Int J Adv Manuf Technol (2010) 48:367–382
DOI 10.1007/s00170-009-2268-4

ORIGINAL ARTICLE

Ontology mapping-based search with multidimensional similarity and Bayesian network

Min Jung · Hong-Bae Jun · Kyun-Woo Kim · Hyo-Won Suh

Received: 7 May 2009 / Accepted: 10 August 2009 / Published online: 27 August 2009

Abstract Users in enterprise information systems want to efficiently search the information that they need. Although several searching approaches have been proposed so far, they still have the limitation in finding the semantically similar information that users need. To overcome the limitation, it is essential to consider the semantics of user keyword and terms (concepts) stored in the ontology repository and continuously update the ontology repository for information searching. To this end, in this study, an ontology mapping-based search methodology (OntSE) is proposed. The OntSE consists of three phases: ontology building, ontology mapping, and ontology updating. Its objective is to find the terms which have the same semantics with user's keywords, based on multidimensional similarity and Bayesian network. To show the benefits of the proposed methodology, a case study has been carried out.

Keywords Ontology · Semantic mapping · Searching · Ontology similarity · Taxonomy · Bayesian network

M. Jung · K.-W. Kim · H.-W. Suh (✉)
Department of Industrial Engineering,
KAIST, 373-1, Kusong-dong, Yusong-ku,
Daejeon 305-701, Republic of Korea
e-mail: hw_suh@kaist.ac.kr

M. Jung
e-mail: hamstein@kaist.ac.kr

K.-W. Kim
e-mail: kun00@kaist.ac.kr

H.-B. Jun
Department of Industrial and Information Engineering,
Hongik University,
72-1 Sangsu-dong, Mapo-gu,
Seoul 121-791, Republic of Korea
e-mail: hongbae.jun@hongik.ac.kr

M. Jung · K.-W. Kim · H.-W. Suh
Department of Industrial Engineering,
Korea Advanced Institute of Science and Technology (KAIST),
335, Gwa-hak ro, Yu-seong gu,
Daejeon 305-701, Republic of Korea

1 Introduction

Modern manufacturing environment is highly distributed and collaborative. The users in this environment should search the information that they need from lots of documents accumulated throughout their works. In order to facilitate their works, it is important to help them efficiently locate necessary information in the related documents. Otherwise, they will spend a lot of time on searching in vain, which will directly affect the performance of their works such as product design among several partners, supply chain management over several companies, and so on. Ideally, if the search keyword of users is clearly stated without any ambiguity, it would not be difficult for search engine to help them locate the relevant information in related documents. However, the challenging problem is that users in various enterprise information systems handle the information in their own ways. They usually define new information and also search necessary information with their own terms. Furthermore, depending on users, the semantics of their terms are usually different. For example, some users use the same semantics with different terms. As a result, it is required to develop some semantic-based searching methods for providing the proper information that users need, as many researchers [27, 31] already pointed out.

To this end, ontology-based search methods have been highlighted from academia [14, 17, 29, 33], with the advent of semantic web technologies. Unlike the keyword-based search that users simply get necessary information with a few user-defined keywords, the ontology-based search methods apply the semantics of keywords to searching. Ontology specifies the semantic meaning of a keyword explicitly and logically so that a computer easily understands it and efficiently searches the information. However, previous ontology-based search approaches also have some limitations as follows: first, it is not easy to find what users exactly look for because they have the limitation in providing user-intended information considering the degree of similarity between keywords' semantics and term's semantics already built in an ontology repository. To provide the information that users want in a more exact way, it is crucial to adequately use the user's description of a keyword and user's decision. Furthermore, it is necessary to continuously update the ontology repository based on user decision results. However, previous approaches have the limitations in those points. To cope with these limitations, in this study, we propose an ontology mapping-based search method (OntSE). In the OntSE, in order to more precisely look for the user-intended information, the degree of multidimensional similarity is used as a reference value for guiding user selection. Moreover, to update the ontology repository with user selection results, Bayesian network (BN) is applied.

The OntSE consists of three phases: (1) *ontology building*: defining the ontology of user's keyword; (2) *ontology mapping*: ontology-based mapping between user keyword and terms (concepts) stored in an ontology repository; and (3) *ontology updating*: updating the ontology repository. Through three phases, users can efficiently search the information that they want. In the first phase, to describe the ontology of user keyword, we refer to the six-tuple ontology structure of the Karlsruhe ontology and semantic web tool suite (KAON) [3] and use description logic (DL) [1]. In the second phase, in order to find the terms which have the same or similar semantics to user's keyword, the ontology mapping is performed considering the degree of multidimensional similarity including character similarity, taxonomy similarity, and definition similarity. In the last phase, the BN [15] is used to consider the effect of mapping results into ontology repository. By this way, the ontology repository is continuously updated with users' keyword ontology and its mapping result.

