International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0047
ISSN (Online): 2279-0055

International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Semantic Web user personalization and search techniques based on time and weather condition

[1]R. Kousalya, , [2]Dr.V.Saravanan
[1]P.hD Scholar, Manonmaniam Sundaranar University, Tirunelveli,Tamilnadu, India
Asst. professor, Dr. N.G.P. Arts and Science College,Coimbatore, Tamilnadu, India
[2]Director, Department of Computer Applications
Sri Venkateshwara College of Computer Applications and Management, Coimbatore, Tamil nadu, India

---

Abstract*: As a result of the rapid advancements in Information Technology, Information Retrieval on Internet is gaining importance, day by day. The web comprises of huge amount of data and search engines provide an efficient way to help navigate the web and get the relevant information. General search engines, however, return query results without considering user's intention behind the query. Personalized Web search is carried out for information retrieval for each user incorporating his/her interests. This paper presents a method which extracts the user's interests and preferences of content according to weather and time. Also it calculates the semantic relativity between the given words, and it will generate the semantic measures automatically. The result shows that the model built with user preferences by time-framed navigation sessions improve the effectiveness of personalization process.*

Keywords*: web personalization, ontology, time zones, semantic similarity*

I. Introduction

Web contains very large amount of information, which are scattered and dynamic as well as diverse in terms of content and nature. Since people with different background, knowledge, and expectation organize the information in web, users query is not adequate to represent the information they want to retrieve. Keyword matching technique fails to retrieve semantically or lexically related document thus retrieving more irrelevant results [1]. Such techniques are constrained by attempting to match the user keyword to the source document and present information to the user with documents that matched the user keyword. Our method uses the Information content approach to calculate similarity between two keywords in the taxonomy to discover the related concepts, which are not implicit in the query [2]. For example a search query seeking for the information on given term would return hits containing the specified term but would fail to retrieve the document that is described by its synonymy term.

In this paper, we presented an approach for capturing similarity between words that is concerned with the syntactic similarity of two strings. Semantic similarity is a confidence score that reflects the semantic relation between the meanings of two sentences. It is difficult to gain a high accuracy score because the exact semantic meanings are completely understood only in a particular context. Some dictionary-based algorithms are available to capture the semantic similarity between two words [3, 4].Context used in search query is of great importance in retrieving relevance information thus finding the meaning of the each word used in query is essential. For this similarity score of the concepts represented by each word in the query is computed. The pair of concept that has higher similarity value is chosen as the concept described by the words. This discovered concept is used to supplement users query with its synonyms based on relatedness score. And also, the main concept involved in the paper is extraction of data based on the time of search. I.e. if the user is querying in a time, all his/her past navigation patterns are extracted in order to give the exact result. For example, if the user had searched for 'Java', it may exhibit polysemy (same word in different senses such as Java programming language or Java islands). In this case, semantic similarity might be exhausted since both are semantically similar. Here we go for, time based extraction of past data, regarding which 'Java' he has searched previously. For this, user profile has been created and maintained, with all his past data and sessions. The highly relevant result will be displayed to the corresponding query. The discussions below are about calculating the semantic similarity between the keyword and content in chapter 2. Chapter 3 includes filtering the content based on time-zones and ranking the contents based on its content similarity. Chapter 4 implements the proposed model and provides the experimental results.

II. CALCULATING THE SIMILARITY MEASURE

The proposed semantic web personalization process exploits the expressive power of content semantics that are represented by ontology terms. Using such a representation, the similarity between documents is deduced to the distance between terms that are part of a hierarchy. The need for such a similarity measure is

encountered throughout the personalization process, namely during content characterization, keyword translation, document clustering and recommendations' generation. A popular similarity measure for ontology concepts is proposed by Resnik [5]. The author illustrates similarity between two ontology concepts is based on the "depth" of their least common ancestor, where the "depth" is measured using the information content.

