Exploiting Taxonomic Knowledge for Personalized Search: A Bayesian Belief Network-based Approach

JAE-WON LEE, HAN-JOON KIM and SANG-GOO LEE

School of Computer Science and Engineering
Seoul National University
Seoul, 151-742 Korea

"School of Electrical and Computer Engineering
University of Seoul
Seoul, 130-743 Korea

Keyword-based search returns its results without concern for the information needs of users. In general, search queries are too short to represent what users want, and thus it is necessary to represent users' intended semantics more accurately. Our goal is to enrich the semantics of user-specific information (e.g., users' queries and preferences) and documents with their corresponding concepts for personalized search. To achieve this goal, we adopt a Bayesian belief network (BBN) as a strategy for personalized search since the Bayesian belief network provides a clear formalism for mapping user-specific information to its corresponding concepts. Nevertheless, since the concept layer of the Bayesian belief network consists of only index terms extracted from documents, it does not use the domain knowledge which is required for search systems to understand the intended semantics of queries. Therefore, we extend the Bayesian belief network to represent the semantics of user-specific information as concepts (not index terms). The concepts are extracted from a taxonomic knowledgebase such as the Open Directory Project Web directory. In our experiments, we have shown that an extended Bayesian belief network using taxonomic knowledge significantly outperforms the other Bayesian belief network-based approaches and conventional approaches (i.e., query expansion and result processing) for personalized search.

Keywords: taxonomic knowledge, Bayesian belief network, personalized search, conceptual matching, probabilistic model

1. INTRODUCTION

Until now, a popular search paradigm in most search engines is keyword-based search due to its simplicity and efficiency [1]. Most keyword-based search engines exploit the exact match between the index terms derived from documents and user-specific information (e.g., users' queries and preferences), and the actual search queries are too short to represent what users want exactly. Therefore, some documents, which do not include a user’s query, may not be returned to the user although they are semantically relevant to the given query. As a result, it is necessary to exactly represent the intended semantics of users’ queries for more reliable search; this is associated with personalized search.

The quality of personalized search depends on the quality of obtained user-specific
information. When using user-specific information in personalized search, it is important to exactly capture the semantics of the data. In our work, we adopt a Bayesian belief network (BBN) to enrich the semantics of user-specific information since the BBN-based model provides a clear formalism for mapping user-specific information to its corresponding concepts in the concept layer [2, 3]. Moreover, the BBN-based model improves the quality of the ranked list of retrieved documents compared to keyword-based search. Since humans interpret the semantics of specific terms with their domain knowledge or experience, search systems also require vast amounts of domain knowledge to understand the semantics of terms [4, 5]. However, the conventional BBN-based model does not make use of domain knowledge. This is because its concept layer consists of only index terms extracted from documents.

In our work, we extend the conventional BBN-based model to represent the semantics of user-specific information and documents with the concepts (not index terms) of domain knowledge. In recent years, a practical approach for exploiting domain knowledge is to use a taxonomic knowledgebase such as the Web directory. In particular, since the Open Directory Project (ODP) Web directory is the largest and most comprehensive knowledgebase that has been constructed and maintained by a global community of volunteers [6], it can be useful to help search systems to understand the semantics of user-specific information and documents. In our work, each category of the ODP Web directory is regarded as a concept1. By doing so, we can use additional semantic information such as taxonomic relationships between concepts for more accurate and personalized search. Each concept in our search model is described by the terms of descriptions that belong to the concept2. By using relationship information and terms of descriptions that belong to concepts, the semantics of user-specific information can be enriched in our search model; that is, the number of concepts mapped to the given user-specific information is increased.

Through taxonomic knowledge, we can provide search systems with humans’ domain knowledge to understand the intended semantics of users’ queries. As far as we know, our work is the first attempt to apply taxonomic knowledge to the Bayesian belief network for personalized search. Currently, although we use the ODP Web directory as our taxonomic knowledgebase, our approach can be applied to other knowledge bases that have relationship information of concepts and terms of descriptions that belong to the concepts. Furthermore, when a user accesses (or clicks) documents, the user may be affected by document popularity; that is, it is likely that a user accesses popular documents more often than less popular ones. Therefore, our ranking function considers not only individual preferences but also document popularity for personalized search.

The remainder of this paper is organized as follows. Section 2 reviews several work related to personalized search and belief network-based search. Section 3 describes how to extend the Bayesian belief network-based model. Then, it explains the detailed probabilistic approach to ranking documents with respect to the given query. Section 4 presents performance improvements over the other personalized approaches through various experiments. Section 5 concludes this paper.

1 Index terms are referred to as concepts in the conventional BBN-based model, whereas in our model categories of the Web directory are referred to as concepts.

2 Web pages that belong to a concept (or a category) are referred to as the descriptions of the concept. And, the terms extracted from the descriptions are used to define the semantics of the concept.
2. RELATED WORK

2.1 Personalized Search

The goal of personalized search systems is to provide differently ordered results for similar or identical queries when submitted by different users with different information needs [7]. Pitkow et al. [8] have introduced two approaches for personalized search: query expansion and result processing. Query expansion is to expand a user’s query with the user’s preferences (or preferred concepts) to remove the ambiguity of the given query. Result processing is to re-rank search results according to the user’s preferences. Specifically, query expansion focuses on mapping a submitted query to its corresponding concepts in the user’s preferences to remove the ambiguity of the given query [9-14]. Result processing re-ranks the search results in the concepts of the user’s preferences according to the relevance degree with the concepts [1, 6, 15-17]. The concepts are explicitly selected by the user or implicitly derived from the user’s history logs, and the strength of a user’s information need is defined by the relevance degree with the concepts.

In recent studies, the user’s preferences are represented by a set of concepts derived from domain ontologies in which each concept of ontology has its own weight to indicate the user’s interest in the concept. Although some studies have created their own domain ontologies to serve personalized search [12-14], creating domain ontologies is a time consuming task in general. And, it is difficult to find an appropriate domain ontology for the given search systems. In this respect, recent studies regard a taxonomic knowledge-base (such as the Web directory) as a kind of domain ontology [17-19]: that is, each category of the Web directory is regarded as a concept of ontology.

