Modeling Hierarchical User Interests Based on HowNet and Concept Mapping

Yihong Li[#1], Fang Li[#2]

[#] *Dept. of Computer Science & Engineering, Shanghai Jiao Tong University*
No.800 Dong Chuan Rd. Shanghai 200240, P.R. China

[1]flyrain@sjtu.edu.cn

[2]fli@sjtu.edu.cn

***Abstract*—Modeling user interests plays an important role in personalized service on the Internet. Many systems use classification methods to construct the user profile to represent user interests. However, it is difficult to cover all individual interests and capture the changes of user profiles. This paper introduces a method with two steps to solve the problem. In the first step it clusters the viewed web pages to extract individual interests by extending the vector of VSM with sememes from the HowNet. The second step is to map each interest cluster to the reference taxonomy to model hierarchical user interests based on similarity calculation of keywords and the extended vector. Experiments show that the method can model hierarchical user interests with a promising result. When a new interest emerges, it does not need any adjustment like collecting new training data or rebuilding the classifier. It can capture the diversified user interests and map to an interests taxonomy.**

I. Introduction

It is known that Information on the Internet grows exponentially. Users always feel that they are drowning in a sea of information. How to provide useful information for different individual user has become a critical problem. User interests are very important to personalized search, collaborative filtering and recommendation systems. How to model user interests is a key step for providing personalized service. The common representations of user interests are keyword profile and concept profile [1].

Keyword profiles are represented as sets of weighted words extracted from the web pages that a user browsed. Each set of words represents diversified user's interests. However, many sets of keywords have some relationships. The flat structure like keywords profile can not model user interests precisely and correctly.

Hierarchical concept profiles can provide a more standard structural way to model user interests. Node of the concept profile can be considered as an abstract topic that user may be interested in, rather than set of words. The concept hierarchy describes a standard way to model user interests and their relationships. It is close to the human conception.

There are two methods to model hierarchical user interests on the Internet: hierarchical classification and clustering. Classification method needs to train a classifier to represent all individual user interests. Hierarchical clustering methods can achieve individual user interests. However, it is difficult to provide a standard personalized service. Based on our prior research [2], how to capture individual interests and model hierarchical user interests with concept profiles is our research aim.

Our interest profile construction includes two steps: clustering the web pages and mapping each cluster to a concept of the reference taxonomy. Due to the complexity of natural languages, deep content processing is impossible without semantic information. There are synonym, polysemy and other language problems, which challenge in information processing. Therefore, semantic knowledge platform such as Knowledge Grid [3] plays an important role in semantic processing. In our solution, we attempt to use the Hownet [4], a Chinese knowledge base, to capture the semantic information in the traditional vector space model in order to acquire user interests. By similarity calculation of keywords, user interests generated can be mapped to the reference user interest ontology.

The rest of the paper is organized as follows. In Section II we discuss about the cluster construction using clustering algorithm based on HowNet. Then we talk about our concept mapping method in Section III. Our experiment is shown in Section IV. The related work is in Section V and a conclusion in Section VI.

II. Cluster Construction

The content of browsed web page is a good indicator to user interests. Individual user interests can be found based on the result of clustering web pages. The cluster construction consists of three steps. The first step is to extend vector space model based on HowNet to represent the semantic content of web pages. The second step is page clustering. The third step is cluster refinement.

A. Web Page Representation based on HowNet

The content of a web page can be represented as a vector in the Vector Space Model (VSM). In order to solve the sparseness of the VSM vector, the HowNet sememes are used to extend the vector. Each lexical item of HowNet is expressed with a set of ***sememes***.

Sememes are the atomic semantic elements. For example: A Chinese word "农田" (means farmland) in the HowNet can be described as:

DEF= land|陆地,@planting|栽植,#crop|庄稼,agricultural|农

There are four sememes related to the word "农田": "land|陆地", "planting|栽植", "crop|庄稼" and "agricultural|农". The symbol '|' separated the Chinese word and the English word of a sememe. '@' here means that "farmland" is the "location" for "planting". '#' here means that "farmland" and "crop" have some co-relationship.

Synonyms usually share the same sememes in the HowNet. The word "田地" also means farmland. It has the same sememes as the word "农田". If two words have been defined as containing the same sememe X, these two words are called ***related words*** of X. For example, "农田" and "田地" are the related words of sememe "land|陆地".