The remainder of this study is organized as follows. In Section 2, the related works are discussed. Section 3 introduces the framework and procedure of OntSE. In Section 4, we propose the template for a user to define his/her keyword and its informal description and how to transform the informal description into a formal internal ontology. In Section 5, ontology mapping is comprehensively discussed with two subprocesses: taxonomy update and multidimensional similarity calculation. The update of ontology repository is also discussed in Section 6. In order to demonstrate the benefit of OntSE, a case study is carried out in Section 7. Finally, conclusions and discussions are described in Section 8.

2 Related works

There have been several research works on information search methods. The most common method is the keyword-based search which finds the same string patterns specified by users [29]. However, it does not exploit the semantics of keywords in searching. Hence, searching results are often unsatisfactory. To overcome this limitation, the ontology-based semantic search has attracted more attention in the information retrieval community. The ontology-based semantic search method compares the semantics of keywords with the semantics of terms in repository to obtain better answers. Previous ontology-based search methods can be classified into three types: searching with "annotation," "relevance calculation," and "query expansion."

The annotation approaches [14, 16, 17, 29] search information within the documents which are annotated with their semantic information [18]. Although they can increase the quality of information retrieval in a small-sized environment, they require tedious and labor-intensive tasks to annotate all documents. To overcome this limitation, ontology-based relevance calculation approaches [9, 29, 32, 34] have been proposed. The relationships between documents and ontology are defined, and the relevance between documents or between a document and a user's query is calculated based on the ontology. Then, retrieved documents can be further prioritized by their degrees of relevance [34]. As a result, users can get appropriate documents more conveniently. However, the relevance calculation approach also has limitation in the sense that it is hard to retrieve the documents which contain the semantically same concepts of user's keyword because the semantic description of the keyword is not provided.

To cope with this limitation, many research works [5–7, 12, 17, 18, 29, 33] expanded user's query with domain ontology, named "query expansion approach." Their works can be classified into two types: build time approach and run time approach. The build time approaches [5, 7, 12, 17, 29] use ontological relationships which have been defined in domain ontology. For example, Braga et al. [5] proposed a query expansion method which utilized domain ontology to find relevant concepts of user's queries. Sugumaran and Story [29] used a domain ontology which has synonymous relationships between user's queries and concepts in repository. Moreover, Dey et al. [7] found superconcepts, subconcepts, and synonymous concepts of user's keyword from WordNet [10] as well as domain ontology. In addition, Ha and Park [17] inferred semantically equivalent concepts of user's keyword through "sameClassAs" axioms of a domain ontology.

The build time approach can find documents including the meanings of enriched keywords. However, it cannot guarantee semantically precise mapping because the expansion of query is based on predefined domain ontology, which may not infer the intention of user's keyword. In that case, we cannot expect that appropriate documents including the meaning of user's keyword will be retrieved. On the contrary, the run time approaches [6, 18, 19, 33] infer semantically relevant concepts from the mapping between domain ontology and a run time description of a user's query. Thus, it can infer the information having precise meaning of the user's query. There have been some research works on it. For example, Varga et al. [33] proposed an approach to provide a keyword with its superconcepts and subconcepts for searching, which are compared with domain ontology. Chiang et al. [6] proposed a smart web query method that performs semantic retrieval of web data by employing context ontologies interactively and semantic search filters. They used domain semantics represented as context ontologies to specify and formulate appropriate web queries to search. Kim et al. [18] proposed a method for searching product information from different shopping malls. It interacts with users to specify relevant product categories and properties. It then finds the mapping relationships between user's product descriptions and product ontologies of shopping malls by referring a generic ontology such as WordNet [25]. In their other work [19], they have proposed an agent-based intelligent search framework for product information using ontology mapping. They tried to enhance the expressive power in representation of customer's search intent and to allow customers to search product information based on their own context. To this end, they proposed a customer's product search intent representation scheme and an ontology mapping algorithm regarding both taxonomy and attributes.