Even though our approach uses the ODP Web directory for personalized search like the previous studies [1, 6, 10, 11, 15-17], it has several features different from the studies. First, since the previous studies only focus on providing the personalized results through query expansion or result processing, they do not consider the semantics of queries and documents; that is, the studies depend mainly upon keyword matching. On the other hand, our approach focuses on capturing the semantics of queries and documents, and provides the personalized results through conceptual matching. Second, some studies [1, 6, 10, 15, 16] use the exact match of index terms for mapping the query (or re-ranking results) to several concepts. As a result, only when the descriptions of a concept contain the terms of the query (or results), the concept is considered to be related to the query (or results). However, our approach assumes that concepts are related to each other. Even though some concepts do not contain the terms of query (or results) in their descriptions, the concepts may be related to the query (or results). Although the authors in [11, 17] consider the relationships between concepts, the chosen categories are too few to exactly represent the semantics of user-specific information. For example, Sieg et al. [17] consider only 563 concepts with a depth of six levels in the ODP Web directory while we consider 11,584 concepts with twelve levels for our experiments.

2.2 Belief Network-based Search

Ribeiro et al. [3] have introduced a Bayesian belief network (BBN)-based model for
search, and compared it to an inference network-based model [20]. Fig. 1 illustrates the structures of the inference network-based model and the BBN-based model.

The inference network-based model is a kind of belief network model first proposed for information retrieval. In the inference network-based model, a variable associated with a document represents the event observing the document. The observation of a document is the cause for observing any of its assigned index terms in the concept layer. This causal relationship is modeled by directing edges in the network from the document layer towards the concept layer. The direction of dependency links between the document and concept layers is opposite to that of the BBN-based model.

Eq. (1) is the statistical representation of the inference network-based model to compute the relationship between a query \( q \) and a document \( d_x \), and Eq. (2) is the statistical representation of BBN-based model.

\[
Pr(q \cap d_x) = \sum_{t_i \in V_r} Pr(q \mid t_i) \cdot Pr(t_i \mid d_x) \cdot Pr(d_x)
\]  
\[
Pr(q \cap d_x) = \sum_{t_i \in V_r} Pr(q \mid t_i) \cdot Pr(d_x \mid t_i) \cdot Pr(t_i)
\]

where \( V_r \) denotes the set of index terms related to the query and documents. \( Pr(q \mid t_i) \) in Eq. (1) is the probability of observing the query \( q \) when the index term \( t_i \) has been observed. \( Pr(t_i \mid d_x) \) corresponds to the probability of observing the index term \( t_i \) when the document \( d_x \) has been observed. In contrast, \( Pr(d_x \mid t_i) \) in Eq. (2) is the probability of observing the document \( d_x \), given that the index term \( t_i \) has been observed. Moreover, \( Pr(t_i) \) and \( Pr(d_x) \) are the prior probabilities of observing the index term \( t_i \) and the document \( d_x \), respectively.

The main difference of Eqs. (1) and (2) is the causal relationship between \( t_i \) and \( d_x \). As shown in Eq. (1), a document \( d_x \) is the cause of the observed concept \( t_i \) in the inference network-based model. Moreover, concepts are also the cause of a query. However, in the BBN-based model, the query and the documents are derived from the given concepts. As a result, the BBN-based model needs to adopt a clearly defined concept space compared to the inference network-based model. Furthermore, separation between the query and the document layers simplifies the modeling task, and makes the BBN-
based model reproduce any ranking strategy generated by the inference network-based model while the converse is not true [2, 3]; that is, the BBN-based model is more general than the inference network-based model.

Despite such benefits of the BBN-based model, it has limitations (i.e., lack of humans’ domain knowledge and relationship information between concepts). To make search systems understand the semantics of a given query and documents, humans’ domain knowledge is required [4, 5]. However, the conventional BBN-based model does not make use of domain knowledge since its concept layer consists only of index terms extracted from documents. Furthermore, it is difficult to define the relationships between concepts (or index terms) since the BBN-based model contains only index terms extracted from documents in the concept layer.

3. EXTENDING BAYESIAN BELIEF NETWORK FOR PERSONALIZED SEARCH

To overcome the limitations of the conventional BBN-based model, we extend the Bayesian belief network-based model by exploiting a taxonomic knowledgebase such as the ODP Web directory in the concept layer. Fig. 2 illustrates the conventional Bayesian belief network (BBN) and extended Bayesian belief network for personalized search ($\rho$EBBN).

As shown in Fig. 2, the query layer of $\rho$EBBN is different from that of BBN. The query layer of $\rho$EBBN contains not only a query but also a user’s preferences for personalization. We refer to the combination of the query and preferences as the enhanced query in Fig. 2 (b), which will be detailed in section 3.1.3. A user’s preferences can be extracted from the user’s log history because the history includes documents accessed by the user. The main difference between BBN and $\rho$EBBN is the structure of the concept layer. As shown in Fig. 2 (a), BBN contains only index terms extracted from documents in the concept layer, thus it is difficult to define the relationships between concepts (or index terms). Although BBN contains co-occurrence information between concepts, the co-occurrence information is statistical. In contrast, since $\rho$EBBN uses a taxonomic knowledgebase in the concept layer, it contains the semantic relationships, which are
based on humans’ domain knowledge, between concepts (or categories). As a result, the relationships may be more accurate than those of statistical approach. By using taxonomical information, we can increase the probability of matching an enhanced query to documents. For example, in Fig. 2 (b), we assume that an enhanced query is mapped to concepts $c_3$ and $c_5$, and a document $d_1$ is mapped to concepts $c_4$ and $c_5$ (see dotted lines). When using the taxonomical relationship that $c_3$ is related to $c_4$ and $c_5$ in this example, the enhanced query may be related to the document $d_1$. If the relationship information is not used, we cannot obtain related documents as a result of the enhanced query.

Furthermore, each concept in pEBBN has descriptions that explain the semantics of the concept. When using descriptions of each concept, the number of concepts that are mapped to an enhanced query or a document can be much larger than that of BBN. For example, in Fig. 2, the concepts $t_1$ and $t_2$ are mapped to the document $d_1$ in BBN, whereas the concepts $c_1$, $c_2$, $c_3$, $c_4$, and $c_5$ are mapped to the document $d_1$ in pEBBN (see dotted lines and solid lines). This is because terms of the given document may be included in the descriptions of all concepts ($c_1$-$c_5$) of pEBBN. In Last.fm data of our experiments, a document is mapped to an average of 13 index terms of the concept layer in BBN. For pEBBN, a document is mapped to an average of 445 concepts of the concept layer.

### 3.1 Modeling Extended Bayesian Belief Network

In this section, we explain how to model the concept, document, and query layers in our pEBBN-based model.

