

## User Intention based Personalized Search : HPS(Hierarchical Phrase Serch)

GunWoo Park, JinGi Chae, Dae Hee Lee, SangHoon Lee

Dept. Computer Science and Information, Korea National Defense University, South Korea

pgw4050@hotmail.com, hoony@kndu.ac.kr

Department of Tourism Management, DongA University, South Korea

Dehi73@korea.com

*Abstract.* Personalized search has recently got significant attention in the web search. Accordingly, user's intention of search is very important information to retrieval information in aspect of personalized Search. Although many personalized search and strategies have been proposed, the majority of web users are difficulty in retrieval information corresponding their search intention. In this paper, we present personalized search based on User Intention through the HPVM(Hierarchical Phrase Vector Model) to solve these problems using and machine learning methodology. Users can navigate through the prior user's intention by their own needs. This is especially useful for various meaning and poor queries. By analyzing the results, we find out that there is an unique representation of user intention under different queries, contexts and users. Furthermore, we can find out that this knowledge is very important in improving personalized information retrieval performance by filtering the results, recommending a new query, and distinguishing user's characteristics. With this approach, search engines can provide more predictive information for Web searchers. Based on this approach, we developed a personalized search engine, HPS (Hierarchical Phrase Search).

*Key-Words:* - Personalization, Information retrieval, Hierarchical phrase search, Support vector machine, Machine learning, Relevance feedback

### 1 Introduction

It is necessary that the majority of people in the world use World Wide Web, and search engines are developing as the main portal of the Web. Search engines are becoming as "tools" for retrieving the information, internet sites, and many other resources that people want on the web [1], [2]. Beyond their popularity, how are people using these web search engines? How can we determine what these people are seeking for? Search engines encounter problems such as query ambiguity and results ordered by popularity rather than relevance to the user's individual needs. In fact, most of queries to search engines are short and ambiguous and users may have completely different information needs and goals under the same query.

To solve these problems, there have been many strategies to improve search accuracy based on personalized information [3], [4], [5] [6], [7], [8]. Relevance Feedback (G. Salton and C. Buckley, 1990) is the main method for automatically improving a system's accuracy of individual needs. Relevance feedback has been proved to be effective for improving retrieval accuracy (G. Salton and C. Buckley, 1990; J. J. Rocchio, 1971). Therefore, we

research the method to find out the searcher's implication intention more easy and efficiently.

It is possible to characterize many features of variation with a small number of dimensions that could be useful to represent a searcher's intention. What's needed is a unique criteria to be considered. Namely, different people find different relevant things depending on the characteristic of user and the context of retrieved results. For a given query, a personalized search can provide different results for different users or organize the same results differently for each user.

It can be implemented on either the server side (search engine) or the client side (user's computer). But server side implementation is not proper when performing expensive data mining algorithm and pattern recognition functions in real-time environment. A personalized search on the client side can be achieved by suggesting top-n representation of user's intentions or researching new queries associated with query and result processing [Pitkow et al. 2002]. The result processing includes result filtering (such as removal of some results) and reorganizing (such as re-ranking, clustering, and categorizing the results).

Research shows that relative preferences derived from immediately viewed documents are reasonably accurate on average (T. Joachims et al., 2005) and straight-forward click-based personalization strategies perform well.

In this paper, we utilize the immediately clicked documents (6-10) by explicit relevance feedback. Those documents are positive or negative to searcher's intention [9]. Then our client side application find out every possible patterns hierarchically mined from advanced Set-based Vector Model, we named this as HPVM (hierarchical Phrase Vector Model) [10], [11], [12], [13]. Then the application learns patterns of hierarchical phrases from user's positive and negative documents by means of SVM (Support Vector Machine) and returns most relevant phrase [14]. We use this information as personalized search criteria to filter the results, recommend a new query, and distinguish user's characteristics [15].

The main contributions of this paper are as follows.

- We propose the first Semantic Criteria for personalized search, which is fully adaptive to represent ambiguous user needs. This is an expansion of Set-based Vector Space Model, 'HPVM (Hierarchical Phrase Vector Space Model)' to compensate the weakness of original vector space model. With these criteria, we can build each user's profile based on user's interaction with search engine from a tiny set of choices (about 3-5).
- We have implemented and engineered a public prototype that includes all the features above on the client side. "Fig 1, 2" shows our prototype system. In this framework, searcher's intentions from different query and content are gathered and system's efficiency is evaluated in real-time environment.
- We also find out that this methodology has high accuracy and effectiveness on different queries, users, and search contexts.

The remaining sections are organized as follows. In Section 2, we discuss related works. We present a advanced Set-based Vector Model, HPVM and introduce how to score each Vector with SVM to extract the searcher's intention in Section 3. In Section 4, we give a relevance-evaluation framework for personalized search. In Section 5 we discuss the query-set used in our experiments and detailed data statistics. Experimental results and

analysis of this framework are presented in Section 6. Future works are discussed in Section 7.

## 2 Related Work

Some of the existing researches capture users' information need by exploiting query logs. For example, M. Speretta and S. Gauch (2005) build user profiles based on activity at the search site and study the use of these profiles to provide personalized search results. Some researches improve retrieval performance by exploiting users' browsing history (F. Tanudjaja and L. Mu, 2002; M. Morita and Y. Shinoda, 1994) or Web communities (A. Kritikopoulos and M. Sideri, 2003; K. Sugiyama et al., 2004) Some researches utilize client side interactions, for example, K. Bharat (2000) automatically discovers related material on behalf of the user by serving as an intermediary between the user and information retrieval systems. Some latest researches combine several types of implicit feedback information. J. Teevan et al. (2005) explore rich models of user interests, which are built from both search-related information, such as previously issued queries and previously visited Web pages, and other information about the user such as documents and emails, the user has read and created.

