Scientific Research

Dynamic ontology-based user modeling in personalized information retrieval system

Dangxiang Ai[1], Hui Zuo[2], Gaoyong Liu[1]
[1]School of Management, Guangdong University of Technology, Guangzhou, China
[2]School of Economics and Tade, Guangdong University of Technology, Guangzhou, China
aidx78@gmail.com, zuohui_ne@163.com, gaoyongliu@yahoo.com.cn

Abstract: Personalized retrieval is the inevitable trend of information services, which can meet users' diversified and specialized knowledge requirements. The learning of user features and the building of user model are essential to realize the personalization in information retrieval system. In this paper, we introduced ontology into this research, and study the representation of user interests using concepts and relations, and based on which, develop a dynamic approach of modeling. We analyzed the main principle and the two key steps of this approach including user features primitive learning and deep mining. The application of the model is also discussed at the end of the paper. Our study shows that dynamic ontology-based user modeling can improve the retrieval quality by better representing, discovering and utilizing user implicit interests.

Keywords: ontology; user modeling; information retrieval; personalization

1. Introduction

Information Retrieval System (IRS) has taken a crucial role in the modern economic society to meet people's knowledge demands. However it is a big problem to handle the diversification of IRS users. People with various habits and backgrounds have different interests in knowledge and different understanding even on the same piece of information. It is hard to retrieve accurate and valid information without considering the user's special requirements. The only way to resolve this problem is personalized information retrieval, which offers searching result according to each user's particular features [3]. The principal challenges of personalization are user features acquiring and modeling. In current studies, there are two main approaches to build user model: static and dynamic. The former is easy to implement. When a user registers, let him answer some questions about his age, profession, preference, education experience etc, from which user's possible interests are analyzed and the model is built once for all. This static approach has obvious limitations of bad adaptability to user features change. Therefore, the dynamic modeling approach has become popular in current IRSs, which requires the system to automatically track users' behavior, obtain user interests by machine learning and adjust user model regularly. This method is more objective without more burdens to the user [2].

In this paper, we utilized ontology technology to propose a new way of dynamic user features modeling. Ontology is a hot issue in recent research of machine learning. The use of ontology enables to define concepts and relations representing user interests in structured and explicit way. In the following section 2, an ontology-based user model is exhibited. Section 3 describes the process of user features learning and modeling. Section 4 discusses one possible way of applying the model in IRS. Finally, section 5 provides a summary of the research results and suggestions for future work.

2. Ontology-based user model (OBUM)

User Features can be described in many ways. The most common one is weighted keyword list. But the list is linear, and keywords in the list have no relations with each other. It is difficult to describe and discover more implicit user interests. So more and more IRSs apply structured concept space to build user model, and ontology is one of the effective methods.

Ontology-based user model (OBUM) is built by adopting ontology elements, including Concept, Property, and Relation etc. Concepts are used to describe user's main interests, and we call them User Feature Concepts (UFCs). Properties are used to represent the natures of the UFCs, like Chinese and English name, subject category, relevance degree to user and so on. Relations are used to state the possible correlations among UFCs. The whole OBUM is a concept hierarchy of a certain field [9].

Figure 1 illustrates an example of OBUM in the field of computer application. In this model, Every UFC has a property of "User Relevance Degree (URD)" (see the value behind each UFC in Fig 1.), which reflects how much the UFC is relevant to the user's interests. The higher the value of the URD is, the more interested in the UFC the user will be. On the other hand, if the value of the URD is zero, the user has no interests to the UFC. The model in Fig 1 shows that the user is mainly interested to two domains: "Natural Language Process" and "Automatic Translation". Furthermore, three relations among UFCs are defined in the model, which are "super domain of", "equivalent to", and "related to". "Super domain of" relation represents that one concept is in the domain of another concept. For instance, "Expert Sys-

tem" belongs to the domain of "Artificial Intelligence", so we can say that "Artificial Intelligence" is "super domain of" "Expert System". "Equivalent to" relation represents that one concept has almost the same domain of another concept, for example, "Computer Vision" is "equivalent to" "Machine Vision". "Related To" relationship represents that the domains of two concepts are partly overlapped, but not completely same, for example, "Natural Language Process" is "related to" "Natural Language Interface".

3. Dynamic user features learning and modeling

We design an intelligent and dynamic approach to learn user features and build the OBUM. The system constantly traces user's behavior, automatically acquires all UFCs and calculates URD value of every UFC.