#### 3.1.1 Concept layer

Each concept within a taxonomic knowledgebase has descriptions that discriminate the concept. Because the terms extracted from descriptions can explain the semantics of a concept, we exploit these terms for modeling the concept. Formally, a concept can be modeled as follows.

Given a taxonomic knowledgebase $C = \{c_1, \ldots, c_i, \ldots, c_n\}$, let $C$ be a set of concepts, and $n$ be the total number of concepts. Since we consider the hierarchical relationships between concepts, the semantics of each concept is described not only by itself, but also by its sub-concepts’ descriptions. Each concept $c_i$ is defined as the average of the description term vectors under $c_i$ and its sub-concepts, which is computed as follows,

$$ c_i = \frac{1}{|D_{c_i}|} \sum_{d_j \in D_{c_i}} d_j^i, \text{ where } D_{c_i} = \{d_1^i, \ldots, d_j^i, \ldots, d_m^i\} \cup \bigcup_{s_0 < c_i} D_{s_0}^i $$

(3)

where $c_i \prec c_0$ means that $c_i$ is a sub-concept of the $i$th concept $c_0$ and $D_{c_i}$ is the set of descriptions that belong to $c_i$ and its sub-concepts. $d_j^i$ is the $j$th description in $c_i$. Each description $d_j^i$ is represented as the weighted term vector $d_j^i = \langle w_{j_1}, \ldots, w_{j_k}, \ldots, w_{j_l} \rangle$ where $w_j$ is the weight of the $j$th term, and $V$ is the set of index terms. Each term weight $w_j$ is computed by using the term frequency (TF) and the inverse document frequency (IDF). When computing term weights, we remove the stop words (such as articles, propositions, and conjunctions) from the description of $c_i$ and use the Porter stemming.

---

1 Hereafter, vectors are denoted as boldface.
algorithm [21] to transform inflected words to their stem or root forms.

3.1.2 Document layer

In our extended BBN-based model, documents are mapped to a set of concepts. This section describes how to map a document to the concepts extracted from a taxonomic knowledgebase. To represent a document as a set of concepts, we construct two matrices: the term-concept matrix (TCM) and the document-term matrix (DTM). TCM represents the relationships between terms and concepts. A relationship between the term \( t_k \) and the concept \( c_i \) is defined as follows,

\[
Pr(t_k | c_i) = \frac{w(c_i, t_k)}{\sum_j w(c_i, t_j)}
\]  

(4)

where the numerator \( w(c_i, t_k) \) denotes the weight of the \( k \)th term (denoted as \( t_k \) in \( c_i \)), and the denominator \( \sum_j w(c_i, t_j) \) denotes the total weight sum of all terms in \( c_i \). \( Pr(t_k | c_i) \) in Eq. (4) is an element of TCM, and the normalized value of the \( k \)th term weight in \( c_i \).

DTM represents the relationships between documents and terms. A document \( d_x \) is represented as a \( V \)-dimensional term vector where \( V \) is the set of index term, and each element of the term vector represents the relationship between a term and the document. The relationship between the term \( t_k \) and the document \( d_x \) is defined as follows,

\[
Pr(t_k | d_x) = \frac{w(d_x, t_k)}{\sum_j w(d_x, t_j)}
\]  

(5)

where the numerator \( w(d_x, t_k) \) denotes the weight of the \( k \)th term in \( d_x \), and the denominator \( \sum_j w(d_x, t_j) \) denotes the total weight sum of all terms in \( d_x \). The value, \( Pr(t_k | d_x) \) is an element of DTM.

We can derive a document-concept matrix (DCM) through the matrix multiplication of DTM and TCM. As a result, a document \( d_x \) is represented as the elements of DCM; i.e., \( d_x = \langle Pr(d_x | c_1), ..., Pr(d_x | c_i), ..., Pr(d_x | c_n) \rangle \), where \( Pr(d_x | c_i) \) is the probability of observing the document \( d_x \), given that the concept \( c_i \) has been observed. Actually, since documents and concepts are represented with the term vectors in DTM and TCM, we join the factors of terms into \( Pr(d_x | c_i) \), which explains the relationship between a document and a concept by applying the definition of conditional probability and the law of total probability.

\[
Pr(d_x | c_i) = \frac{1}{\Pr(c_i)} \sum_{t_k} Pr(d_x | t_k) \cdot Pr(c_i | t_k) \cdot Pr(t_k)
\]

By using Bayes’ theorem, the above Eq. is written as follows,

\[
\frac{1}{\Pr(c_i)} \sum_{t_k} \frac{Pr(d_x)}{Pr(t_k)} \cdot \frac{Pr(t_k | d_x)}{Pr(t_k)} \cdot \frac{Pr(c_i | t_k)}{Pr(t_k)} \cdot Pr(t_k)
\]
The approximation process is detailed in [17].

\[
\sum_{k} \frac{Pr(d_k)}{Pr(t_k)} \cdot Pr(t_k \mid d_x) \cdot Pr(t_k \mid c_j)
\]

where \( Pr(c_i), Pr(d_x) \) and \( Pr(t_k) \) are the prior probabilities for the concept \( c_i \), the document \( d_x \) and the term \( t_k \), respectively. For simplicity’s sake, \( Pr(c_i) \) is assumed to be equivalent for all concepts; \( Pr(c_i) = \frac{1}{\text{total number of concepts}} \). Similarly, the other probabilities (i.e., \( Pr(d_x) \) and \( Pr(t_k) \)) are also assumed to be equivalent for their corresponding variables. Consequently, Eq. (6) is rewritten as follows,

\[
Pr(d_x \mid c_j) \propto \sum_{t_k} Pr(t_k \mid d_x) \cdot Pr(t_k \mid c_j)
\]

where \( Pr(t_k \mid d_x) \) is an element of DTM, and \( Pr(t_k \mid c_j) \) is an element of TCM.