Our approach is to ask users to specify general intention. The user intention is distinguished from user's interest in terms of time. Intention of a searcher exists very short time, about 1-3 minutes from issued time. The paradigm allows us to evaluate the contribution of different sources of information to the quality of personalization in different contexts and users.

We take a relevance-feedback perspective on modeling personalization. Relevance feedback has a solid theoretical foundation and a long history of application to information retrieval. Our approach differs from standard relevance feedback in that it requires small number of explicit judgments. So a searcher's effort can be minimized, and usefulness of system can be maximized. This method is distinguished from blind or pseudo-relevance feedback as they operate over a longer time frame than an individual query. To summarize, in our approach to Web search personalization, we use a wide range of implicit user activities over a long period of time to develop an implicit user profile. This profile is used to re-rank Web search results employing a relevance feedback framework. In our current approach, all profile storage and processing is done on the client machine.

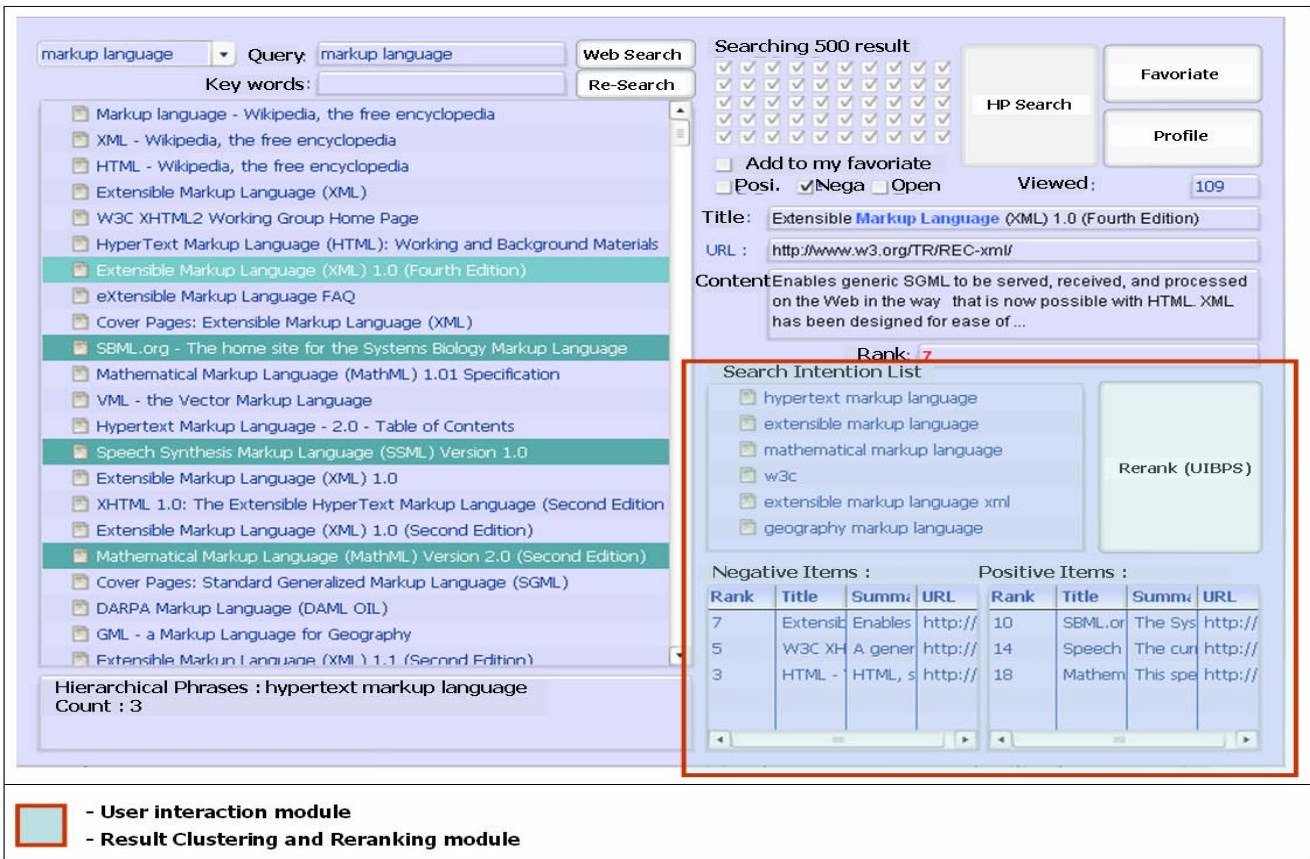


Fig 1. User Intention-based personalized Search System



① divided into a word unit → ② Execution of Hierarchical Phrase Generation → ③ Execution of Weight extracting for indexed word

Fig 2. Process of extracting user intention

## 2.1 Set-based Vector Space Model

The set-based model is the first information retrieval model that exploits term correlations effectively, provides significant gains in precision, and has processing costs close to the costs of the vector space model, independently of the collection and query type considered. The model exploits the intuition that semantically related terms often occur close to each other by implementing a pruning strategy that restricts computation to proximate term sets.