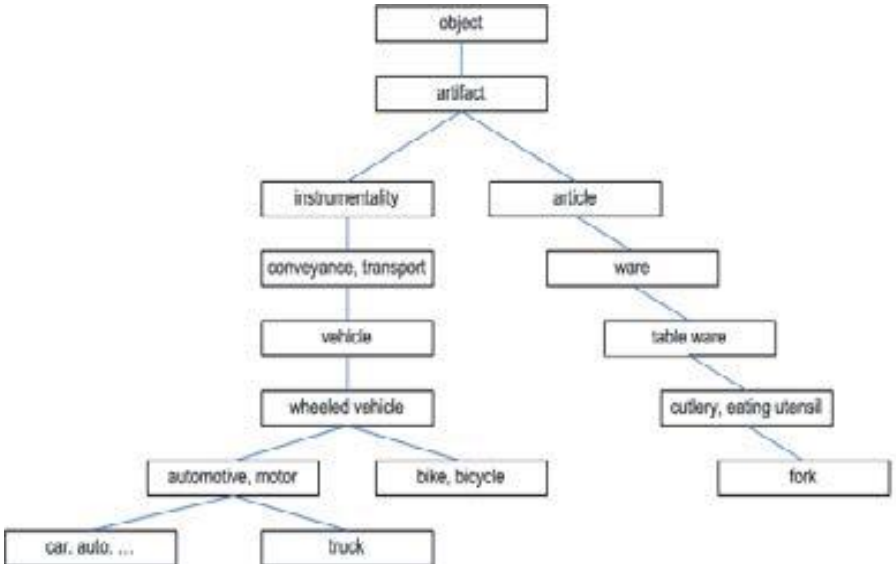

Fig 2.1: Tree to calculate the similarity measure

In the above figure, we observe that the length between car and auto is 1, car and truck is 3, car and bicycle is 4, car and fork is 12.A shared parent of two synsets is known as a sub-sumer. The least common subsumer (LCS) of two synsets is the subsumer that does not have any children that are also the subsumer of two synsets[6]. In other words, the LCS of two synsets is the most specific subsumer of the two synsets. Back to the above example, the LCS of {car, auto} and {truck} is {automotive, motor vehicle} since the {automotive, motor vehicle} is more specific than the common subsumer {wheeled vehicle}.

A. An ontology for Content

We assume that if a keyword/phrase exists frequently in the web-snippets arising from the query q, it represents an important concept related to the query, as it co-exists in close proximity with the query in the top documents. Thus, our content concept extraction method first extracts all the keywords and phrases from the web-snippets arising from q. After obtaining a set of keywords/phrases (C Ci), the following support formula, which is inspired by the well-known problem of finding frequent item sets in data mining [3], is employed to measure the interestingness of a particular keyword/phrase (Ci) with respect to the query q:

$$\text{Support } (C_i) = \frac{sfCi}{n} \mid C_i \mid \qquad \ldots\ldots\ldots 1$$

Where $sf(Ci)$ is the snippet frequency of the keyword/phrase Ci (i.e.) the number of web-snippets containing Ci , n is the number of web-snippets returned and $\mid Ci \mid$ is the number of terms in the keyword/phrase Ci . If the support of a keyword/phrase is higher than the threshold s (s = 0.03 in our experiments), we treat Ci as a concept for the query q. As mentioned, we use ontologies to maintain concepts and their relationships extracted from search results. We capture the following two types of relationships for content concepts:

Similarity: Two concepts which coexist a lot on the search results might represent the same topical interest. If coexist $(Ci, Cj) \succ \delta_1$ (δ_1 is a threshold), then, Ci and Cj are considered as similar. Parent-Child Relationship: More specific concepts often appear with general terms, while the reverse is not true. Thus, if $pr(Cj, Ci) > \delta_2$ (δ_2 is a threshold), we mark Ci as Cj 's child.

B. An Ontology for Location

Our approach for extracting location concepts is different from that for extracting content concepts. First, a document usually embodies only a few location concepts. As a result, very few of them co-occur with the query terms in web snippets. To alleviate this problem, we extract location concepts from the full documents. Second, due to the small number of location concepts embodied in documents, the similarity and parent-child relationship cannot be accurately derived statistically. Additionally, the geographical relationships among many

locations have already been captured as facts. Thus, we create a predefined location ontology consisting of about 17,000 city, province, region, and country names obtained from [4] and [5]. In the location ontology, we organize all the cities as children under their provinces, all the provinces as children under their regions, and all the regions as children under their countries. The location ontology extraction method first extracts all of the keywords and key-phrases from the documents returned for q. If a keyword or key-phrase in a retrieved document d matches a location name in our predefined location ontology, it will be treated as a location concept of d.

C. Mining Content and Location Notion

Different queries may be associated with different amount of content and location information. For example, queries such as Overseas Study may have strong associations to a large number of location concepts. However, queries such as Programming tend to be content-oriented with only weak association to location concepts (i.e., most concepts, such as books and software tools, related to computer programming are location independent). Meanwhile, some queries (e.g. Shopping) can be rich in both content and location information. To formally characterize the content and location properties of a query, we use *entropy* to estimate the amount of content and location information retrieved by a query.