3.1.3 Query layer

In pEBBN, the query layer consists of a user’s query and preferences, and we refer to this combination as the enhanced query (denoted as \( q_e \)). This section describes how to map the enhanced query to a set of concepts from the taxonomic knowledgebase. Like the document layer, the enhanced query \( q_e \) is also represented as a concept vector with \( n \) dimensions: i.e., \( q_e = \langle Pr(q_e \mid c_1), \ldots, Pr(q_e \mid c_i), \ldots, Pr(q_e \mid c_n) \rangle \), where \( Pr(q_e \mid c_i) \) is the probability of observing the enhanced query \( q_e \), given that the concept \( c_i \) has been observed. Because the enhanced query is represented as the combination of a query and a user’s preferences, the \( i \)th element of \( n \) dimensions, \( Pr(q_e \mid c_i) \) can be defined as follows,

\[ Pr(q_e \mid c_i) = Pr(q \cup p_u \mid c_i). \]

To compute the union of two probabilities in the classic probability theory, we must consider the intersection probability \( Pr(q \cap p_u \mid c_i) \). However, computing the intersection probability is difficult because a user’s query \( q \) depends on the user’s preference \( p_u \). Therefore, for simplicity’s sake, \( Pr(q_e \mid c_i) \) is approximated to the maximum of \( Pr(q \mid c_i) \) and \( Pr(p_u \mid c_i) \) according to the principle of minimum entropy [22].

\[ Pr(q_e \mid c_i) \approx \{Pr(q_e \mid c_i), Pr(p_u \mid c_i)\} \]

where \( Pr(q \mid c_i) \) and \( Pr(p_u \mid c_i) \) are the probabilities of observing \( q \) and \( p_u \) in the query layer respectively, given that the concept \( c_i \) has been observed. To complete the computation of \( Pr(q_e \mid c_i) \), we must explain how to compute \( Pr(q \mid c_i) \) and \( Pr(p_u \mid c_i) \).

First, let \( q \) be a query, then \( q = \langle Pr(q \mid c_1), \ldots, Pr(q \mid c_i), \ldots, Pr(q \mid c_n) \rangle \), where \( n \) is the total number of concepts in the concept layer. \( Pr(q \mid c_i) \) is the probability of observing the query \( q \), given that the concept \( c_i \) has been observed. Thus, each \( Pr(q \mid c_i) \) can be estimated as follows,

\[ Pr(q \mid c_i) = \frac{w(e_i, q)}{\sum_{t_j} w(c_i, t_j)} \]

\(^4\)The approximation process is detailed in [17].
where the numerator \( w(c_i, q) \) denotes the weight of query \( q \) in \( c_i \) and the denominator \( \sum_{t_j} w(c_i, t_j) \) denotes the total weight sum of all terms in \( c_i \). In Eq. (8), we assume that the query consists of a single term. However, if the query consists of several terms, we use Eq. (7) to compute \( Pr(q | c_i) \) because a query with several terms can be regarded as a document.

Second, let \( p_u \) be a user’s preference in the query layer, then \( p_u = (Pr(p_u | c_1), \ldots, Pr(p_u | c_i), \ldots, Pr(p_u | c_n)) \), where \( n \) is the total number of concepts in the concept layer. \( Pr(p_u | c_i) \) is the probability of observing the preference \( p_u \), given that the concept \( c_i \) has been observed. A user’s preference can be represented with the documents that he/she accessed, and thus we join the factors of documents into \( Pr(p_u | c_i) \) by applying the definition of conditional probability and the law of total probability.\(^5\)

\[
Pr(p_u | c_i) \propto \sum_{d_x \in D} Pr(d_x | p_u) \cdot Pr(d_x | c_i)
\]

where \( Pr(d_x | c_i) \) is an element of DCM, and \( Pr(d_x | p_u) \) is the probability of observing the document \( d_x \) given that the preference \( p_u \) has been observed. \( Pr(d_x | p_u) \) can be estimated by \( \frac{\text{access}(u, d_x)}{\sum_{d_y \in D} \text{access}(u, d_y)} \) where the numerator \( \text{access}(u, d_x) \) is the access count of the user \( u \) for the document \( d_x \), and the denominator \( \sum_{d_y \in D} \text{access}(u, d_y) \) is the total access count of the user \( u \) for all documents in the collection.

### 3.2 Probabilistic Approach to Ranking

Now, we need to explain how to probabilistically match the enhanced query \( q_e \) to documents in the pEBBN-based model. Wong et al. \([22]\) suggested two measures to specify how to rank the documents in the collection: \( Pr(q \mid d_x) \) and \( Pr(d_x \mid q) \). In \([22]\), \( Pr(q \mid d_x) \) is interpreted as a measure of the precision of \( d_x \) with respect to \( q \), and \( Pr(d_x \mid q) \) is interpreted as a measure of the recall of \( d_x \) with respect to \( q \). However, the authors demonstrated that \( Pr(q \mid d_x) \) and \( Pr(d_x \mid q) \) can be made equivalent through proper normalization. Thus, it is not critical which one is used as the ranking function. In our approach, it is more natural to consider that the query is provided as evidence of a user’s information needs. Therefore, we use \( Pr(d_x \mid q) \) as the rank function of the document \( d_x \) with respect to the query \( q \).

Our ranking function \( \text{Score}\(q_e, d_x) \) considers not only \( Pr(d_x \mid q) \) but also document popularity (denoted as \( Po(d_x) \)). This is because we assume that each user may be affected by document popularity when accessing a document. Since users might access popular documents more often than less popular ones, we conclude that popularity will be a critical factor in determining the rank. Since the popularity of documents is in proportion to the access counts (e.g., click-through counts) of users for the documents, we estimate popularity through the access counts. In Eq. (10), the numerator is an access count of all users for the document \( d_x \) and the denominator is the sum of access counts of all users for all documents in the collection.

\[
Po(d_x) = \frac{\text{access}(u, d_x)}{\sum_{d_y \in D} \sum_u \text{access}(u, d_y)}
\]

\(^5\) The derivation of Eq. (9) is identical to that of Eq. (7) by replacing \( d_x \) and \( p_u \) with \( t_i \) and \( d_s \), respectively.
Although document popularity is not derived from pEBBN, we consider popularity in the ranking mechanism for more accurate personalized search. Formally, our ranking function $Score(q_e, d_x)$ is defined as follows.

$$Score(q_e, d_x) = \gamma \cdot Pr(d_x | q_e) + (1 - \gamma) \cdot Po(d_x)$$

(11)

where $\gamma$ is the weight that ranges from 0 to 1. The more $\gamma$ decreases, the more document popularity dominates the score function.