Set-based Vector Space Model automatically discovers phrases based on co-occurrence probabilities. But the gains are so small that they can't be worth the cost of computation in server-side application. But there are some problems in Set-Based Vector model. Each term's order information with previous and next words is ignored in Set-based Vector Model. Therefore we made advanced Set-based Vector Model which contains ordering information as shown in "Table 1" We named this as HPVM. With HPVM, the machine can understand human's concept structure and natural language more easily and efficiently.

**Table 1.** Comparison of Models

Model	Term-set	Level	Feature
VSM	$S_a, S_b, S_c, S_d$	1	4
SBVM	$S_a, S_b, S_c, S_d, S_{ab}, S_{bc}, \dots, S_{abcd}$	4	15
HPVM	$S_a, S_b, S_c, S_d, S_{ab}, S_{ba}, S_{ac}, S_{ca}, \dots, S_{abcd}, S_{bacd}, S_{acbd}, \dots$	4	64

When  $n$ -queries are issued, the feature amount is  $n$  in Vector Space Model,  $2^{n-1}$  in the Set-based Vector Space Model, and  $2^{2n}$  in HPVM. To compute term-sets, we propose an effective algorithm based on association rules theory [Agrawal et al. 1993]. Association rules are interesting because they provide all the elements of the TF  $\times$  IDF scheme in an algorithmically efficient and parameterized way.

## 2.2 SVM

The theory and algorithm about SVM are originally established by Vapnik (1998) and have been applied to solve many practical problems since 1990s. SVM benefits from two good ideas: maximising the margin and the kernel trick. These good ideas can

guarantee high testing accuracy of classifiers and overcome the problem about curse of dimensionality.

Designing effective schemes for term weighting is a critical step in a search system if improved ranking is to be obtained. However, finding good term weights is an ongoing challenge. In this work we propose a new term weighting schema that leads to improved ranking and is more practical through the information of user's choices. However, it is generally accepted that exploitation of the correlation among index terms in a document might be used to improve retrieval effectiveness with general collections. In fact, distinct approaches that take term co-occurrences into account have been proposed over time [Wong et al. 1985, 1987; van Rijsbergen 1977; Harper and van Rijsbergen 1978; Raghavan and Yu 1979]. All these approaches suffer from a common drawback; they are too inefficient computationally to be of value in practice.

In this paper, we propose a new scheme for computing index term weights that takes into account patterns of term co-occurrence and is efficient enough to be of practical value with SVM. Moreover, they naturally provide for the quantificational information of user's ambiguous selection from patterns of term co-occurrence.

Many learning models make use of the idea that any learning problem can be made easy with the right set of features. The trick, of course, is discovering that "right set of features," which in general is a very difficult thing to do. SVM are another attempt at a model that does this. The idea behind SVM is to make use of a (nonlinear) mapping function.

- Transforms data in input space to data in feature space in such a way as to render a problem linearly separable. The SVM then automatically discovers the optimal separating hyper plane
- SVM are rather interesting in that they enjoy both a sound theoretical basis as well as state-of-the-art success in real-time applications.

Nonlinear Support Vector Machine (SVM) has been widely used in many applications, like text categorization determining the right parameters of such functions is not only computationally expensive, the resulting models are also susceptible to over fitting when the number of training examples is small. The quality of approximation by  $F$  can be measured by various objective functions.

### 3 Extract User Intention

#### 3.1 Hierarchical Phrase Vector Model

In this section, we introduce the concept of term-sets as a basis for modeling dependences among index terms in the set-based model. HPS uses an innovative bottom-up hierarchical mining algorithm whose aim is to construct a hierarchical phrase. The HP Procedure is a discovery of useful linguistic patterns.

*Definition 3.1.* Let  $T = \{k_1, k_2, \dots, k_t\}$  be the set of  $t$  unique terms that appear in document  $C$ . There is a total ordering among the vocabulary terms, which is based on the lexicographical order of terms, so that  $k_i < k_{i+1}$ , for  $1 \leq i \leq t-1$ .

*Definition 3.2.* Let Minimum Frequency  $MF = \{mf_1, mf_2, \dots, mf_n\}$  be the minimum frequency of each term-set, where  $n$  is level. This value is used for pruning term-set. If term-set  $p$ 's level  $j$  and the frequency is  $p.f$ , then  $p.f > mf_j$ .

*Definition 3.3.* Let  $P-n$  be level- $n$  term-set,  $P-1 = T$ ,  $P-2 = \{[k_1, k_2], [k_2, k_1], \dots, [k_m, k_n]\}$  is a level-2 set of terms, where  $n$  is level.

*Definition 3.4.* Relation Concept-Reachable be  $p_k \sim p_l$ , where  $p_k$  and  $p_l$  are term-sets from  $P$ , and  $m, n$  is level where  $m \neq n, |m-n| \geq 1$  and of each term-set. This relation is symmetric.

$$(p_{k-m} \approx p_{l-n}) = (p_{l-n} \approx p_{k-m})$$

*Definition 3.5.* Relation Concept-Connected be  $p_k \gg p_l$ , where  $p_k$  and  $p_l$  are term-sets from  $P$ , and  $m, n$  is level where  $m < n, n-m=1$  and of each term-set. This relation is asymmetric.