3.1. Modeling approach

Figure 2 illustrates the principle of our modeling approach. First and foremost, a reference ontology is needed. It's also a concept hierarchy of a certain field, containing all the concepts and relations in this field. The reference ontology acts as the initial model of all users and the URD values of all the concepts are set to zero at the beginning. The system will adjust the concepts and their URD values for each user during the modeling process.

An OBUM is built according to the following steps:

Step1: User agent observes user's retrieval behavior and record the retrieved documents information into user cache.

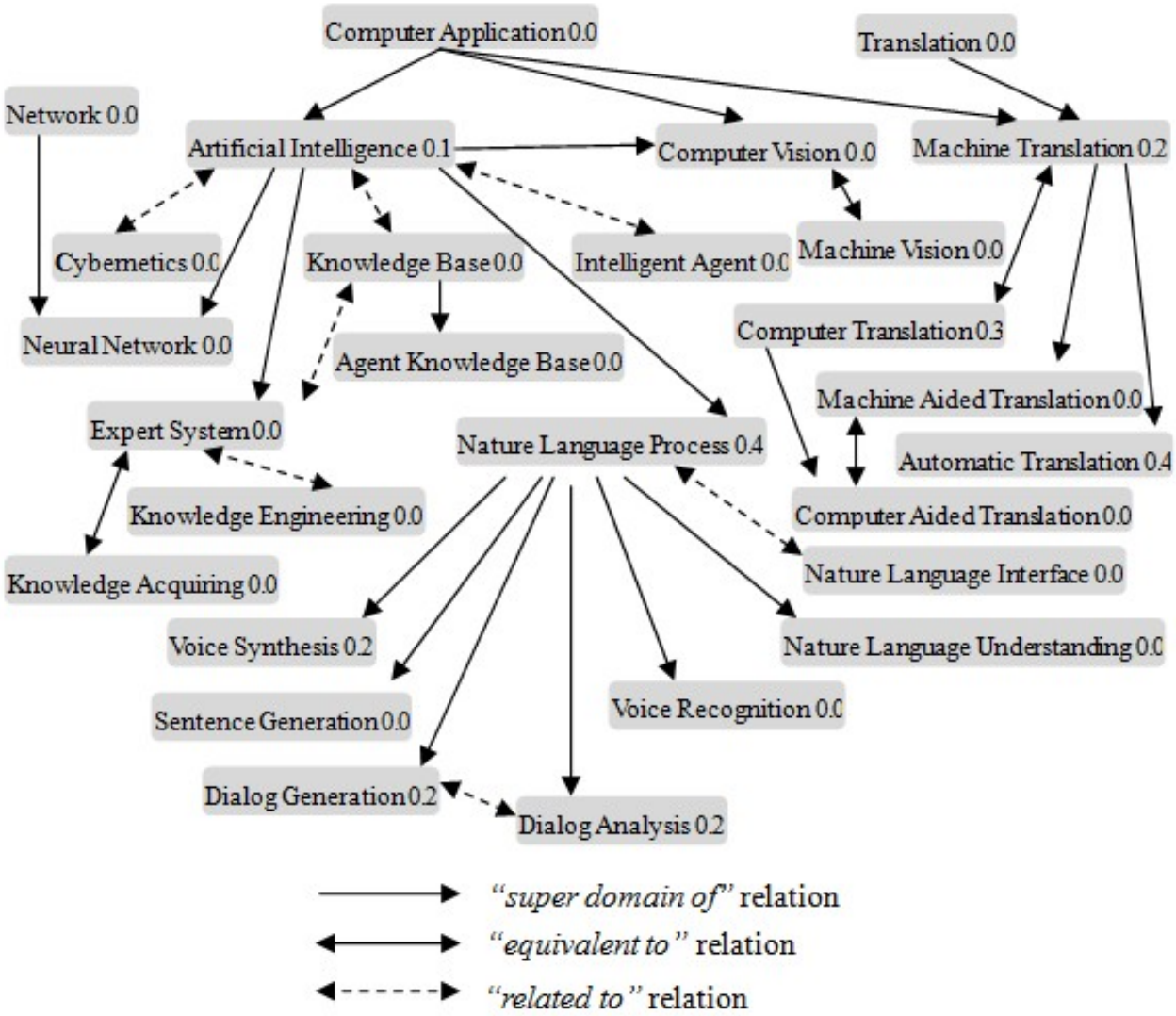

Figure 1. Example of OBUM.