As stated before, a document $d_x$ and an enhanced query $q_e$ are represented as a set of concepts in pEBBN. Thus, the function $Score(q_e, d_x)$ is written as follows according to the definition of conditional probability and the law of total probability.

$$Score(q_e, d_x) = \gamma \cdot \left\{ \frac{Pr(q_e \cap d_x)}{Pr(q_e)} \right\} + (1 - \gamma) \cdot Po(d_x)$$

$$= \gamma \cdot \left\{ \frac{1}{Pr(q_e)} \sum_{c_i} Pr(q_e \cap d_x | c_i) \cdot Pr(c_i) \right\} + (1 - \gamma) \cdot Po(d_x)$$

(12)

As shown in Fig. 2 (b), since instantiation of the concepts logically separates the query layer from the document layer to make them mutually independent, we can modify Eq. (12) as follows,

$$\gamma \cdot \left\{ \frac{1}{Pr(q_e)} \sum_{c_i} Pr(q_e | c_i) \cdot Pr(d_x | c_i) \cdot Pr(c_i) \right\} + (1 - \gamma) \cdot Po(d_x)).$$

Moreover, $Score(q_e, d_x)$ can be rewritten as follows since the prior probabilities $Pr(q_e)$ and $Pr(c_i)$ are assumed to be equivalent for all values of $q_e$ and $c_i$, respectively.

$$Score(q_e, d_x) = \gamma \cdot \left\{ \sum_{c_i} Pr(q_e | c_i) \cdot Pr(d_x | c_i) \right\} + (1 - \gamma) \cdot Po(d_x)$$

(13)

4. EXPERIMENTS

We used 11,584 concepts whose domain is limited to music (i.e., /Top/Arts/Music) from the ODP Web directory. In general, a user’s access (i.e., listening) time in the music domain is shorter than those in other domains (e.g., movies and books), and a user may listen to the same track many times [24]. As a result, the volume of a user’s access log for a certain period is relatively large. Recently, since collecting users’ access logs and analyzing users’ preferences in the music domain are easier than in other domains, the music domain has been widely used to prove the effectiveness of a personalized system [25, 26]. Moreover, our approach can be applied to the other domains easily. Since regarding a document as a term vector\(^6\), we can ignore the attribute types of a document (e.g., movies and books). We examined 100 users’ listening logs that were crawled from

\(^6\)In section 3.1, we mentioned that a document $d_x$ is represented as a $V$-dimensional term vector where $V$ is the set of index terms.
Last.fm. Then, 22,216 tracks crawled from Last.fm were used in the document layers of the belief networks. Each track datum consists of several attributes such as title, artist name, album name, genre, artist type, country, and explanation. For evaluation, we introduce two performance measures: precision\(_k\) and inverse rank precision\(_k\). First, precision\(_k\) is defined as follows,

\[
\text{precision}_k = \frac{\text{number of documents that a user accessed in top-}k\text{ documents}}{k}
\]

We use precision\(_k\) to examine how many documents accessed by a user are retrieved in top-\(k\). Another measure, inverse rank precision\(_k\) is similar to mean reciprocal rank [23]. Mean reciprocal rank is the average of the reciprocal ranks of documents for a sample of queries, and our inverse rank precision\(_k\) is the weighted reciprocal ranks over a query. Thus, as the ranks of retrieved documents are more identical to the top-ranks of correct answer, inverse rank precision\(_k\) is closer to 1. Second, inverse rank precision\(_k\) is defined as follows,

\[
\text{inverse rank precision}_k = \frac{1}{\sum \frac{1}{\text{rank of document according to a user's preferences}}}
\]

where the rank of a document according to a user’s preferences is estimated by the user’s click-through count for the document.

For example, suppose a user issued a query such as ‘rock’ to documents in the music domain. Table 1 shows the correct ranks of documents and the ranks of the documents according to the user’s preferences in different approaches. In this example, precision\(_3\) values of approaches ‘A’ and ‘B’ are the same (i.e., precision\(_3\) is 2/3 = 0.67).

<table>
<thead>
<tr>
<th>Documents</th>
<th>Correct Rank</th>
<th>Rank According to Preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Approach A</td>
</tr>
<tr>
<td>By the way</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Bad day</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Go to sleep</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1. Ranks of documents according to a user’s preferences.

Inverse rank precision\(_3\) values of approaches ‘A’ and ‘B’ are computed as follows,

\[
\text{inverse rank precision}_3 \text{ of ‘A’} = \frac{1}{\frac{1}{2} + \frac{1}{1} + \frac{1}{4}} = 0.955,
\]

\[
\text{inverse rank precision}_3 \text{ of ‘B’} = \frac{1}{\frac{1}{2} + \frac{1}{3} + \frac{1}{6}} = 0.818.
\]

\(1\) Listening logs crawled from Last.fm consists of user ID, artist name, track title, and timestamps.
Inverse rank precision of the approach ‘A’ is larger than the one of the approach ‘B’. Thus, we can say that the approach ‘A’ reflects a user’s top-ranked preferences better than the approach ‘B’, although they show the same value in a user’s overall preferences. That is, precision measures how much the system can reflect the user’s overall preferences, whereas inverse rank precision measures how much the system can reflect the user’s top-ranked preferences. The closer inverse rank precision gets to 1.0, the more the search results match with the user’s preferences.

4.1 Relevance Judgment for Personalized Search using Tags

In the evaluation of personalized search, the main obstacle is how to judge the relevance of search results to a user’s information needs. In general, the relevance judgment approaches are classified into two categories [1]: user-based judgment [6, 8, 15] and log-based judgment [11, 16, 17]. Since the user-based judgment approach makes many users to conduct user-specific relevance judgments for search results, this approach causes high costs in experiments. Moreover, since the users who involved in the experiments know that they are being tested, the users may bias the experiment results [1]. In contrast, the log-based judgment approach causes lower costs and biases than the user-based judgment approach, and however this approach needs to collect a lot of query logs for experiments. Actually, since most search engines such as Google and Yahoo! do not release their query logs to protect users’ privacy, collecting a lot of query logs is not possible for most of researchers including us. Due to the limitations of the previous approaches to relevance judgment, Xu et al. [1] have proposed the tag-based approach to relevance judgment with the following Assumption 1.

Assumption 1 A user’s tagging action reflects the user’s personal relevance judgment.

If a user assigned a tag ‘Jazz’ to a document, the user will regard this document as relevant when the user issues ‘Jazz’ as a query. Since our experimental environment is similar to that of Xu et al. [1], we follow their strategy to judge the relevance of search results.

However, since our data (i.e., users’ listening logs in Last.fm) do not contain any information on who tagged the particular document (i.e., track in case of the music domain), we propose the following assumption in our experiments.

Assumption 2 A user, who accessed a particular document, is related to the tags attached to the document.