Although the run time approaches consider user's intention interactively and more precisely, there are still some limitations. The main limitation is that their performances depend on predefined domain ontologies, which should be developed based on agreements of domain experts prior to searching. However, if there is no efficient update mechanism for domain ontology, it will take many efforts to continuously manage and update the domain ontology because there are many users from various domains to use the domain ontology for information searching. To overcome this limitation, the OntSE provides the mechanism to update the domain ontology with the matched concepts selected by users. It requires keyword and its description such as superconcepts, relations, or relevant concepts provided by users. With the description, the OntSE provides a new taxonomy, updated with a keyword and existing concepts which have been provided by former users. Furthermore, the OntSE continuously updates the domain ontology with matching results based on BN and similarity calculation between concepts. Hence, the user can use the updated domain ontology for information search.

Currently, ontology mapping using BN is an active issue in the ontology community. There have been several techniques using BN in ontology mapping. For example, Helsper and Gaag [13] firstly proposed a method to connect BNs with ontology. They showed that ontology provides an appropriate framework for BN construction. Later, the BN has been applied into the field of ontology mapping [8, 26, 28]. Recently, Tang et al. [30] have proposed risk-minimization-based ontology mapping to investigate the ontology mapping problem considering Bayesian decision theory. Kim et al. [20] have proposed a conversational agent that infers the intentions of the user based on BN and their semantic information. However, the quantitative aspect of the BN has not been explored in detail, and the work to get probability distributions is left for "domain experts." It is not an easy task for the domain expert to define the probability distribution of the BN, especially the high-dimensional conditional probability table (CPT). Some research works [26, 28] tried to resolve this issue through several approaches. For example, Mitra et al. [26] tried to collect the data for probability distribution in a statistical way, and Pan et al. [28] tried to use the iterative proportional fitting procedure approach that helps experts to generate the required CPT in a consistent way. In this study, we define a similarity-based probability generation like the works of Blok et al. [2] and Lee et al. [22], in which the probability distribution can be defined by the similarity in the semantic structure of the ontology.

3 OntSE: framework

The framework of the OntSE is composed of four libraries and three modules. The four libraries consist of user ontology library (UOL), internal formal ontology library (IOL), taxonomy ontology library (TOL), and matched ontology library (MOL). They are used to store the four types of ontologies such as user ontology (UO), internal formal ontology (IO), taxonomy ontology (TO), and matched ontology (MO). These libraries are updated based on the user's keyword ontology and matching results. The three modules of OntSE are: ontology building, ontology mapping, and ontology updating.

The searching procedure of OntSE is conducted based on the interactions between these four libraries and three modules as shown in Fig. 1. It can be chronologically explained based on the sequence of three modules. In the ontology-building module, a user inputs a keyword and the necessary information of the keyword according to the guideline based on KAON's ontology structure [3]. In order