Each sememe in HowNet belongs to a category. The most important categories are ***event category, entity category*** and the ***secondary feature category***. Sememes have their depth information in the up-down relationship (see Fig. 1). General sememes are close to the root and special sememes are deep down in the tree. For example, the depth of sememe "crop|庄稼" is 6, while the depth of sememe "animate|生物" is 4.

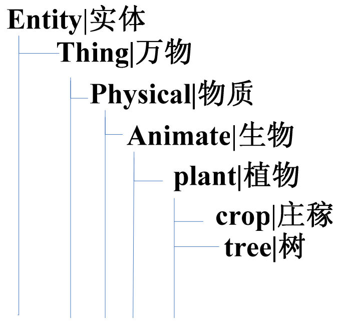

Fig 1. The tree of HowNet up-down relationship

Since sememes have so much useful information, important sememes (whose categories are entity, event and secondary feature) are extracted from each word. They are added to the document vector as a new dimension.

Let D = {$d_1,d_2,\dots,d_n$} denotes a set of documents that a user has browsed. The set T = {$t_1,t_2,\dots,t_m$} denotes the words of all documents. Of course, the stopwords and words with DF (document frequency) less than 2 are filtered. These words are not included in T. Let S = {$s_1,s_2,\dots,s_p$} denotes the set of all the useful sememes related to any word in T. The HowNet extended document vector is defined as:

Definition 1 (Extended Document Vector) An extended document vector of the i-th document d_i is defined as vector

$$v_i = [tfidf(t_1), tfidf(t_2), ..., tfidf(t_m), w(s_1), w(s_2)..., w(s_p)],$$

where tfidf(t_i) is the TFIDF weight of word t_i, w(s) is the weight of the sememe s. Each sememe s has its related words t in the document. The weight of the sememe s is defined as

$$w(s) = \lambda(s)\cdot wdepth(s)\cdot \sum_{t\in RelatedWords(s)} tfidf(t). \quad (1)$$

As Formula 1 shows, three kinds of contributions of sememe s are considered.

1) *The category of sememe:* In HowNet, there are three most important categories such as entity, secondary feature and event. Entity category is about the nouns, while the event category is about the verbs. Secondary feature category describes the domain of a word. Only sememes in these categories are used to extend the document vector.

Suppose category(s) denotes the category of sememe s, then the category contribution of sememe s is defined as:

$$\lambda(s) = \begin{cases} 0.9 & category(s) = entity; \\ 0.8 & category(s) = secondaryFeature; \\ 0.7 & catergory(s) = event. \end{cases} \quad (2)$$

2) *The depth of sememe:* Special sememes can contribute more than the general sememes. Special sememes are deep down in the up-down relationship tree (see Fig. 1). The deeper a sememe is, the bigger the contribution of the sememe is. The depth contribution of a sememe s is defined as:

$$wdepth(s) = \begin{cases} 0 & depth(s) < \min Depth_k \\ \dfrac{depth(s)}{\max Depth_k} & depth(s) \ge \min Depth_k \ \& \ depth(s) \le \max Depth_k \\ 1 & depth(s) > \max Depth_k \end{cases} \quad (3)$$

The depth(s) is the sememe depth of sememe s. The parameter $\mathrm{minDepth}_k$ and $\mathrm{maxDepth}_k$ are for adjustment. Because the words which are too general are useless, the sememes which have the depth less than $\mathrm{minDepth}_k$ are ignored. Then we suppose that the sememes which have the depth more than $\mathrm{maxDepth}_k$ have equal contributions. Because sememes in different categories can't be treated equally, different values are assigned to $\mathrm{minDepth}_k$ (or $\mathrm{maxDepth}_k$) for different sememe categories.

3) *The TFIDF weights of related words:* Considering a sememe have some related words. If these words are mentioned for many times in a document, the sememe should be important. Suppose the set of related words of sememe s is RelatedWords(s), we accumulate the TFIDF weights of each related word t in RelatedWords(s) as a contribution.

The first and third contributions are defined from the Zhi Cai's document representation [5]. We implement the depth of sememe as another contribution of a sememe.

B. Clustering

Based on the extended vector of each browsed web page, the clustering process is applied to the vectors. The cluster algorithm used is the Sphere K-means [6] for its fast and efficient features. The Kaufman approach (KA) [7] is chosen as the initialization method of the Sphere K-means.