For example, a user who listened to a track ‘yesterday’ of ‘the Beatles’, the user may be related to the tags such as ‘British’, ‘class’, ‘sad’ which are attached to the track. Of course, the user may not be truly related to the tags. However, this assumption makes tag-based relevance judgment to be flexible; that is, although there is no information on who tagged a track, tag-based relevance judgment can be applied to evaluate personalized search systems. Finally, we propose the following assumption to determine the correct ranks of retrieved documents which is shown in Table 1.
Assumption 3  A user’s click-through counts reflect the user’s preference degree (or rank) on a document.

If a user’s click-through count for ‘By the way’ is the highest in the example of Table 1, we assume that the user’s preference degree on the track ‘By the way’ is also the highest. As a result, the track ‘By the way’ is top-ranked in the given user’s preferences; that is, the ranks that are derived from users’ click-through counts are regarded as the correct ranks of documents according to the users’ preferences.

4.2 Queries

To evaluate the performance of personalized search, we have to examine the queries that are submitted by each user. This is because the queries depend on individual preferences. However, due to the privacy issues as mentioned in section 4.1, it is difficult to collect a user’s query log in the real world. If the queries that are randomly generated are used, there is no way to judge the relevance between a query and search results. As a result, we regard the tags that were crawled from Last.fm as test queries like in Xu et al. [1]. Tags may reflect users’ information needs or preferences. Moreover, tags are useful evidence to evaluate the difference between conceptual matching and keyword matching since the terms of tags may not appear in the documents. Before choosing queries for experiments, we performed two preprocessing steps. First, we manually removed overly personal or meaningless tags, such as ‘my favorite’, ‘yeaaaa’ and the stop words. Second, we removed tags written in non-English terms because our knowledgebase was written in English. Then, we chose the most popular 100 tags as test queries. For reliable evaluation, we created five query sets named QS1, QS2, QS3, QS4, and QS5. Each set was composed of 15 randomly chosen queries.

4.3 Comparison of Search Approaches

In this section, we compare five personalized search approaches that are summarized in Table 2.

$p_{RP}$ and $p_{QE}$ refer to result processing and query expansion approaches for personalized search, respectively. In our experiments, query expansion and result processing for personalized search are referred to as $p_{QE}$ and $p_{RP}$ to discriminate them from non-personalized approaches (i.e., QE and RP) in section 4.5. Particularly, $p_{RP}$ re-ranks the search results in order of relevance degrees with the concepts (i.e., categories of ODP Web directory) derived from a user’s preferences after finding search results matched to a given query. After mapping a given query to related concepts, $p_{QE}$ merges the search results matched to the given query and the index terms of the concepts. In this work, $p_{RP}$ is based on the approach of Xu et al. [1], and $p_{QE}$ is based on the approach of Liu et al. [11] among the previous studies introduced in section 2.1. This is because their approaches are the most up-to-date and similar to our work although their approaches are based on keyword matching rather than conceptual matching. For experimentation fairness, the query layer of $p_{BBN}$ is combined with a user’s preferences for personalization like $p_{EBBN}$ although the conventional BBN [3] (see Fig. 2 (a)) does not include a user’s preferences in the query layer. And, $p_{BBN}$ exploits co-occurrence information of concepts.
Table 2. Five approaches for evaluation.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result Processing for Personalized Search</td>
<td>The $p$RP approach re-ranks the search results in the order of relevance degrees with the concepts (i.e., categories of the ODP Web directory) derived from a user’s preferences after finding search results matched to a given query. Our implementation of $p$RP is based on the approach of Xu et al. [1], which derives a user’s preferences automatically from the ODP Web directory.</td>
</tr>
<tr>
<td>Query Expansion for Personalized Search</td>
<td>After mapping a given query to related concepts (i.e., categories of the ODP Web directory) derived from a user’s preferences, the $p$QE approach merges the search results matched to the given query and the index terms of the concepts. Our implementation of $p$QE is based on the approach of Liu et al. [11], which derives a user’s preferences automatically from the ODP Web directory.</td>
</tr>
<tr>
<td>Bayesian Belief Network for Personalized Search</td>
<td>The query layer of $p$BBN is combined with a user’s preferences for personalization, although the conventional BBN [3] does not include a user’s preferences in the query layer (see Fig. 2 (a)). And, $p$BBN exploits co-occurrence information of concepts (i.e., index terms derived from documents) when mapping an enhanced query (i.e., a user’s query and preferences) and documents to their corresponding concepts.</td>
</tr>
<tr>
<td>$p$EBBN (The proposed Approaches)</td>
<td>$p$EBBN-NR does not exploit relationship information between concepts (i.e., categories) extracted from a taxonomic knowledgebase such as the ODP Web directory when mapping an enhanced query (i.e., a user’s query and preferences) and documents to the concepts.</td>
</tr>
<tr>
<td>$p$EBBN-R</td>
<td>$p$EBBN-R exploits relationship information between concepts (i.e., categories) extracted from a taxonomic knowledgebase such as the ODP Web directory when mapping an enhanced query (i.e., a user’s query and preferences) and documents to the concepts.</td>
</tr>
</tbody>
</table>

(i.e., index terms derived from documents), when mapping an enhanced query and documents to the concepts. $p$EBBN maps queries and documents to the categories of the Web directory, and has two distinct approaches: $p$EBBN-R and $p$EBBN-NR. In $p$EBBN-R, concepts (i.e., categories) are assumed to be related to each other, which is expressed by Eq. (3). Hence, the semantics of a concept are described not only by itself, but also by its sub-concepts’ descriptions. In contrast, since concepts are assumed not to be related to each other in $p$EBBN-NR, each concept can be explained only with descriptions that belong to the concept. In this experiment, the value of $\gamma$ in Eq. (13) was set to 0.9 in $p$EBBN (i.e., $p$EBBN-NR and $p$EBBN-R) and 0.8 in $p$RP, $p$QE and $p$BBN, which will be explained in section 4.4. The number of concepts mapped to the enhanced query was 2,500, which will be detailed in section 4.5.