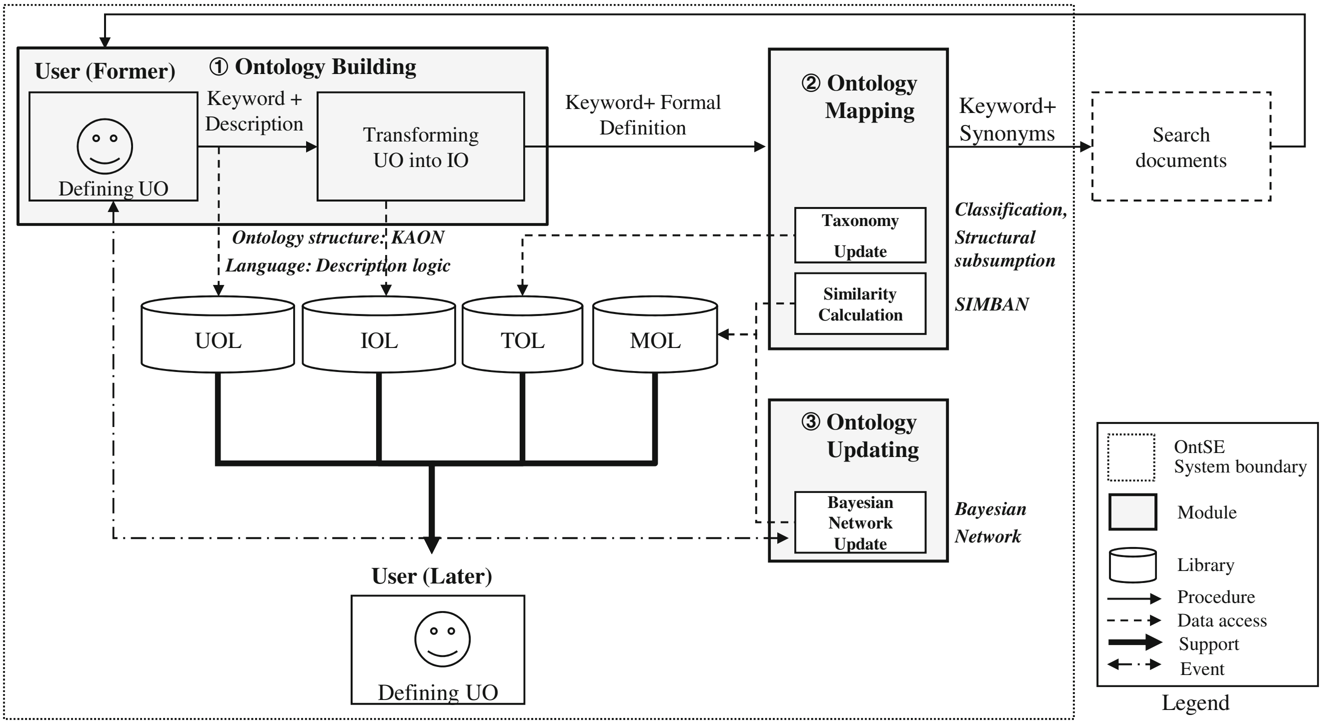


Fig. 1 The architecture of the OntSE

to help users in defining UO, the OntSE can provide a taxonomy updated with a keyword and existing term's ontological definitions which have been accumulated by former users. In this module, the defined UO is stored in the UOL. Then, the OntSE transforms the UO into an IO used for taxonomy reasoning and ontology mapping. IOs are represented in DL. The transformed IOs are stored in the IOL. In the ontology mapping module, the OntSE performs ontology mapping between keyword's IO and existing IOs in IOL and recommends similar concepts to the user's keyword. For this ontology mapping, OntSE calculates the degree of multidimensional similarity including character similarity, taxonomy similarity, and definition similarity. With the calculated similarity, the OntSE then requests user's decision for final matching between the several recommended terms and the user's keyword. After user decision, matched concepts which are the semantically same or similar relationship are stored in MOL. In the ontology updating module, BN is applied for updating the similarities between the concepts in the IOL. In the BN, its CPTs are updated with matched concepts and saved in the MOL for later users. In the following sections, we describe the details on the three modules of OntSE.

4 Ontology building

The OntSE requires the formal definition of user's keyword and its description. However, it is hard for users to define the formal definition because the formal definition is not for human interpretation but for computer understanding. To resolve this problem, the OntSE provides a template that can guide users to define UO in an easy manner. After the OntSE gets a UO from a user, it transforms the UO into an IO for computer processing.

4.1 Defining UO

For designing the template of UO, we refer to the six-tuple ontology structure of KAON [3] and the Methontology [11] which is the ontology-building methodology. Based on the template, users are supposed to define keyword, informal description, superconcept, verbs (relations), and related terms. In the proposed template, we take only user-friendly expressions in UO and rule out rarely used expressions such as "negation" and "number restriction" in user's informal description. As a result, we define the UO template and guideline instruction as shown in Table 1, which can be provided to users.

4.2 Transformation of UO into IO

The OntSE transforms a UO into an IO to represent user's input in formal representation for computer processing. For the formal representation, we use a FL_0 [1], which is one of the DL expression languages without negation and number restrictions and allows only conjunction and universal

Table 1 The template for users to define a UO

Step	Instruction
(1) Keyword	Insert a keyword
(2) Keyword's informal description	Write your meaning of the keyword
(3) Super concept of the keyword	Insert super concept of the keyword
(4) Related term	Write related terms with which you could explain the keyword
(5) Term–verb–term	Write verbs between terms or between terms and the keyword

quantifiers. The transformation process consists of five steps as follows:

Step 1. Transform a keyword and its super concept into concepts in library.
Step 2. Transform verbs into relations.
Step 3. Set a domain and a range for each relation.

(a) Let a concept (left of a verb) be a domain of a relation.
(b) Let a concept (right of a verb) be a range of a relation.

Step 4. Define DL phrases (FL_0) for the ontology IO of UO.

(a) If a relation's domain is a keyword, then phrase is "Keywords =∀ relation.range."
(b) If a relation's domain is the same with another relation's range, then phrase is "∀ latter relation.[domain of the former relation] ∩ ∀ former relation.[range of the former range]." You can do this recursively.

Step 5. Combine the DL phrases and a superconcept together by conjunction.