In information theory [6], *entropy* indicates the uncertainty associated with the information content of a message from the receiver's point of view. In the context of search engine, entropy can be employed in a similar manner to denote the uncertainty associated with the information content of the search results from the user's point of view. Since we are concerned with content and location information only in this paper, we used two entropies, namely, content entropy and location entropy, to measure, respectively, the uncertainty associated with the content and location information of the search results. The information entropy of a discrete random variable X is defined as:

$$H(X) = -\sum_{i=1}^{n} p(xi)\log\, p(xi) \qquad \text{……..2}$$

Where n is the possible values{x1,x2,…xn) of *X* and *p(xi) =Pr(x=xi)* We adopt the above formula to compute the content and location entropies of a query q (i.e.H_C(q) and H_L(q)) as follows:

$$Hc(q) = -\sum_{i=1}^{k} p(ci)\log\, p(ci)$$

$$HL(q) = -\sum_{i=1}^{m} p(li)\log\, p(li) \qquad \text{……. 3}$$

(3) where k is the number of content concepts *C={ c_1,c_2,c_3... c_k)* extracted, $|C_i|$ is the number of search results containing the content concepts Ci, $|C| = |C_1| + |C_2| + |C_3| + \ldots |C_K|$,
Pci = $|ci|$ by $|C|$, m is the number of location concepts *L={ l_1,l_2,...l_m}* extracted. $|li|$, is the number of search results containing location concepts li, $|L| = l_1+l_2+\ldots l_k$ and pli = $|li| / |L|$

III. TIME BASED CONTENT FILTERING

Ontology for user preference According to Time zones

While doing personalization we consider time, it returns results based on times it perceives you are typically working. A personalization system that handle user's profile, content and location entropies. Application of the user profiles on that content and location entropies is the first step towards incorporating time in the personalization process. The proposed method involves calculating the similarity measure and incorporating the similarity in the time and weather based data. In detail, such a system should be able to:

1. Calculate the semantic similarity measure between the keyword and the web content.
2. Capture the user's preference or interest according to time zones and maintains the click-through ontology along with time zones.
3. Capture the user's device profile.
4. Combine the user's preferences for the particular time zone along with semantic similarity ontology in order to select the desired content for that time zone.
5. Train and update the user profile according to the user preferences

To achieve time based personalization we need to know how the user's preferences change over the 24 hour day cycle on different weather/climate condition. To represent time we suggest dividing the day into different time-zones. To represent the climate change we suggest the climate condition into summer and winter season. Because the browsing mood may vary depends on the changes in the climate. This is possible if we study the daily routine of our users and then split it into time zones based on the user's activities for each period.

Table 1 User's preferences for 24 hour per day cycle on summer season

Time Zone	User Preference
8-12	Work
12-14	Lunch
14-18	Work
18-21	Recreation
21-23	Dine out
23-8	Rest

Table 2 User's preferences for 24 hour per day cycle on winter season

Time Zone	User Preference
10-12	Work
12-14	Lunch
14-18	Work
18-20	Recreation
201-22	Dine out
22-10	Rest

By dividing the day in time-zones, we drastically reduce the possible combinations between time and user's preferences keeping our design scalable. From the user click through data, click through ontology for content and location will be created for all time zones. Time zones may vary based on climate of the day.

IV. Time based personalization algorithm

The Proposed algorithm for time based personalization will be called as **Time based Personalization Algorithm** (TBPA) as follows:

Step 1: Define clickthrough ontology for each timezones during training

Step 2: If new query is submitted, the middleware (exits between user and search engine) extracts time from the system

Step 3: Extracted time is matched with each timezone, and the matching timezone's clickthrough ontology will be considered to identify user preference on that instance

Step 4: Search results will be re-ranked for the user

Step 5: If the user prefers the concept other than the top ranked concepts then equal weight (i.e. the weight of the existing top ranked documents) is assigned for the new concept

Step 6 : Next time when the user submits the query repeat the steps 3

Step 7: If the concepts having equal weights Middleware ask user which concept is preferred at this moment

Step 8: Based on the user response again weights will be updated for the concepts

Learning User Preference

1) Ranking SVM: Here is the algorithm for Ranking SVM

- Input Space:X

  Ranking Function : $f : X \to R$

  Ranking function

  $$xi \succ xj \Leftrightarrow f(xi; w) \succ f(xj; w)$$

  Linear Ranking function:

  $$f(x; w) = \prec w; x \succ$$

Ranking SVM is employed in our personalization approach to learn the user's preferences. For a given query, a set of content concepts and a set of location concepts are extracted from the search result as the document features. Since each document can be represented by a feature vector, it can be treated as a point in the feature space. Using click through data as the input, RSVM aims at finding a linear ranking function, which holds for as many document preference pairs as possible. It outputs a content weight vector and a location weight vector **,** which best describes the user interests based on the user's content and location preferences extracted from the user clickthroughs, respectively. In the following, we discuss two issues in the RSVM training process: 1) how to extract the feature vectors for a document; 2) how to combine the content and location weight vectors into one integrated weight vector.