Fig. 3 shows the mean precision and the mean inverse rank precision of various search approaches according to query sets. As seen in this figure, $p$EBBN-R provides the most accurate results for all query sets. By using taxonomical information, $p$EBBN-R can increase the possibility that a given query is mapped to documents compared to $p$EBBN-NR. In addition, by comparing $p$BBN with $p$EBBN-NR we can see the effect of representing the enhanced query and documents as a set of concepts instead of index terms. Specifically, since each concept is described by the descriptions in $p$EBBN, the number of concepts that are mapped to an enhanced query is larger than those of $p$BBN and $p$QE.
By analyzing the properties of the query sets, we saw that the number of concepts mapped to a query set affected the search accuracy. In other words, the greater the number of concepts mapped to the query becomes, the greater the possibility of retrieving semantically related documents becomes. In this experiment, $p_{RP}$ provides the worst results for all query sets since it exploits the exact match between a given query and documents; that is, as the number of concepts mapped to the query is the least, the values of the mean precision and the mean inverse rank precision are the lowest among five approaches.

### 4.4 Effect of Weight $\gamma$

In Eq. (13), we have introduced the weight $\gamma$ which ranges from 0 to 1. If $\gamma$ is 1, then the rank between documents and an enhanced query is dominated only by $Pr(d_i|q_e)$. In contrast, as $\gamma$ decreases, document popularity $Po(d_i)$ becomes to dominate the rank. In this experiment, we varied the weight from 1.0 to 0.1. Fig. 4 shows the mean precision and mean inverse rank precision for the query set QS1. Since the patterns of other query sets were similar to QS1, we abbreviated their results.

We can see that both mean precision and mean inverse rank precision of $p_{EBBN}$ (i.e., $p_{EBBN-NR}$ and $p_{EBBN-R}$) are the highest when the weight $\gamma$ is 0.9. Moreover, for $p_{BBN}$ and $p_{QE}$, performance is the highest when $\gamma$ is 0.8. As the weight decreases, mean precision and mean inverse rank precision also decrease because document popularity represents the masses’ preferences which may be opposite to the personal preferences.
Moreover, the popular documents may not be related to the given query. In this experiment, we can see that considering document popularity can improve the search performance compared to considering only personal preferences. However, in case of $p_{RP}$, document popularity does not affect mean precision$_{20}$ and mean inverse rank precision$_{20}$. This is because the number of documents related to a given query is smaller than that of $p_{EBBN-NR}$, $p_{EBBN-R}$, $p_{BBBN}$ or $p_{QE}$. Since the value of $Pr(d_s \mid q_e)$ is much larger than that of $Po(d_s)$ in Eq. (11), the effect of document popularity is ignored in $p_{RP}$.

4.5 Effect of the Number of Concepts

When representing an enhanced query and documents as the concepts of ODP, it is critical to determine the number of concepts mapped to the enhanced query and documents; that is, the more the number of concepts increases, the more the overhead (i.e., execution time) increases. From the point of view of the application, the number of concepts mapped to the enhanced query is more critical than the number of concepts mapped to documents in $p_{QE}$, $p_{BBN}$, and $p_{EBBN}$ (i.e., $p_{EBBN-NR}$ and $p_{EBBN-R}$). It is because representing documents as a set of concepts can be done in off-line, but representing the enhanced query with concepts has to be done in real time according to a user’s information needs. However, in case of $p_{RP}$, the number of concepts which are related to documents is more critical because $p_{RP}$ re-ranks the documents$^8$ in order of relevance degrees between documents and concepts in real time. Thus, we intend to observe the effect of the number of concepts mapped to an enhanced query in $p_{QE}$, $p_{BBN}$ and $p_{EBBN}$, and the effect of the number of concepts mapped to documents in $p_{RP}$.

Fig. 5. Effect of number of concepts: mean precision$_{20}$

Fig. 5 shows the mean precision$_{20}$ of $p_{RP}$, $p_{QE}$, $p_{BBN}$, and $p_{EBBN}$. The maximum number of concepts is changed from 100 to 4,000. This experiment was performed for only QS1 since the patterns of the other query sets was similar to QS1. The value of $\gamma$ was set to 0.9 in $p_{EBBN}$ (i.e., $p_{EBBN-NR}$ and $p_{EBBN-R}$) and 0.8 in $p_{RP}$, $p_{QE}$ and $p_{BBBN}$ like the other experiments. As shown in Fig. 5, the mean precision$_{20}$ is saturated when the maximum number of concepts is larger than 2,500 in $p_{BBBN}$ and $p_{EBBN}$. Otherwise, the mean precision$_{20}$ is proportionate to the number of concepts. However, the performance improvements of $p_{RP}$ and $p_{QE}$ are not significant because the weights of

$^8$ Listening logs crawled from Last.fm consists of user ID, artist name, track title, and timestamps.
added concepts do not contribute to the retrieval performance. Table 3 shows the execution time of \( p_{RP} \), \( p_{QE} \), \( p_{BBN} \), and \( p_{EBBN} \) according to the maximum number of concepts mapped to enhanced queries (or documents). The execution time of five approaches increases as the maximum number of concepts increases. Particularly, the execution time in \( p_{QE} \) and \( p_{BBN} \) is faster than that in \( p_{EBBN} \) since the number of concepts mapped to the enhanced query in \( p_{QE} \) and \( p_{BBN} \) is less than that in \( p_{EBBN} \). In case of \( p_{RP} \), the execution time increases with the number of concepts mapped to documents since the search results have to be re-ranked in the order of relevance degrees for the concepts. By analyzing the experimental results, we have found that the number of concepts mapped to the enhanced queries derived from QS1, QS2, QS3, QS4, and QS5 is actually smaller than 2,500. Thus, the improvement of the mean precision and the execution time in \( p_{QE} \), \( p_{BBN} \) and \( p_{EBBN} \) are not significant when there are more than 2,500 concepts.

Table 3. Comparison of execution time (sec.) of five approaches for personalized search.