5 Ontology mapping

After defining an IO, the OntSE performs ontology mappings between keyword's IO and the archived IOs in the IOL. The procedures of ontology mapping consist of taxonomy update and multidimensional similarity calculation. In the taxonomy update, a user's keyword and existing concepts in taxonomy tree are combined into new taxonomy trees by normalization and classification algorithms [4]. This new taxonomy is used for calculating the degree of taxonomy similarity and guiding next users to define an UO. In the multidimensional similarity calculation, the multidimensional similarity value is calculated to quantitatively evaluate the similarity between the user's keyword and a concept in archived IOs. Its calculation considers the degrees of character similarity, taxonomy similarity, and definition similarity, which will be explained in detail in Section 5.2. If the degree of multidimensional similarity is over a threshold value, the OntSE recommends the concept as a candidate of the matched concept with the keyword. After completing ontology mapping with all concepts in the IOL, the OntSE shows the candidates in order to let a user select the concepts that the user intends to find. For more detailed description, in this study, we use the following notations.

Notations

ChaSim	the degree of character similarity
TaxoSim	the degree of taxonomy similarity
DefSim	the degree of definition similarity
BNSim	the degree of similarity between two different concepts
C_i	concept
$(C_i - C_j)$	a pair of concept C_i and C_j, a node in the BN
C_A	a concept in the IOL
C_B	a user's keyword in IO form
C_S	a selected concept by a user after mapping
IO_A	ontological definition of a concept C_A
IO_B	ontological definition of a keyword concept C_B
IO_S	ontological definition of a selected concept C_S
$[C_{AP}]$	a set of primitive concepts in IO_A
$[C_{BP}]$	a set of primitive concepts in IO_B
$[C_{SP}]$	a set of primitive concepts in IO_S
α	coefficient of the *ChaSim* (C_A, C_B)
β	coefficient of the *TaxoSim* (C_A, C_B)
γ	coefficient of the *DefSim* (C_A, C_B)
P_p	prior probability
η	a set of nodes in BN (general)
η'	a set of nodes in BN (based on the matched result)
d	concept node in η'
A	primitive concept nodes in η'
A'	a set of nodes that have the same primitive concepts, which have the same character strings, in η'
A''	a set of nodes that have the different primitive concepts, which have different character strings, in η'

5.1 Taxonomy update

The taxonomy update is the process of concept classification which builds subsume relationships between concepts.

Through the taxonomy update, users can know where the user keyword can be located in the existing taxonomy, and users can define the description of the keyword properly. The concept classification requires normalization and subsumption. In this study, we adopt the previous normalization process and subsumption algorithm [4]. Normalization generates normalized description of all concepts. After normalization of user's keyword using the normalized concepts in IOL, all concept definitions in IOL should be normalized further if they include the concept which is the same with the normalized keyword. Thus, we should check all definitions of the concepts in the IOL for further normalization. After that, in order to find concept hierarchies in the TOL, we apply the subsumption algorithm [4]. The subsumption algorithm finds where a new concept is located in a taxonomy. It creates some concept hierarchies with a new concept and existing ones in the taxonomy and redirects them in a new taxonomy.

5.2 Multidimensional similarity calculation

After taxonomy updating, the OntSE calculates the degree of similarity between user's keyword and a concept in IOL. We call it the *degree of multidimensional similarity* (noted as *Similarity*) because it is calculated by several aspects such as character similarity, taxonomy similarity, and definition similarity. Some concepts that have the *degree of multidimensional similarity* over a threshold value are recommended to users as candidates for semantically similar or same concepts of user's keyword. Here, the threshold value can be decided in an empirical way.

The *degree of multidimensional similarity* between a concept (C_A) in the IOL and user's keyword (C_B) is defined as follows.

$$\begin{aligned}\mathrm{Similarity}(C_A, C_B) = {} & \alpha \cdot \mathrm{ChaSim}(C_A, C_B) \\ & + \beta \cdot \mathrm{TaxoSim}(C_A, C_B) \\ & + \gamma \cdot \mathrm{DefSim}(C_A, C_B)\end{aligned} \tag{1}$$

where $\alpha + \beta + \gamma = 1$, $\alpha, \beta, \gamma \geq 0$. ChaSim($C_A$, C_B), *TaxoSim*(C_A, C_B), and *DefSim*(C_A, C_B) are functions of the degrees of character similarity, taxonomy similarity, and definition similarity, respectively. The values of α, β, and γ can be decided in an empirical way by users or experts. We recommend that γ should be bigger than α and β since the OntSE puts emphasis on semantic mapping based on concept's ontological definitions.