C. *Cluster Refinement*

After all the clusters have been generated, the clusters are refined.

First, the top 3 nouns or verbs with the highest weights in each cluster are checked. If two clusters have the same words in the top 3 words, we assume these two clusters share the same topic and merge them to be one cluster. Then the centroid is reconstructed for the new cluster. The process is repeated until no such two clusters need to be merged. Finally we get the clusters to represent diversified user interests.

III. Mapping Cluster to Concept based on HowNet

A reference taxonomy T has been borrowed from the IAsk site of Sina.com (http://dir.iask.com). Suppose c is a cluster, o is a concept node of the reference taxonomy. The mapping process is to map each c to the concept node o of the taxonomy.

In our method, clusters and concept nodes are regarded as documents. Both of them can be represented by the extended document vector which defined in the section II. The extended vector can represent the semantic content of cluster and concept nodes. The cosine function is used to calculate the similarity between the vectors.

Due to various web page contents, the sparse of the extended vector still exists. For example, the word “图书(book)” have sememes “publications|书刊” and “mass|众”, while another word “短篇小说(short story)” have a sememe “readings|读物”. These two words are similar. However they have no common sememes. The cosine of extended vectors can’t discover the similarity between these words.

In order to solve the problem, the HowNet similarity calculation from Qun Liu [8] is used to evaluate the similarity between keywords. In the above example, the up-down relation between the sememes such as “publications|书刊” and “readings|读物” will be considered semantic similar according to the calculation result.

For this reason, the keywords are extracted from each cluster and concept. Each time we get one from cluster and one from concept. Then the HowNet similarity is calculated between keywords of cluster and concept. The HowNet similarity values can be considered as part of similarity between cluster and concept.

In the following, first the representation of cluster and concept are defined. Then the similarity calculation between cluster and concept is defined. Finally an algorithm to map each cluster to a concept node of a taxonomy T is discussed.

A. *Cluster Representation*

In mapping process, each cluster c is represented as an extended vector CV (see Definition 2) and a keyword set CW = $\{cw_1, cw_2, \ldots cw_{nc}\}$. Each cw_i is a keyword from the cluster.

Definition 2 (Extended Vector of a Cluster) The extended vector of a cluster c is defined as the centroid of the cluster:

$$CV = \frac{1}{|c|}\sum_{d_i \in c} v_i \quad , \tag{4}$$

where |c| is the number of pages in c, d_i is a document in cluster c, v_i is the extended document vector of d_i (see Definition 1). The extended vector considered the words and sememes of the whole content in each document. It provides a global view of the cluster.

Keywords (CW) of the cluster c are also extracted for HowNet similarity calculation. They are the top 30 words (either nouns or verbs) extracted from the extended vector of cluster. These words are used to be compared with keywords of concept according to the HowNet similarity. The Keyword sets are used to evaluate the semantic meaning between a cluster and a concept.

B. *Concept Representation*

Each concept o of a reference taxonomy T is regarded as a document. The labels in the subtree of the concept o can be regarded as the content of the “document” o.

Like cluster, the concept o is represented as an extended vector OV and a keywords set OW = $\{ow_1, ow_2, \ldots ow_{no}\}$.

Definition 3 (Extended Vector of a Concept) An extended vector of a concept o is defined as:

$$OV=[tfidf(t_1),tfidf(t_2),\ldots,tfidf(t_m),w(s_1),w(s_2),\ldots,w(s_p)],$$

where each word t_i belongs to the words set T, each s_j belongs to the sememes set S (see Section II-A), $tfidf(t_i)$ is the TFIDF weight of word t_i in the “document” o, $w(s_j)$ is the weight of sememe s_j (see Formula 1).

Suppose the taxonomy T is as Fig. 2, then the “document” of concept “Basketball” is “Basketball NBA CBA”. The “document” of concept “Sports” is “Sports Football Basketball Ping-Pong NBA CBA ...”.