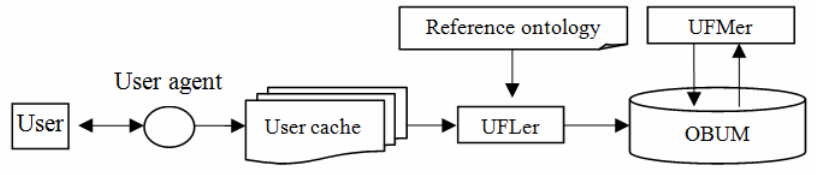

Figure 2. Modeling approach.

Step2: User Features Learner (UFLer) executes user features primitive learning. It accesses user cache to get the documents information, calculates this user's UFCs and their URD values by applying machine learning algorithm. The result is stored in the User's initial OBUM.

Step3: User Features Miner (UFMer) implements user implicit features mining. By analyzing the concept relations in the OBUM, it adjusts URD values and discards irrelevant concepts to optimize the model.

In the next two sections, we will discuss step2 and step 3 in details.

3.2. User features primitive learning

1) Data training

Before learning user features, the UFLer must be trained. The training data are "relevant" documents manually labeled to each concept in the reference ontology. "Relevant" means that the content of the document reflects the domain of the concept. Documents labeled to different concepts should not be the same. The aim of data training is to create a vector for each concept and calculate its normalized weight.

Suppose that c is a concept in the reference ontology and its relevant documents set is DC. Let VC stand for the vector of concept c. At the beginning, VC is composed of all the lexical items appeared in the documents in DC, i.e. VC: $(t_1,t_2,t_3,\ldots\ldots)$. For each item t_i, its importance are decided by the two main factors: appearance frequency and locations in the documents. The higher the frequency is, the more important t_i will be. At the same time, if t_i appears at different locations in the documents, its significance is different too. When t_i locates in the title, abstract or keywords of the document, it is of greatest importance. When t_i locates in the beginning paragraph, ending paragraph, or some paragraph's first and final sentences, it is of secondary importance. Therefore, we set factor ρ_{loc} to reflect location weight. The value of ρ_{loc} is usually between 0.5 and 1.0, and the more important the location is, the higher the value. Take both frequency and location into account, the UFLer calculates t_i's weight WT_i according to (1), such that:

$$WT_i = (\sum_{1}^{tf_i} \rho_{loc}) \times idf_i \qquad (1)$$

with:

tf_i : Total appearance frequency of t_i in DC,

$\sum_{1}^{tf_i} \rho_{loc}$: Accumulated location weight of ti in DC,

$idf_i = \log_2(\frac{ndoc}{df_i})$: Inverse document frequency of t_i,

$ndoc$: Quantity of documents in DC,

df_i : Quantity of documents containing t_i in DC.

Since relevant documents of concepts have different lengths, concept vectors contain different number of lexical items, which cause the deviations in the weight calculation. So we normalize the weight of t_i according to (2), such that:

$$NWT_i = \frac{WT_i}{\sum_i WT_i} \qquad (2)$$

NWT_i is the normalized weight of t_i. Keep down the giving number of items with the highest NWT_i values to form the final vector VC. After all the concept vectors with normalized weight have been generated and stored in the reference ontology, Data training is finished.

2) Dynamic learning

The trained UFLer can execute dynamic user features learning. When a new user registers, the UFLer will create a copy of reference ontology to act as this user's initial OBUM. And then the UFLer will regularly read from the cache the user's visiting log, including the contents and lengths of retrieved documents, user durations of browsing and so on. From the log, the UFLer will generate a user relevant documents set DU, and for each document in DU, a vector and its normalized weight will be calculated using the same approach for concepts discussed above.

With the vectors of documents in DU, the UFLer adjusts the UFD values of the concepts in initial OBUM, and therein the calculation of the relevant degree (RD) between concept and document is the crucial step. With concept c in OBUM and document d in DU, their RD can be computed according to (3), such that:

$$RD(d,c) = tl\_factor \times \frac{\sum_{i=1}^{n}(NWD_i \times NWT_i)}{\sqrt{\sum_{i=1}^{n} NWD_i^2 \times \sum_{i=1}^{n} NWT_i^2}} \qquad (3)$$

with:

NWT_i : Normalized weight of VC (the vector of concept c),

NWD_i : Normalized weight of VD (the vector of document d),

n : Quantity of the items in VC,

978-1-935068-43-3 © 2010 SciRes.