User profiles:

Considered content, location and time concept in which exploiting timing enables us to capture the changes in user interest based on time of the day and adopt preference accordingly. It effectively merges user's preferences under the appropriate time zone which create a dynamic user file. In this proposed work also concentrate on climatic /weather conditions of the day. The user browsing behavior may vary depends on the time and the climate which provides a dynamic user's profile. This dynamic user profile can accurately cover the preference of a user at all times and situations. While updating user profile according to content and climate /time concepts

which are associated with the query and user preferences on both concepts for that query over a time increases the effectiveness rate of web search according to user interest.

Table 3 Statistics to display the relevant links within top 20 links for user 1

	Existing			Proposed		
Query	No of results	Irrelevant result	Accuracy	No of results	Irrelevant result	Accuracy
1	76	6	92	74	3	96
2	56	10	82	55	3	95
3	45	6	87	42	4	90
4	50	7	86	52	3	94
5	60	10	83	60	2	97
6	65	9	84	64	4	94

V. Experimental results

The Experimental resuls for the personalized web search based on content,time and climate preferences based on TBPAlgorithm are provided in this chapter. The experiments carried out using the implemented prototype. 20 users a re in

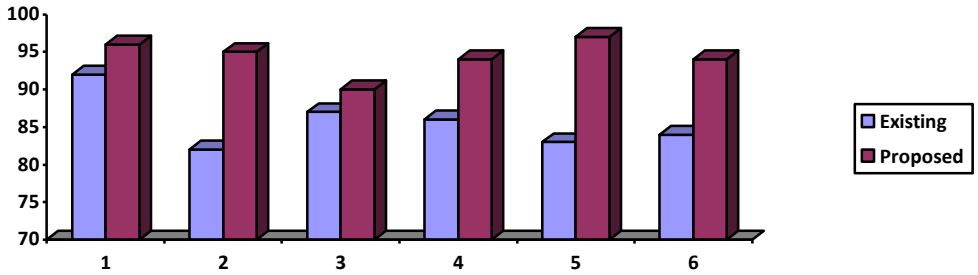

Fig 1: Performance rate of a proposed time based search for user 1

VI. CONCLUSION

The paper includes the methods finding the concepts from the web which are semantically similar to the keywords in the query. Furthermore, we have proposed an idea that, user preferences according to the time zones are extracted from the semantically similar data which will create click-through ontology based on time-zones. Based on the ontologies user profile will be created and updated using RSVM. The experimental results show that the proposed personalization approach provides higher performance rate when compared with separate time-based extraction or similar content extraction from the web.

REFERENCES

[1] Jay J. Jiang and David W. Conrath. 1997. "*Semantics and Similarity Based on Corpus Statistics and Lexical Taxonomy"*. In Proceedings of International Conference on Research in Computational Linguistics, Taiwan.

[2] R. Baeza-Yates, A. Tiberi. "*Extracting semantic relations from query logs",* Knowledge Data and Discovery, 2007

[3] Krishna Sapkota,Laxman Thapa, Shailesh Pandey, *Efficient Information Retrieval using measures of Semantic Similarity*,2006.

[4] George Tsatsaronis and Vicky Panagiotopoulou, *A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness*,2009.

[5] P. Resnik, *Using information content to evaluate semantic similarity in a taxonomy*, in Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI), 1995

[6] Chen, T., and Han, W. *Content recommendation system based on private dynamic user profile. Machine Learning*, Vol. 4, 2007, pp. 2112-2118

[7] Koren, Y. *Collaborative filtering with temporal dynamics*. Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining, Vol. 53, No. 4, 2009, pp. 447.

[8] T. Joachims "*Optimizing search engines using click through data*", in Proc. of ACM SIGKDD Conference, 2002.

[9] E. Agichtein, E. Brill, and S. Dumais "*Improving web search ranking by incorporating user behavior information*", in *Proc. of ACM SIGIR Conference*, 2006.

[10] P. Nithiya, V. Vidhya, Dr. L. Ganesan, Development of semantic based information retrieval using word-net approach. Second International Conference on Computer and Network Technology, 2010

[11] Magdalini Eirinaki, Dimitrios Mavroeidis, George Tsatsaronis,and Michalis Vazirgiannis, "Introducing Semantics in Web Personalization:The Role of Ontologies"Semantics, Web, and Mining 2005, LNAI 4289, pp. 147 – 162, 2006.© Springer-Verlag Berlin Heidelberg 2006.