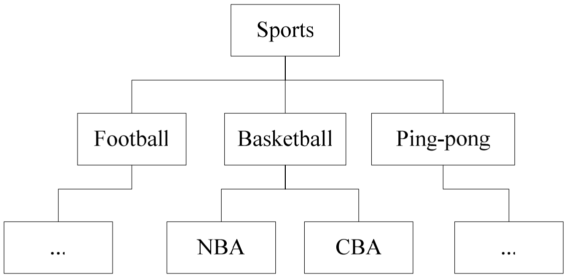

Fig. 2 A snippet of a reference taxonomy T

In order to construct the keywords set of a concept, the labels which are nouns and verbs from the subtree of the node are selected. Considering the complexity of HowNet similarity computation, only labels from the top 2 levels of the subtree are chosen. For example, the keywords set of the concept “Sports” is {Sports, Football, Basketball, Ping-Pong} in the Fig. 2.

C. Similarity calculation between Concept and Cluster

After the above two processes, each cluster c is represented as an extended vector CV and a keyword set CW = $\{cw_1, cw_2, \ldots cw_{nc}\}$. Each concept o is represented as an extended vector OV and a keyword set OW = $\{ow_1, ow_2, \ldots ow_{no}\}$. The similarity between concept and cluster can be calculated according to these extended vectors and keywords.

The extended vector similarity between concept o and cluster c is calculated as definition 4.

Definition 4 (Extended Vector Similarity) The extended vector similarity between concept o and cluster c is defined as:

$$ExtendedVectorSim(o,c) = \cos(OV, CV) = \frac{OV \cdot CV}{\|OV\| \cdot \|CV\|}, \quad (5)$$

where OV is the extended vector of concept o, CV is the extended vector of cluster c. Here the cosine function is used to evaluate the similarity between extended vector OV and CV.

The keywords similarity between concept o and cluster c is calculated as definition 5.

Definition 5 (Keywords Similarity) The keywords similarity between concept o and cluster c is defined as:

$$\begin{aligned} KeywordsSim(o,c) &= \beta \cdot MaxKeywordsSim(o,c) \\ &+ (1-\beta) AverageKeywordsSim(o,c), \end{aligned} \quad (6)$$

where AverageKeywordsSim(o,c) is the average value of the HowNet similarities [8] between each pair of keywords, MaxKeywordsSim(o,c) is the maximum value of the HowNet similarities between each pair of keywords, β is a weight parameter for adjustment. It is between 0 and 1.

The HowNet similarity is computed between each pair of keywords from cluster and concept. The maximum value and average value of these HowNet similarity values are then combined together to be the keywords similarity.

The maximum value of the HowNet similarities between each pair of keywords is calculated as

$$\begin{aligned} &MaxKeywordsSim(o,c) \\ &= \underset{i,j}{MAX}\,(HownetSim\,(ow_i, cw_j)), \end{aligned} \quad (7)$$

where ow_i is a word in the keywords set OW of the concept o, cw_j is a word in the keywords set CW of the cluster c, HownetSim(ow_i, cw_j) is the HowNet similarity between word ow_i and cw_j.

The average value of the HowNet similarities between each pair of keywords is calculated as

$$\begin{aligned} &AverageKeywordsSim(o,c) \\ &= Y \sum_{i=1}^{|OW|} \sum_{j=1}^{|CW|} tfidf\,(cw_j) HownetSim(ow_i, cw_j), \end{aligned} \quad (8)$$

where |OW| is the number of the keywords in concept o, |CW| is the number of the keywords in cluster c, tfidf(cw_j) is the TFIDF weight of word cw_j in the extended vector of cluster c,

$$Y = \frac{1}{|OW| \sum_{j=1}^{|CW|} tfidf\,(cw_j)}$$

is a normalization factor. Considering the different importance of the keywords in the cluster, the TFIDF weight of the keyword is implemented when computing the average value of the HownNet similarity.

The similarity calculation between concept and cluster is based on the extended vector similarity (see Definition 4) and the keywords similarity (see Definition 5).It shows in the formula 9. α is the weight parameter for adjustment. We set it as 0.9, 0.1, 0.1 corresponding to the first, second and third levels of the concepts.

$$\begin{aligned} Similarity\,(o,c) &= \alpha \cdot ExtendedVectorSim\,(o,c) \\ &+ (1-\alpha) KeywordsSim\,(o,c). \end{aligned} \quad (9)$$