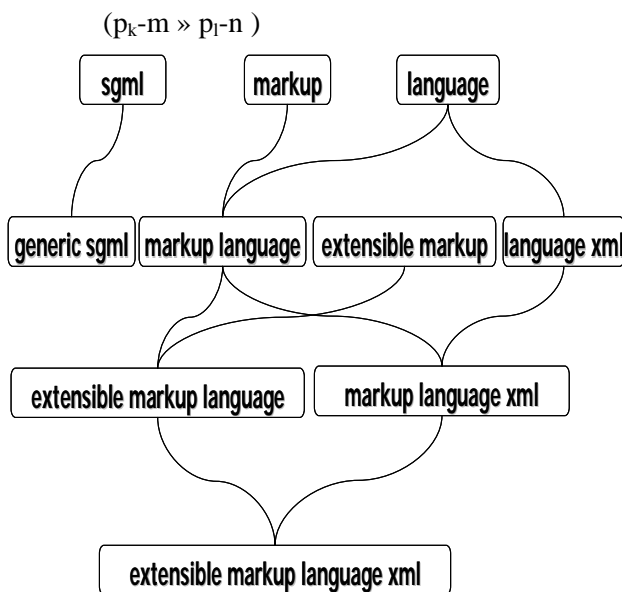


Fig 3. Concept Graph of 'xml'

For example, when query 'xml' is issued, "Fig 3" shows a hierarchical construction process. This inductive step of the bottom-up hierarchical construction process consists of three main phrases: preprocessing, term-counting and pruning by MF. After the pruning, the remaining phrases provide the next level upon which the bottom-up process is repeated again. The process is stopped after that  $n$  levels have been built (a deeper hierarchy would not be necessary).

```

HPVM(D, P, MF)
1. for each i, 0 < i < m, //term-set counting
2. if (D[i] » P-n) P-n.addItem(D[i])
3. else P-n.Item(D[i]).count += 1
4. end
5. for each j, 0 < j < P-n, //term-set pruning with MF-n
6. if (p.j.count < MF-n) p.j.pop()
7. end
8. for each i, 0 < i < m, //n+1 step term-set generate
9. for each j, 0 < j < P-n,
10. if ((D[i-1]+D[i]) » P-(n+1)) P-(n+1).addItem((D[i-1]+D[i]))
    else P-(n+1).currentItem.count += 1
11. end
12. end
13. for each l, 1 < level < n, //Merge each level term-set
14. if (P.addItem(P-level))
15. end
    
```

Fig 4. Hierarchical Phrase Generating Algorithm

Let  $G = (V;E)$  denote the concept graph, where  $V$  is the term-set in  $P$  and  $E$  is a directed edge of rule concept-connected; For a vector  $v$ ,  $v(p)$  denotes term-set  $p$ , the  $p$ -th component of  $v$ . The magnitude of a vector  $v$  is defined to be  $W_i$ , where  $i=1, 2, \dots, p$ . In this paper, vector magnitudes are always in  $[0; 1]$ . In an implementation, a vector may be represented as a list with its nonzero weight and level is 4 and  $MF = \{2,1,2,2\}$ . Details of the Hierarchical Phrase Generating algorithm are presented in "Fig 4".

Each term-set is represented as hierarchical phrase. And Hierarchical Phrase Generating Algorithm is as follow; let  $P-n$  be a set of phrase of  $q$  level  $n$ ,  $P-1$  be a set of 1-termsets  $T$ , the level is 1,2,3,4 and  $D$  is an array document array with the length  $m$  spitted by blank space.

### 3.2 SVM Weight

To successfully identify intention of searcher in retrieved results, it was necessary to associate queries with the intention of the user, rather than relying on the exact query string being repeated.

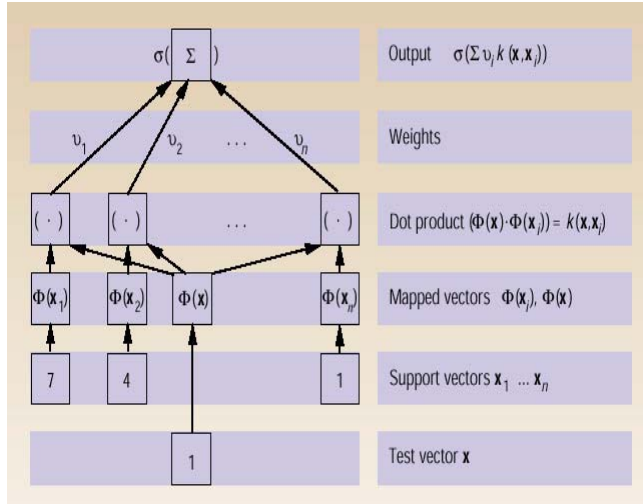


Figure 5. SVM Architecture

```

SVM_W_3Obj(D, PV, ER, PL, NL)
1. Let PV phrase vector with count and level
2. Let L vector be existence Boolean value at each Dk
3. Let NEV be the negative selection feature space vector
4. Let POV be the positive selection feature space vector
5. for each i, 0 < i < |Fk|,
6.   if (Fi = 1) POV.addItem(Lk)
7.   if (Fi = -1) NEV.addItem(Lk)
8. end
9. Let Combination Vector COMV = {NEVi, POVj, POVj-1}
10. Let FSCV be the Feature Selection Criteria Vector
11. for each k, 0 < k < |COMV|, SVM_temp = array
12.   SVM_temp = SVM(NEVk, POVk, POV2k)
13.   for each m, 0 < m < |PV|, ER_temp = 0
14.     if (!SVMD(PVm)) ER_temp += 1
15.   end
16.   if (ER_temp < ER) FSCV.addItem(SVM_temp)
17. end
18. Sort FSCV descending order
19. Return FSCV(PL)
20. Sort FSCV ascending order
21. Return FSCV(NL)
    
```

Figure 6. SVM –Weight Generation Algorithm

VSM is vulnerable with spamming because of using term-frequency. Instead of counting each term’s frequency like VSM we are interested in extracting each term-sets’ weights from pattern of user’s choice. The SVM architecture and SVM-Weight Generation Algorithm using 3 object is shown in “Fig 5, 6”.