$tl\_factor = \log_2(\frac{doc\_length}{time})$: Adjusting factor,

doc_length : Length of document *d*,

$time$: User browsing time at document *d*.

The concepts with the highest RD values reflect the user interests in document *d*. Accumulate the RD value to the URD value of the concept, thus with continual learning, the concept would be highlighted with high URD value and become an important UFC.

3.3. User features deep mining

The OBUM after primitive learning can only describes the user's surface features, with illogical UFCs and even noises. Furthermore, a single user's interest field is usually limited, and there is no need to keep the whole reference ontology in the OBUM. In order to better reflect implicit user features, save the memory space and quick the model processing speed, we utilize the UFMer to mine user's deep interests and integrate the OBUM.

The principle of UFMer is that, user's interest field is relatively centralized, so the UFCs tend to convergence. We can embody this convergence by analyzing the relations among concepts. For example, the current UFCs are dispersed, but some of their correlated concepts may lap over. These overlapped concepts usually represent unexpressed user's deep interests.

Since the structure of the OBUM is similar to that of neural network, we use the heuristic spreading algorithm with the following steps to refine the OBUM:

Step 1: Select the concept with lowest but not zero URD value from the OBUM to be the original node for spread. Set the current original node as N_j, and its URD as w_j;

Step 2: Randomly activate a concept node correlated to N_j, increase the URD value of the activated concept according to constraint conditions. Set the current activated node as N_{j+1}, and its primer URD as w_{j+1}, then the new URD of N_j should be $w_j + factor_{(relation)} \times w_{j+1}$, therein $factor_{(relation)}$ is a condition factor decided by the relations between N_j and N_{j+1}. The value of $factor_{(relation)}$ can be sorted according to:

$factor_{(equivalentTo)} > factor_{(superDomainOf)} > factor_{(relatedTo)}$;

Step 3: Repeat step 2 until all concepts correlated to N_j have been activated;

Step 4: Repeat step 1 to step 3 until all concepts with non-zero URD value in the OBUM have been spread as the original node;

Step 5: Discard the concepts with URD values lower than the threshold and reserve the concepts hierarchy branch with high URD values.

Using the steps above, the OBUM shows in Fig.1 can be optimized to the one in Figure 3. We can see that the user's interest field is more outstanding and some isolated concepts like "cybernetics" are wiped off as noises.

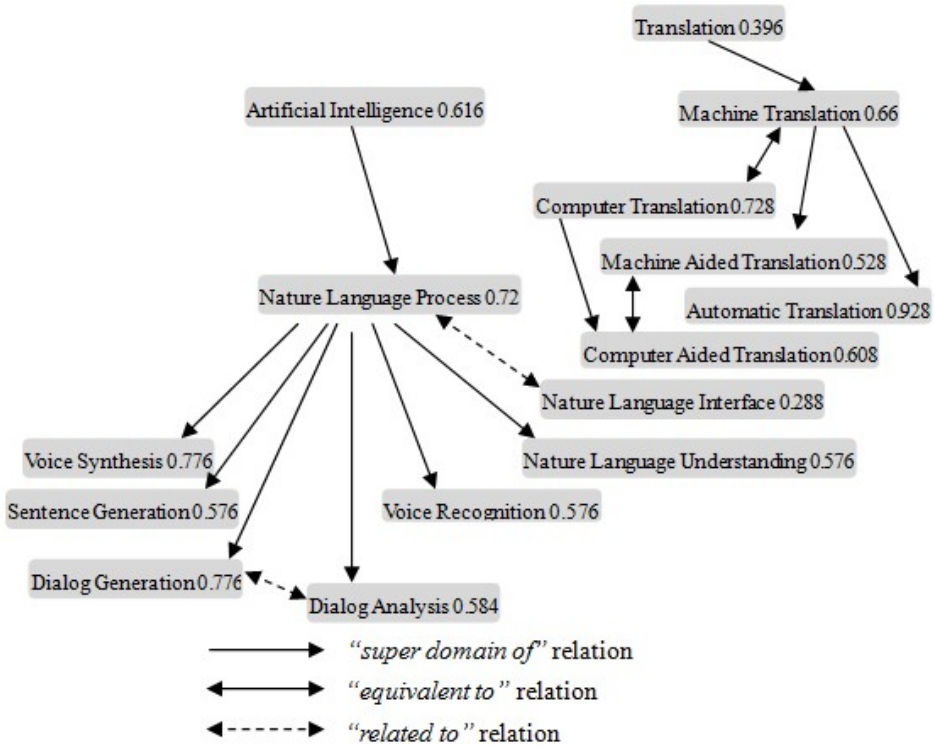

Figure 3. Optimized OBUM.