D. Mapping Algorithm

```
Algorithm 1 ConceptMapping(c, r)
Input: c: A cluster; r: The root concept in the interest taxonomy T for mapping.
Output: o: A concept node in taxonomy T which mapped to cluster c.
/*
This function maps cluster c to a concept o in the subtree of r.
*/
o ← r; /*Initialization*/
if (hasChild(r)) /*node r has child in taxonomy T*/
    topKConcepts ← top K children of concept r which have the highest similarity with cluster c;
    /*
    Return the leaf node o which is most similar to cluster c.
    */
    for each o_i in topKConcepts
        o_temp ← ConceptMapping(c, o_i);
        if Similarity(o_temp, c)>Similarity(o, c)
            o ← o_temp;
        end if
    end for
end if
return o;
```

Fig. 3 The algorithm of concept mapping between cluster and taxonomy

The mapping algorithm is shown in Fig. 3. Given a cluster c and root concept node r, the algorithm searches in the subtree of concept r, and it returns a concept o mapped to the cluster c.

The mapping process is a top-down method. First it gets the top K children of concept r with the highest similarity about cluster c. These children are regarded as the set 'topKconcepts'. Then it runs the same mapping process in the subtree of each node in 'topKConcepts'. The process returns a leaf node for each subtree. These K leaves are regarded as candidates. Finally it chooses the candidate o which is most similar to the cluster c. The 'Similarity' function evaluates the similarity between a concept and a cluster.

It chooses K candidates of concepts to prevent the early wrong mapping in the ancestor nodes. The parameter K is 2 in our system and the experiment.

IV. EXPERIMENT

To evaluate the performance of our method for constructing hierarchical user interests, an experiment is conducted.

Using our IE plug-in, we first collected 13 volunteers' browsing histories of about 2-week information. The total number of the visited web pages is about 3350 web pages which cover different topics including politics, culture, economy, science, entertainment and so on.

For concept mapping, a topic hierarchy is extracted from the IAsk site of Sina.com (http://dir.iask.com) as our reference taxonomy. The taxonomy contains 3 levels of concepts. After filtering some useless concepts, we get about 3662 concepts in the taxonomy.

Then our clustering process is performed to each volunteer's browsing history. The system generated 5-12 clusters for each volunteer. Three mapping methods (including our mapping method) were applied on these clusters. These methods are listed as follows:

1) *The Baseline method:* The method only uses the classic Vector Space Model (VSM) vector to represent a cluster and a concept. It uses the cosine value of the vectors to compute the similarity between cluster and concept.

2) *The ExtendedVector method:* The method uses our extended vector (see Definition 2 and Definition 3) to represent a cluster and a concept. It uses the 'Extended Vector Similarity' (see Definition 4) to compute the similarity between cluster and concept.

3) *The ExtendedVector+Keywords method:* It is the one our system chooses. It uses both extended vector and keywords set to represent a cluster and a concept. Formula 9 was chosen to compute the similarity between cluster and concept. The parameter α of the similarity is set to 0.9, 0.1, 0.1 corresponding to the first, second and third levels of the concepts.

All three methods use algorithm 1(see Fig. 3) for concept mapping.