We modeled user’s choices as two type; positive and negative as +1 and -1. This view point of modeling enables us to determine each term-set’s weight mathematically from hyper plane. Furthermore accuracy of searcher’s choices can be gathered from hyper plane’s error rate.

SVM training task consists of the following Quadratic Programming (QP) dual formulation:

$$\text{maximize : } \sum_{i=1}^o \alpha_i - \frac{1}{2} \sum_{i,j=1}^o \alpha_i \alpha_j y_i y_j \Phi(x_i, x_j) \quad (1)$$

$$\text{subject to : } \sum_{i=1}^o \alpha_i y_i = 0 \quad (2)$$

where scalar b(bias) and vector of alphas  $\alpha$  (of length  $o$ ) are the variables determined by the above QP optimization problem. Where  $o$  is the number of training examples,  $t$  is the number of term-set,  $y_i$  is the label (+1 for positive example, -1 for negative) for the  $i_{th}$  training example ( $x_i$ ), and  $\Phi(x_i, x_j)$  denotes the value of the SVM mapping function for  $i_{th}$  and  $j_{th}$  examples.

The output distribution of the SVM, for any example  $x$  is computed by:

$$W_{p,i} = \left( \sum_{j=1}^o (\alpha_{i,j}) \right) / \left( \sum_{j=1}^o \sum_{i=1}^t \alpha_{i,j} \right) \quad (3)$$

And Error rate is computed by :

$$\begin{aligned} ERP &= ERP + 1, \text{ where } (ax + b_i) > 0 \rightarrow \text{positive} \\ ERN &= ERN + 1, \text{ where } (ax + b_i) < 0 \rightarrow \text{negative} \\ ER &= (ERP + ERN) / o, o : \text{Number\_of\_SVM\_Object} \end{aligned} \quad (4)$$

To evaluate performance of SVM, we made three scheme as follow;

- 2 object SVM
- 3 object SVM (positive enhanced)
- 3 object SVM(negative enhanced)

### 3.3 Identifying Intention Automatically

In order to identify intention of the searchers, we must leverage an increased knowledge of user behavior, especially efforts to extract concept structure from choices of searcher and weight each structure. The following are our priority of identifying user’s web search Intention: (1) SVM Weight, (2) Level of structure, (3) Length of structure.

## 4 System Design and Case Study

### 4.1 System Design

In this section, we present our experimental system HPS, which is based on the popular Web search engine Google. HPS has four main components: Result retrieval component, Feature weighting component, User hp store Component, User interaction component, Result clustering / re-ranking component. The architecture is shown in “Fig 7”.

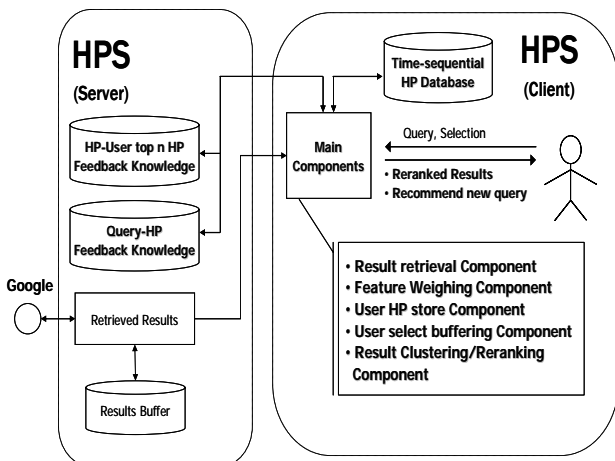


Fig 7. System Framework of HPS

The result retrieval component runs in backgrounds and retrieves results from search engine. When the query has been issued, result retrieval component uses the keywords to continue retrieving 500 results within 2 seconds. The user interactions component can handle three types of basic user actions: (1) submitting a query; (2) clicking to view a search result as positive and negative, (3) clicking the generated ‘User intention’. The Feature weighting component responds with: (a) exploiting and extracting representative term-sets by HPVM in 3 seconds; (b) weighting term-sets by SVM from explicit feedback information in a second; (c) sending the identified user intention. The Result clustering / re-ranking component performs

clustering by generated user intention and re-rank retrieved result. The User HP store Component store user Intention to local repository, when re-ranked results are presented to user. By re-ranking the result, the unseen search results and expanding the original query. Each component is implemented to meet the time requirement.

### 4.2 Sample Intention from a query

Users issued the same query ‘markup language’, but they visit a potentially disjoint set of retrieved results. We recruited 25 participants, all of which were KNDU postgraduate students, mostly from majors in the computer sciences. Their mean age was 28.4 years. In this section, we observed what a user’s intention can be extracted.

System parameters are as follows:

- Query : ‘markup language’
- Retrieved results : 434 items
- Tested user : 25
- MF = {2,1,3,3}
- Maximum SVM Error Rate : 20%

The participants were offered that they should have one intention as topic for positive selection. They randomly select various type of intentions like “Fig 8”. Much of them select relevant as ‘extensible markup language’, and ‘hypertext markup language’, and ‘wikipedia’. At the end of intention generating procedure, participants we asked correctness of phrase that contains searcher’s intention. The accuracy was about 96% among 216 selections.

