetadataResponse ...>
   <m:Now>15:25</m:Now>
   <m:Latitude>45.13</m:Latitude>
   <m:Longitude>135.68</m:Longitude>
...
```

Fig. 3. Query generation mechanism

3. When the user is satisfied to the query, the user sends the query to the search engine. Otherwise, the user feeds back some modification to query generation mechanism.

4. The search engine sends results to the user. When the user is not satisfied to the scoring results, the user returns to the third phase. When the user is not satisfied to the matching results, the user returns to the second phase.

A query related to TPO is formed by user's metadata in RDF format. Fig. 2 shows user's metadata syntax. This circumstantial information is written in each part Time (from 11:00 to 19:00) and Position (latitude is 35.57 and longitude is 135.57). The quantity of property has no preference either a lot of metadata are written in one file, or each metadata is written in some files.



```
<?xml version="1.0"?>
 <math xmlns="http://www.w3.org/1998/Math/MathML">
  <mi><time>Now</time></mi>
  <mo>-</mo>
  <mi><time>Close</time></mi>
 </math>
```

Fig. 4. A scoring function written in MathML

When a search engine demands static user's metadata from a user in the first phase, a query generation mechanism chooses user's metadata written in RDF. Fig. 3 shows a model of query generation mechanism. Each messages are described in SOAP (Simple Object Access Protocol) [11]. In this instance, a user requests current time and current position. Then, the query generation mechanism answers ”15:25,” ”45.13,” and ”135.68” to the user.

3.2. Contents retrieving mechanism

A matching metadata method of M3 search engine compares each metadata but not full-text search. A user submits this metadata as a query to M3 search engine.

M3 search engine has matching functions to compare a query with contents' metadata as follows.

$$A = \prod_{i=T,P,O} \prod_{j=1}^{n_i} fm_{ij}(M_{fm_{ij_1}}, M_{fm_{ij_2}}, ..., M_{fm_{ij_k}}) \quad (1)$$

$$\begin{array}{rcl} fm_{ij} & = & \begin{cases} 1 : \text{when matches} \\ 0 : \text{when not matches} \end{cases} \quad (2) \\ M_{fm_{ij}k} & : & \text{metadata as argument} \end{array}$$

A matching function (2) returns ”1” when a content matches metadata. Otherwise it returns ”0”. If all results of matching function (1) return ”1”, it means query matches a content.

These matching functions are chosen by a mechanism selecting matching functions. Their functions are written in XML syntax, so the mechanism selecting matching functions is built as XSLT [12]. Matching functions are written in MathML [13] syntax which is specified by W3C. Generally, matching functions are stored in M3 search engine because of their communication costs.

M3 search engine sorts matched results by scoring function as follows.

$$S = \sum_{i=T,P,O} \sum_{j=1}^{n_i} \frac{k_i}{n_i} \cdot \frac{fs_{ij} - \overline{fs_{ij}}}{\overline{fs_{ij}}} \quad (3)$$

$$k_T + k_P + k_O = 1$$

$$\begin{array}{rcl} fs_{ij}(M_{fs_{ij_1}}, M_{fs_{ij_2}}, ..., M_{fs_{ij_k}}) & : & \text{scoring function} \\ k_T, k_P, k_O & : & \text{TPO weight value} \end{array}$$

Content's metadata are evaluated by each scoring function. Calculating with average value normalizes those results, and their total becomes a score of the content (3). The value k_i is weight value included in a query.

Fig. 4 shows scoring function written in MathML in situation of shop search. The more expand the gap between a current time and closing time of shop, the higher the score. The quantity of functions should be sufficient for existing metadata.

4. FUTURE WORKS

For the future, we contemplate to mount M3 search engine based on former designing. We need to set some scenario (e.g. sight-seeing, business trip, and so on) that circumstantial information of a user has various values. Of course, results of retrieving contents vary with user's movement. Furthermore, the results might be changed by user's occasion.

Also, we need to discuss data mining technique for user's behavior prediction and discovering metadata similarity. Some study can be found in pervasive computing research [14]. And, we need to tune some parameters such as weights of scoring functions and threshold point of matching functions to get better results.

5. CONCLUSION

To realize information retrieval reflecting user's circumstantial information, we paid attention to the metadata describing user's circumstantial information and content's status. And, we arranged requirements of an information retrieval using metadata to reduce pains of users for manage own metadata. And then, we proposed an information retrieval method consisting of three parts (query generation mechanism, matching part, and scoring part). The proposed method flexibly takes the data mining technique and natural language processing. Resulting from our proposal, users choose their own metadata easily in an information retrieval.

6. REFERENCES

[1] "The World Wide Web Consortium (W3C)," http://www.w3.org/.

[2] "Resource Description Framework Model and Syntax," http://www.w3c.org/RDF/.

[3] "SemanticWeb.org," http://www.semanticweb.org/.

[4] Harry Chen, Tim Finin, Anupam Joshi, and Lalana Kagal, "Intelligent Agents Meet the Semantic Web in Smart Spaces," *IEEE Internet Computing*, November 2004.

[5] Weigou Fan, Michael D. Gordon, and Praveen Pathak, "Discovery of Context-Specific Ranking Functions for Effective Information Retrieval Using Generic Programming, Knowledge and Data Engineering," *IEEE Transactions*, vol. 16, no. 4, pp. 523–527, April 2004.

[6] Akio Sashima and Koichi Kurumatani, "Seamless Context-Aware Information Assists Based on Multiagent Cooperation," *AESCS'02*, pp. 39–46, 2002.

[7] Santtu Toivonen, Juha Kolari, and Timo Laakko, "Facilitating Mobile Users with Contextualized Content," *AIMS2003*, October 2003.

[8] Jiawei Han, Yongjan Fu, and Wei Wang et al, "Dbminer: A system for mining knowledge in large relational databases," *2nd International Conference on Knowledge Discovery and Data Mining*, pp. 250–255, August 1996.

[9] "Google," http://www.google.com/.

[10] Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *Stanford Digital Library Technologies, Working Paper*, 1998, SIDL-WP-1999-0120.

[11] "Simple Object Access Protocol (SOAP)," http://www.w3.org/TR/soap/.

[12] "XSL Transformations (XSLT)," http://www.w3.org/TR/xslt/.

[13] "MathML," http://www.w3.org/Math/.

[14] Ugo Galassi, Attilio Giordana, and Dino Mendola, "Learning user profile from traces," *The 2005 Symposium on Applications and the Internet Workshops (SAINT-W'05)*, January 2005.