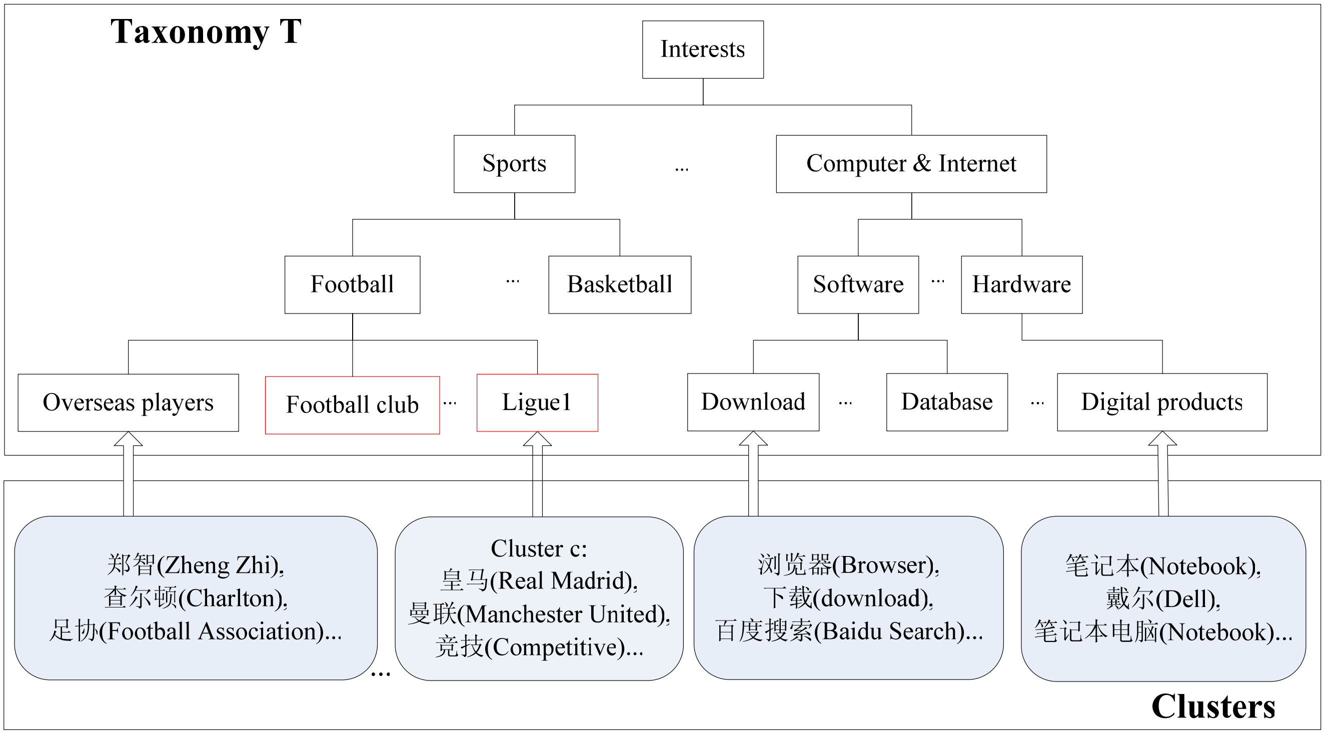

Fig. 4 Part of concept mapping about user U1

TABLE I

THE PRECEISION OF MAPPING (EXTENDEDVECTOR+KEYWORDS METHOD)

User ID	**U1**	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**U8**	**U9**	**U10**	**U11**	**U12**	**U13**
Clusters	10	12	10	12	8	10	7	5	7	8	10	12	11
P (Level$_1$)	1	0.8	0.9	1	0.8	1	1	0.8	1	0.75	1	0.92	0.91
P (Level$_2$)	1	0.7	0.8	0.8	0.6	0.6	0.7	0.8	0.86	0.75	0.8	0.92	0.73
P (Level$_3$)	0.7	0.5	0.6	0.8	0.4	0.6	0.6	0.6	0.86	0.5	0.5	0.58	0.64

TABLE II

MAPPING RESULT OF THE THREE METHODS

Method	**P (Level$_1$)**	**P (Level$_2$)**	**P (Level$_3$)**
ExtendedVector + Keywords	**0.91**	**0.78**	**0.60**
ExtendedVector	0.88	0.61	0.49
Baseline	0.86	0.57	0.41

We evaluate the mapping precisions of the three methods on volunteers' clusters. When a cluster has been mapped to a concept, we say the cluster 'belongs to' the concept and its ancestor. The mapping precision on the i-th level of taxonomy T is defined as:

$$\Pr ecision(Level_i) = \frac{\sum_{c\in C} rightMapping(o^i, c)}{|C|}, \qquad (10)$$

where Level$_i$ is the i-th level of taxonomy T, C is the set of evaluation clusters, |C| is the number of clusters in C, c is a cluster in the set C, o^i is the concept of the i-th level which cluster c 'belongs to'.

rightMapping(o^i,c) returns 1 when the cluster c really belongs to the concept node o^i on the i-th level. Otherwise, it returns 0.

The Fig. 4 shows part of the user U1's interest profile. The cluster c has been mapped to the concept 'Ligue1', in fact the cluster is about the 'Football club'. On the 3rd level of the taxonomy T, concept o^3 is 'Ligue1' which the cluster c 'belongs to'. It is a wrong mapping on this level, so the rightMapping(o^3,c) is 0. On the 2nd level of taxonomy, concept o^2 is 'Football', and cluster c really belongs to it. So the rightMapping(o^2,c) is 1.