4. Application of OBUM in IRS

The OBUM can be used to modify the retrieval result to better meet the IRS user's personalized demand.

Suppose *Dr* is the initiative retrieved document set. Relevant concepts for each document in *Dr* are acquired by using the approach similar to UFCs learning. The IRS will check the OBUM to get these concepts' URD values and calculate the document importance to the user.

Assume that *dr* is a retrieved document in *Dr, wr* is the original weight of *dr, Cr* is the relevant concept set for *dr*, and ufd_k is the UFD value of the concept c_k in *Cr,* then the new weight of *dr* can be calculated according to (4), such that:

$$new\_wr = wr * (0.5 + \frac{1}{n}\sum_{k=1}^{n} ufd_k) \quad (4)$$

with:

n: Quantity of concepts in *Cr*.

We can see from formula (4) that the user interest is considered in the new weights calculation. Based on these weights, the retrieval result can be resorted and with more precision.

5. Summary and future work

In order to realize personalization in information retrieval system, we designed an approach to automatically create ontology-based user model in this paper. The approach can effectively solve the problem of user diversification and discover implicit user interests. We firstly discussed the user features representation with ontology elements and the structure of the model, and then we described dynamic modeling steps in details, including user features primitive learning and deep mining, finally we discussed the application of the model to improve the retrieval accuracy.

Our future work will be focused on the further expansion of ontology-based user model by enriching the concept relations and using more sophisticated inference mechanism. There may be also other studies on different combinations of modeling technologies.

6. Acknowledgment

This research is supported by Natural Science Foundation of Guangdong Province (10451009001004318), Eleventh-Five Year Plan Project in Philosophy and Social Science of Guangdong Province (08YM-02), and Philosophy and Social Science Project of Guangzhou (090071).

References

[1] Wu Huijuan, Yuan Fang. "Research of Technologies on Personalized Information Service". Computer Technology and Development, 2006, 16(2), pp. 32-34, 37.

[2] Ji Meijun. "Research on Problems of User Modeling Oriented to Personalized Service". Journal of Information, 2006, 25(3), pp. 77-79.

[3] Wu Lihua, Liu Lu. "User Profiling for Personalized Recommending Systems-A Review". Journal of the China Society for Scientific and Technical Information, 2006, 25(1), pp. 55-62.

[4] Zuo Hui, Zhang Yufeng, and Ai Danxiang. "Research on Ontology-Based User Interests Mining in Personalized Knowledge Services". Journal of the China Society for Scientific and Technical Information, 2008, 27(1), pp.18-23.

[5] S. Gauch, J. Chaffee, and A. Pretschner. "Ontology-based Personalized Search and Browsing". Web Intelligence and Agent Systems, 2003, 1(3-4), pp. 219-234.

[6] T. Kurki, S. Jokela, R. Sulonen, and M. Turpeinen. "Agents in Delivering Personalized Content Based on Semantic Metadata". Proceedings of 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace, Stanford, USA, 1999, pp. 84-93.

[7] R. Studer, V. Richard Benjamins, and D. Fensel. "Knowledge Engineering: Principles and Methods". Data and Knowledge Engineering, 1998, 25(102), pp. 161-197.

[8] N.Guarino, C. Masolo, and G. Veter. "OntoSeek: Content-based Access to the Web". IEEE Intelligent Systems, 1999, 14(3), pp. 70-80.

[9] S. Decker, M. Erdmann, and D. Fensel. "ONTOBROKER: Ontology Based Access to Distributed and Semi-Structured Information Web Semantic Issues in Multimedia Systems". Proceedings of DS-8, Boston, 1999, pp.351-369.

[10] F. N. Natalya, L. M. Deborah. "Ontology Development 101: A Guide to Creating Your First Ontology". http://ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html, 2010-05-20

[11] A. G. Perez, V. R. Benjnmius. "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods". *Proceedings of the IJCAI'99*, deAgosto, Estocolmo, 1999, pp.1-5.