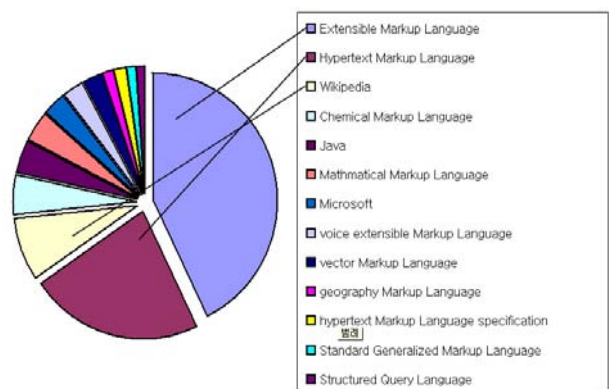


Fig 8. User Intentions of query ‘markup language’

In “Table 2” we present the results of each user’s choices that include positive and negative when we assume that user’s Intention is ‘hypertext markup language’. These results indicate there is unique representation of semantic structure from different users under different contexts. Our application is effective under 10 choices because the execution

time is under 2 seconds. This responsibility is reasonable for many users.

**Table 2.** Intention of 'hypertext markup language'

Negative Choices	Positive Choices	Text Length	SVM Time(ms)	HPVM Time(ms)	ER(%)
4	4	1596	397	982	9.71%
5	5	1889	811	1201	5%
4	5	1684	846	1020	14.85%
6	3	1721	359	1195	21.5%
10	7	3288	11215	6845	18.26%
5	5	1851	1338	2049	10.29%

**Table 3.** Discovery of useful linguistic patterns

Query	User Intentions	Coverage
markup language	markup language	269
	<i>xml</i>	168
	extensible markup language	149
	hypertext markup language	68
	wikipedia	48
	vector markup language	25
	geography markup language	21
extensible markup language	w3c	13
	extensible markup language xml	254
	<i>xml</i>	89
	second edition	37
	third edition	28
	fourth edition	12
	FRC	37
voicexml	29	
xml	amazon	11
	<i>xml</i>	384
	extensible markup language xml	178
	xml tutorial	18
	xml example	16
	xml scheme	24
	xpath	11
	xquery	8
	xml signature	7
javascript	6	
hypertext markup language	hypertext markup language	321
	<i>xml</i>	73
	w3c	43
	html 2.0	37
	html 3.0	6
	html 4.0	34
	html Specification	28
wikipedia	6	
sgml	standard generalized markup language	88
	extensible markup language	137
	<i>xml</i>	63
	hypertext markup language	265
language	html	25
	nglish	126
	english language	41
	language tool tip	29
	java language	52

### 4.3 Same Intention from many queries

For this research questions, we qualitatively analyzed samples of queries from Web search engine title and snippets in order to identify characteristics of queries. For the analysis, we selected random samples which can involve same search intention and manually selected 10 queries. We then derived characteristics for each concept structure that would serve to define the searcher's intention.

- Same Participants
- Query : { 'markup language', 'extensible markup language', 'xml', 'html', 'sgml', 'language', 'xsl', 'xpath', 'xquery', 'w3c' }
- Retrieved results : 450-500 items for each query.

There are many searchers' Intentions generated by HPVM and SVM. "Table 3" shows collaboratively extracted User Intentions and the coverage of each. Furthermore there is same intention 'xml' from these 10 queries.

## 5 Experimental Setup

In this section, we will demonstrate that we can perform our Hierarchical Phrase Search System by applying machine learning to a set of features. To examine the usefulness of our system, we created an evaluation collection by having 25 participants evaluate the top 500 Web search results for approximately 10 self-selected queries each. Web search results were collected from Google Search. For each search result, the participant was asked to determine whether they found the result relevant, or not relevant to the query. So as not to bias the participants, the results were presented in a random order. We first discuss the query selection and the scheme of training process.

### 5.1 Query Selection

By allowing the participants to decide whether or not they wanted to evaluate a particular query, we sought to provide them with a query and associated results that would have some meaning for them.

**Simple 50 Words** like {capital, data, java, jaguar, organ, search, plant, seal, suit, web, ...} have multiple meaning according to context. These words are ambiguous than abbreviation words.

**Abbreviation 50 Words** like { dns, ftp, lan, irc, www, smtp, udp, crm, un, usa, nato, asem, xml, rfid, vrml, ...} have an unique meaning regardless of context.



**User Selected 100 Queries.** Each user selected 4 queries with their own intentions.

### 5.2 SVM Scheme

In particular we observe users on their way through the web site and assign positive and negative scores to their actions. With this data, we would like to discover a simple SVM that accurately discriminates the two classes. Since the data is linearly separable, we can use a linear SVM (that is, one whose mapping function ( $\Phi$ ) is the identity function). To select best Linear SVM scheme, we tested 3 different ways. And each mapping function as follows;

- 2 Object SVM

$$\begin{aligned} \alpha_1\Phi(s_1) \cdot \Phi(s_1) + \alpha_2\Phi(s_2) \cdot \Phi(s_1) &= -1 \\ \alpha_1\Phi(s_1) \cdot \Phi(s_2) + \alpha_2\Phi(s_2) \cdot \Phi(s_2) &= +1 \end{aligned} \tag{5}$$