Table I shows the mapping result of each user based on the method of ExtendedVector+Keywords. The second row shows the number of clusters which generated for each user. The next three rows show the precision of concept mapping on the 1st, 2nd and 3rd level of the reference taxonomy, indicated by P(Level$_1$), P(Level$_2$) and P(Level$_3$). The precision of 1st level P(Level$_1$) has achieved better result for each user.

Table II shows the mapping result of all three methods. Each row is the precisions of each method, while each column is the precision of the result evaluated on each level.

The precisions on the 1st level are not so different among these methods. The ExtendedVector +Keywords method only outperforms a little with the precision of 0.91 on the 1st level of mapping. The reason is that each concept on the 1st level has a big subtree. There are enough words which can be used for the VSM vector representation (Baseline) or Extended Vector representation on the 1st level.

When comparing the result on the 2nd level and the 3rd level, the ExtendedVector+Keywords method is much better than the other two. The precision of our method is 0.78 on the 2nd level, while the precision of ExtendedVector method is 0.61. The Baseline method has got the precision which is only 0.57 on the 2nd level. The results on the 2nd and 3rd level show that the HowNet similarity of keywords plays an important role in the concept mapping when there are not so much labels to describe concepts.

V. COMPARISON WITH OTHER METHODS

User interests can be represented as a user profile. Many machine learning methods are employed to construct the hierarchical user interests, such as hierarchical clustering algorithms and classification techniques. Their researches indicate that specific interests are very difficult to be identified.

The hierarchical clustering algorithms can construct an interest hierarchy without any reference taxonomy. Kim [9] provides a hierarchical clustering algorithm called DHC to construct user interests. Godoy [10] provides another hierarchical clustering algorithm called WebDCC to construct the user interests. These unsupervised methods do not need any training web pages. The keywords are extracted to represent an interest node in the methods. However the keywords of a node are diversified. It is difficult to build a standard profile for different users.

The classification technique can construct a standard profile based on the reference taxonomy. Ni [11] constructed Chinese weblogger's interests based on text classification. They use the combination of classifiers such as the Naive Bayes Classifier (NB) [12], Support Vector Machine (SVM) [13] and Rocchio Classifier [14]. The system uses the data of SOHO's

Directory [15] to train the classifiers. The top 2 levels in the hierarchical category space are considered as reference taxonomy. The number of training documents is about 28000. PVA System [16] builds the concept-hierarchy-based user profile by classifying the browsed web pages using the vector space model. It use a three-level concept hierarchy containing 55 concepts of Yam search web site [17] as reference taxonomy. Trajkova's system [18] uses the top 3 levels of ODP (Open Directory Project) [19] as reference taxonomy. The web pages related to the ODP categories are collected as training data. A centroid-based document classifier was built to construct user interests. Liu [20] builds a similar system based on the top 2 levels of ODP taxonomy. Misearch project [21] also uses the ODP taxonomy. However, it builds the profiles by collecting and classifying the user search histories rather than the user's browsing history. User interest profiles are built according to a text classifier. All these systems need training web pages to represent each interest. When a taxonomy changes, the training data related to the taxonomy must be changed too.

Our system provides a method to construct a standard hierarchy profile without any training web pages.

Compared with the hierarchical clustering algorithms, our method can generate more standard and precise profile which can capture diversified individual interests. The unsupervised hierarchical clustering algorithms can generate different user interests for each user. However these different hierarchical user interests are difficult to personalized service. On the other side, keywords extracted from web pages can not be guaranteed to represent a standard user interest. Some words are confusing. Using the reference taxonomy, our method can provide a standard interest view of a user.

Compared with the classification technique, our method provides more adaptable way to individual user interests. The classification technique needs training data as its knowledge source. It collects web pages for each interest node of the taxonomy. The training data (web pages) strongly depends on the taxonomy. As taxonomy changes, the large training data must be changed too. Our method does not need to collect training sets or change classifier when user interests change. Only reference taxonomy and HowNet are used as the knowledge sources in our method. When the reference taxonomy changed, our system keeps unchanged.

The semantic information of HowNet has been applied in many applications such as document representation and words similarity calculation. Based on these works, our system modified the document representation method for clustering web pages and considered the keywords similarity definition in the concept mapping process.

VI. Conclusions

In this paper, we propose a method to construct the hierarchical user interests. It doesn't need any training data for each concept node in the taxonomy.