<table>
<thead>
<tr>
<th>Number of Concepts</th>
<th>( p_{RP} )</th>
<th>( p_{QE} )</th>
<th>( p_{BBN} )</th>
<th>( p_{EBBN-NR} )</th>
<th>( p_{EBBN-R} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.0548</td>
<td>0.4923</td>
<td>0.54</td>
<td>0.4422</td>
<td>0.8271</td>
</tr>
<tr>
<td>200</td>
<td>1.3693</td>
<td>0.4865</td>
<td>0.5868</td>
<td>4.182</td>
<td>4.6774</td>
</tr>
<tr>
<td>300</td>
<td>2.4137</td>
<td>0.4954</td>
<td>0.5477</td>
<td>6.0315</td>
<td>6.3267</td>
</tr>
<tr>
<td>400</td>
<td>3.1517</td>
<td>0.5023</td>
<td>0.6431</td>
<td>8.2318</td>
<td>9.3959</td>
</tr>
<tr>
<td>500</td>
<td>4.9497</td>
<td>0.5279</td>
<td>0.6453</td>
<td>12.0315</td>
<td>11.4085</td>
</tr>
<tr>
<td>600</td>
<td>5.1599</td>
<td>0.5097</td>
<td>0.6258</td>
<td>11.9719</td>
<td>12.1633</td>
</tr>
<tr>
<td>700</td>
<td>6.0691</td>
<td>0.5553</td>
<td>0.6167</td>
<td>12.3493</td>
<td>12.5199</td>
</tr>
<tr>
<td>800</td>
<td>6.5727</td>
<td>0.5621</td>
<td>0.6348</td>
<td>12.6601</td>
<td>12.5454</td>
</tr>
<tr>
<td>900</td>
<td>6.3926</td>
<td>0.5614</td>
<td>0.6589</td>
<td>12.904</td>
<td>12.8038</td>
</tr>
<tr>
<td>1000</td>
<td>7.0688</td>
<td>0.579</td>
<td>0.676</td>
<td>13.2009</td>
<td>13.0461</td>
</tr>
<tr>
<td>1500</td>
<td>7.5868</td>
<td>0.6066</td>
<td>0.6978</td>
<td>13.5749</td>
<td>13.9869</td>
</tr>
<tr>
<td>2000</td>
<td>7.6314</td>
<td>0.6314</td>
<td>0.6203</td>
<td>13.9603</td>
<td>14.2311</td>
</tr>
<tr>
<td>2500</td>
<td>7.7862</td>
<td>0.6325</td>
<td>0.6174</td>
<td>14.6113</td>
<td>14.7076</td>
</tr>
<tr>
<td>3000</td>
<td>8.185</td>
<td>0.6393</td>
<td>0.6374</td>
<td>14.4338</td>
<td>14.5977</td>
</tr>
<tr>
<td>3500</td>
<td>8.1208</td>
<td>0.6382</td>
<td>0.6359</td>
<td>14.3132</td>
<td>14.813</td>
</tr>
<tr>
<td>4000</td>
<td>8.3941</td>
<td>0.6409</td>
<td>0.6392</td>
<td>14.3952</td>
<td>14.7268</td>
</tr>
</tbody>
</table>

4.6 Effect of Personalization

In this section, we compare several personalized search approaches summarized in Table 2 with their non-personalized approaches. The non-personalized search approaches (i.e., \( p_{RP} \), \( p_{QE} \), \( p_{BBN} \), \( p_{EBBN-NR} \), and \( p_{EBBN-R} \)) do not consider a user’s preferences when searching documents related to the given query. Particularly, since the elimination of a user’s preferences in \( p_{RP} \) and \( p_{QE} \) is identical to the conventional keyword-based search model, we include only the performance of \( p_{QE} \) in this section. Furthermore, \( p_{BBN} \), \( p_{EBBN-NR} \), and \( p_{EBBN-R} \) do not also consider a user’s preferences in the query layer, though their concept layer structures remain intact.

Fig. 6 shows the mean precision and mean inverse rank precision of eight search approaches. The value of \( \gamma \) and the number of concepts mapped to an enhanced query are identical to those in section 4.3. Personalized approaches such as \( p_{QE} \), \( p_{BBN} \), \( p_{EBBN}-\)
NR and pEBBN-R have larger mean precision$_{20}$ and mean inverse rank precision$_{20}$ than non-personalized approaches for all query sets. In personalized approaches, pEBBN-R has the greatest mean precision$_{20}$ and mean inverse rank precision$_{20}$ in all query sets. And, the performance of pEBBN (i.e., pEBBN-NR and pEBBN-R) is significantly enhanced compared to the others. From this experiment, we can derive the effect of personalization in which the semantics of user-specific information are represented as the concepts of a taxonomic knowledgebase. That is, by representing users’ preferences as their corresponding concepts in pEBBN, the semantics of the users’ preferences can be represented more accurately than those in pQE and pBBN.

5. CONCLUSION

In this paper, we have proposed an extended Bayesian belief network model (i.e., pEBBN) to improve the accuracy of personalized search. The key idea behind the proposed model is that documents and the enhanced query are statistically mapped to their corresponding concepts. Two main contributions of this paper are (1) to represent the semantics of documents and the enhanced query as a set of concepts that are extracted from a taxonomic knowledgebase such as ODP, and (2) to propose a probabilistic score function for ranking documents with respect to the enhanced query in a probabilistic
approach. In our experiments, we have shown that the extended Bayesian belief network-based model using the relationships of concepts (i.e., pEBBN-R) is the best one. Particularly, the extended Bayesian belief network based model achieves more accurate search using relationship information compared to other BBN-based models. By using relationship information of concepts, the possibility of matching a query with documents can be increased, and moreover considering document popularity helps to improve the performance to some extent. As future work, we plan to improve the execution time of pEBBN. Although pEBBN can provide more personalized search results that the other approaches, its execution time is too large to use in real search systems as shown in Table 3. Thus, we plan to improve the execution time by inventing an index structure or feature selection techniques in future work.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 20110017480).

REFERENCES

11. F. Liu, C. Yu, and W. Meng, “Personalized Web search for improving retrieval ef-


Jae-Won Lee (李在邁) received the B.S. degree in Information and Telecommunication Engineering from Soongsil University, Seoul, Korea in 2003. He is currently a Ph.D. candidate at the School of Computer Science and Engineering in Seoul National University, Seoul, Korea. His research interests include collaborative filtering, ontology and personalized search.

Han-Joon Kim (金漢峻) received the B.S. and M.S. degrees in Computer Science and Statistics from Seoul National University, Seoul, Korea in 1994 and 1996 and the Ph.D. degree in Computer Science and Engineering from Seoul National University, Seoul, Korea in 2002, respectively. He is currently an associate professor at the School of Electrical and Computer Engineering, University of Seoul, Korea. His current research interests include text mining, machine learning, and intelligent information retrieval.

Sang-Goo Lee (李相求) received the B.S. degree in Computer Science and Statistics from Seoul National University, Korea in 1985. He also received the M.S. and Ph.D. in Computer Science from Northwestern University at Evanston, Illinois, USA in 1987 and 1990, respectively. He is currently a professor at the School of Computer Science and Engineering, Seoul National University, Seoul, Korea. He has been the director of Center for e-Business Technology at Seoul National University, since 2001. His current research interests include e-business technology, database systems, product information management, and semantic technology.