- 3 Object SVM (positive enhanced)

$$\begin{aligned} \alpha_1\Phi(s_1) \cdot \Phi(s_1) + \alpha_2\Phi(s_2) \cdot \Phi(s_1) + \alpha_3\Phi(s_3) \cdot \Phi(s_1) &= -1 \\ \alpha_1\Phi(s_1) \cdot \Phi(s_2) + \alpha_2\Phi(s_2) \cdot \Phi(s_2) + \alpha_3\Phi(s_3) \cdot \Phi(s_2) &= +1 \\ \alpha_1\Phi(s_1) \cdot \Phi(s_3) + \alpha_2\Phi(s_2) \cdot \Phi(s_3) + \alpha_3\Phi(s_3) \cdot \Phi(s_3) &= +1 \end{aligned} \tag{6}$$

- 3 Object SVM (negative enhanced)

$$\begin{aligned} \alpha_1\Phi(s_1) \cdot \Phi(s_1) + \alpha_2\Phi(s_2) \cdot \Phi(s_1) + \alpha_3\Phi(s_3) \cdot \Phi(s_1) &= -1 \\ \alpha_1\Phi(s_1) \cdot \Phi(s_2) + \alpha_2\Phi(s_2) \cdot \Phi(s_2) + \alpha_3\Phi(s_3) \cdot \Phi(s_2) &= -1 \\ \alpha_1\Phi(s_1) \cdot \Phi(s_3) + \alpha_2\Phi(s_2) \cdot \Phi(s_3) + \alpha_3\Phi(s_3) \cdot \Phi(s_3) &= +1 \end{aligned} \tag{7}$$

## 6 Evaluation Result

We performed experiments using search engine ‘Google’ in real environment. Each web retrieved containing 500 result pages. Automatic identification of user intent from these web pages also has important implications in building intelligent conversational QA systems. For example, if **Confirming** is identified during interaction, then the system can automatically collect the positive-negative pairs for potential future use. If **Canceling** is identified, the system may put aside the selection that has not been correctly answered and proactively come back to previous step later after more information is gathered. We will investigate these strategies while interact with users.

System setting is as follows;

- System collects top-20 Hierarchical Phrase at each query, which contains intention of searcher.
- For each User-Intention, every queried word is mapped.

- User Profile is made from 100 previous intentions in client side repository.
- Queries are 50 Ambiguous words, 50 Abbreviation words, and user selected 100 Queries.
- Testing Period is an week, 2007-07-09.

### 6.1 Top 10 F-score measurements

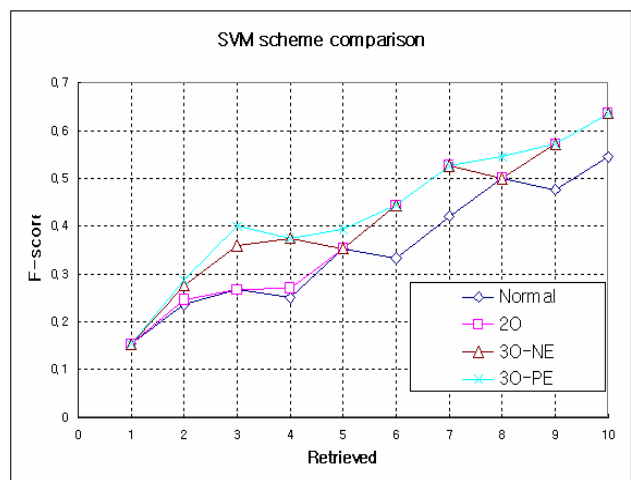
The F-score is a standard evaluation metric which balances between precision and recall measurements; Since both recall and precision are important for evaluating HPS systems, we combine them and compute the standard F-score;

$$F = 2(1/P + 1/R) \tag{8}$$

**Table 4.** Table Score Measurement

Retrieved	Normal	20	30-NE	30-PE
1	0.153	0.153	0.153	0.153
2	0.236	0.246	0.276	0.286
3	0.267	0.267	0.36	0.4
4	0.25	0.27	0.375	0.375
5	0.353	0.353	0.353	0.393
6	0.333	0.444	0.444	0.444
7	0.421	0.527	0.527	0.527
8	0.5	0.5	0.5	0.545
9	0.477	0.572	0.572	0.572
10	0.545	0.636	0.636	0.636

In **Table 4** we present top 10 F-score measurements. SVM-30 Positive enhanced scheme was out performed in “**Fig 9, 10**”, about 21% f-score improvement than the performance of google.



**Fig 9.** SVM scheme comparison

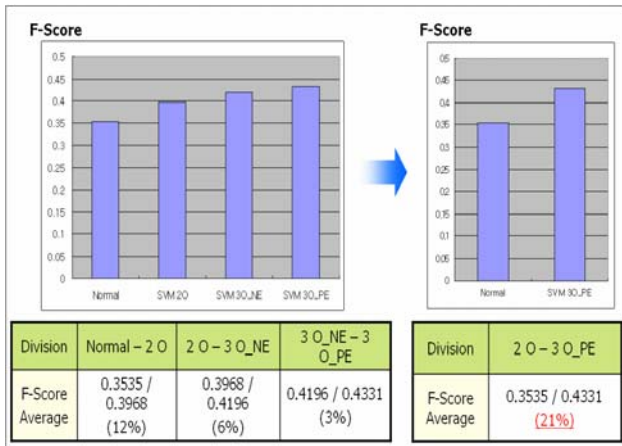


Fig 10. Improvement of F-Score according to each SVM

### 6.2 Conceptual Independency

We manually classified 200 queries as 4 types below to know how much effects a query have;