The interest profile construction consists of two steps: clustering the web pages browsed and mapping each cluster to a concept of the reference taxonomy. The HowNet information is used in the clustering process and the concept mapping process. In the clustering process, each document is represented with a HowNet extended vector. In the mapping process, both extended vector and keywords information are used to compare the similarity between a cluster and a concept.

The experiment shows that our method can construct a fine hierarchical user interest profile. It also shows that with the help of HowNet, the keywords similarity plays an important role in the mapping process.

In the future, the quality of extracted keywords will be improved by identifying the relationship among words, clusters and documents. Since the mapping process needs keywords of each cluster, if better keywords have been found to represent a cluster, the mapping process will work more precisely. As we only used the up-down relationship between words, there are still many other kinds of relationships of words in the HowNet. These relationships can be taken into account for the words similarity computation. Based on our method, different reference taxonomy will be experimented. Personalized service and recommendation system would be explored in the near future.

Reference

[1] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," *The Adaptive Web*, pp. 54-89, 2007.

[2] F. Li, Y. Li, Y. Wu, K. Zhou, F. Li, X. Wang and B. Liu, " Combining browsing behavior and page contents for finding user interests" will be published in the *Autonomous Systems -- Self-Organisation, Management, and Control* edited by Mahr & Sheng, published by Springer Verlag 2008.

[3] H. Zhuge, "China's E-Science Knowledge Grid Environment," *IEEE Intelligent Systems*, vol. 19, no. 1, pp. 13-17, 2004.

[4] Z. Dong and Q. Dong. (2000) HowNet. [Online]. Available: http://www.keenage.com/zhiwang/e_zhiwang.html.

[5] Z. Cai, H. T. Geng, X. Zhao, and Q. S. Cai, "An algorithm for conceptual clustering of Chinese text," in *Proc. the 3rd International Conference on Machine Learning and Cybernetics*, Shanghai, 2004, pp. 26-29.

[6] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143-175, 2001.

[7] J. A. Lozano, J. M. Pena, and P. Larranage, "An empirical comparison of four initialization methods for the kmeans algorithm," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027-1040, 1999.

[8] Q. Liu and S. J. Li, "Word similarity computing based on How-net," in *Computational Linguistics and Chinese Language Processing*, vol. 7, no. 2, Shanghai, 2002, pp. 59-76.

[9] H. R. Kim, and P. K. Chan, "Learning implicit user interest hierarchy for context in personalization," *Applied Intelligence*, 2008.

[10] D. Godoy and A. Amandi, "Modeling user interests by conceptual clustering," *Information Systems*, vol. 31, no. 4, pp. 247-265, 2006.

[11] X. C. Ni, X. Y. Wu, and Y. Yu, "Automatic identification of Chinese weblogger's interests based on text classification," in *Proc. the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Hong Kong, China: IEEE, 2006, pp. 247-253.

[12] A. Mccallum and K. Nigam, "A comparison of Event Models for Naïve Byaes Text Classification," In Proc. the *AAAI-98 Workshop on "Learning for Text Categorization"*, 1998, pp. 41-48.

[13] T. Joachims, "Text categorization with support vector machines: learning with many relevant features", In *Proc. of ECML-98*, 1998, pp. 137-142.

[14] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization", In Proc. *of ICML-97*, 1997, pp. 143-151.

[15] SOHU site, http://www.sohu.com

[16] C. C. Chen, M. C. Chen, and Y. Sun, “PVA: A Self-Adaptive Personal View Agent System,” *Journal of Intelligent Information Systems*, vol. 18, no. 2, pp. 173–194, 2002.

[17] Yam search engine, http://www.yam.com

[18] J. Trajkova and S. Gauch, “Improving ontology-based user profiles,” in *Proc. of the RIAO conference*, Vaucluse, France, 2004, pp. 380-389.

[19] Open Directory Project, http://www.dmoz.org

[20] F. Liu, C. Yu, and W. Meng, “Personalized web search by mapping user queries to categories,” in *Proc. the 11th International Conference on Information and Knowledge Management*, Mclean, Virginia, 2002, pp. 558-565.

[21] A. Sieg, B. Mobasher, and R. Burke, “Inferring users information context: Integrating user profiles and concept hierarchies,” in *2004 Meeting of the International Federation of Classification Societies*, Chicago, 2004.