- Query Level 1 like {‘search’}
- Query Level 2 like {‘web search’}
- Abbreviation words with more than 3 words like {‘nato’}

Query Level 3 like {‘web search engine’}

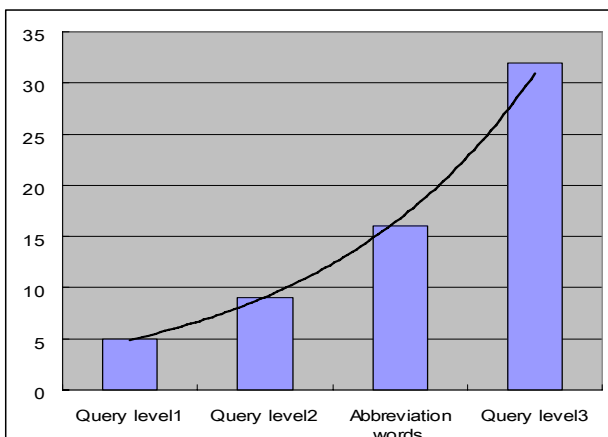


Fig 11. Conceptual Independency Increase

“Fig 11” show that the number of hierarchical phrase increases as the query level are increasing in proportion to  $n^2$ . Our model using hierarchical phrase recognize 4 level has very remarkable performance than other existing models.

### 6.3 Execution Time

We evaluate the efficiency of our system with a modest machine (Intel Pentium Core2 duo 2.16 GHz, 4GB memory). The system is implemented in the php & actionscript language. We cached the

retrieved results of top 500 URLs and snippets for reducing the time cost of database accesses. The average time of processing 10 choices of a user is 1.3 sec.

## 7 Conclusion and Futurework

Our main contributions can be concluded as follows:

- The proposal to study the problem of identifying ambiguous intentions of searcher.
- The proposal of the effective algorithm – Hierarchical Phrase Generating, SVM with 3Object positive enhanced.
- The proposal of Building User-Profile using HP.
- Client-side implementation of HPS and operational qualification of 6 components.

Personalized search opens the door to a new set of challenges and opportunities. One difficult problem is modeling a user’s changing interests and intentions over time. In order for Web search engines to continue to improve, they must leverage an increased knowledge of user behavior, especially efforts to understand the underlying intention of the searchers. This makes it a viable solution for Web search engines to classify user intent based on user intention-based personalizing strategy. Additionally, the larger data set provides more accurate percentages of user intent classification than smaller mostly manual studies.

In the future, we will conduct more user studies for evaluating the effectiveness of our algorithm since browsing problem need more consideration in the view of user. Further more, we are going to automate the process that system divides query set into positive and negative set. And then we are going to reduce the unnecessary effort of web user through this automation system.

### References:

- [1] Kaszkeil, M., Zobel, J., “Passage retrieval revisited,” SIGIR Conference p. 178-185, 1997.
- [2] Paul Ogilvie, Jamie Callan, “Structured documents: Combining document representations for known-item search,” SIGIR conference, 2003.
- [3] Liu, F., Yu, C. and Meng, W., “Personalized Web search by mapping user queries to categories,” CIKM p. 558-565, 2002.
- [4] Pitkow, J., Schutze, H., Cass, T., Cooly, R., Turnbull, D., Edmons, A., Adar, E., And Breuel,

- T., "Personalized search. Commun," ACM 45, 9, 50-55, 2002.
- [5] Paolo Ferragina, Antonio Gulli, Dipartimento di Informatica, Pisa, "A Personalized Search Engine Based on WebSnippet Hierarchical Clustering," WWW Conference 2005.
- [6] Teevan, J., Dumais, S. T. and Horvitz, E., "Beyond the commons: Investigating the value of personalizing Web search," Personalized Information Access(PIA), 2005.
- [7] Teevan, J., Dumais, S. T., And Horvitz, E., "Personalizing search via automated analysis of interests and activities," SIGIR Conference, p. 449-456, 2005.
- [8] Zhongming ma, Gautam pant, and Olivia r. liu sheng, "Interest-based personalized search," ACM Transactions on Information Systems, 2007.
- [9] Thorsten Joachims, Laura Granka, "Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search," ACM Transactions on Information Systems, Vol. 25, No. 2, Article 7, 2007.
- [10] Possas, B., Ziviani, N., And Meira, JR.,W., "Enhancing the set-based model using proximity information," String Processing and Information Retrieval, p. 104-116, 2002.
- [11] Possas, B., Ziviani, N., Meira, JR., W., And Ribeiro-Neto, B., "Set-based model: A new approach for information retrieval," ACM-SIGIR Conference, p. 230-237, 2002.
- [12] Possas, B., Ziviani, N., Ribeiro-Neto, B., And Meira, JR., W., "Processing conjunctive and phrase queries with the set-based model," String Processing and Information Retrieval, 171-183, 2004.
- [13] G. Salton, A. Wong, C. S. Yang, "A vector space model for automatic indexing," ACM, Vol. 18 Issue 11, 1975.
- [14] A. Wong, C. S. Yang, "A vector space model for automatic indexing," Communications of the ACM, Vol. 18 Issue 11, 1975.
- [15] Yuanhua Lv ,Le Sun, et. al., "An Iterative Implicit Feedback Approach to Personalized Search," Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 585–592, 2007.