5.2.1 Character similarity

In this study, to compute the degree of character similarity for two concepts, we use the string matching [23] measure as shown in Eq. 2. The edit distance between two strings had been proposed by Levenshtein [21], which indicates the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.

$$\mathrm{ChaSim}(C_A, C_B) = \max\left(0, \frac{\min(\langle C_A \rangle, \langle C_B \rangle) - \mathrm{ed}(C_A, C_B)}{\min(\langle C_A \rangle, \langle C_B \rangle)}\right) \tag{2}$$

where $\langle C_A \rangle$ (or $\langle C_B \rangle$) is the number of characters in C_A (or C_B) and ed(C_A, C_B) indicates the edit distance [21] between C_A and C_B. The higher the ChaSim for the two concepts' character strings is, the more similar the two concepts are.

5.2.2 Taxonomy similarity

Since two concepts located near each other in a taxonomy tree can have more similar semantics, we calculate the degree of taxonomy similarity based on the taxonomical relationships of the two concepts. The taxonomical relationships between user's keyword and a concept in the IOL can be obtained by the approach of taxonomy update described in Section 5.1. The degree of taxonomy similarity between C_A and C_B, TaxoSim (C_A, C_B), can be gotten from Table 2.

5.2.3 Definition similarity

To compute the degree of definition similarity between two concepts C_A and C_B, we should compare their primitive concepts and relations in their ontological definitions. Note that primitive concepts and relations indicate the concepts and relations used to describe the semantics of the concepts, C_A and C_B, in detail in their ontology definitions. There may be several factors to compare their primitive concepts and relations. In this study, three factors are considered: the number of identical primitive concepts (i.e., having the same character strings) having the same relations, the number of identical primitive concepts having different relations, and *BNSim*

Table 2 Taxonomy similarity definition

TaxoSim(C_A, C_B)	Relationship between C_A and C_B
1	If C_A and C_B are identical
0.8	If C_A and C_B have a direct hierarchical relation
0.7	If C_A and C_B have a sibling relation
0.5	If C_A and C_B have a grandparent relation
0.3	If C_A and C_B have an uncle relation
0.2	If C_A and C_B have an ancestor relation,
0	Otherwise

that represents the degree of similarity between two different primitive concepts (having different character strings) in the definitions of two concepts C_A and C_B, which is generally obtained from BN.

Then, the degree of definition similarity (DefSim) is calculated as the sum of three factors as follows.

$$\mathrm{DefSim}\,(C_A, C_B) = \frac{N_R \times 2}{N_{C_A}+N_{C_B}} + \delta \times \frac{N_R' \times 2}{N_{C_A}+N_{C_B}} + \frac{\mathrm{BNSim}\left(\left[C_{AP}'\right],\left[C_{BP}'\right]\right)\times 2}{N_{C_A}+N_{C_B}}, 0 \leq \delta \leq 1 \tag{3}$$

where $N_{C_A}(N_{C_B})$is the number of primitive concepts in C_A (C_B). N_R (N_R') indicates the number of same primitive concepts of C_A and C_B that have the same (different) relations; $\left[C_{AP}'\right]$ $\left(\left[C_{BP}'\right]\right)$ is a set of primitive concepts in C_A (C_B), which are not the same with primitive concepts in C_B (C_A); and δ is a constant coefficient.

Here, how to calculate a BNSim will be explained in Section 6.3. We set the constant (δ) to less than 1 because the same primitive concepts having different relations can have less effect on DefSim than the same primitive concepts having the same relations.

6 Ontology updating

In this study, we use a BN to consider the effect of matching result and infer the BNSim of the primitive concepts. After user's matching decision which is the selection of some of the candidates recommended by OntSE between two concepts (keyword and existing IO), the two matched concepts and primitive concepts in their definitions are transformed into a BN, which is then merged into the existing BN. As a result, new nodes are added to build new BN, and the probabilities of nodes (similarities of two concepts) are updated with the matching results (new evidences).

6.1 Network construction

The OntSE builds a BN that consists of nodes and arcs with a keyword and the selected concept and their primitive concepts. The network is then merged into the existing BN. In the BN that we consider here, a keyword and the selected concept are formed into the node of concept pair (shortly *concept node*), and the pair of their primitive concepts are formed into the node of primitive concept pair (shortly *primitive concept node*). Thus, the nodes have two types: concept nodes and primitive concept nodes. Again, the *concept node* represents the pair of (user's keyword—selected concept), and *primitive concept node* represents the following pair: (*one of primitive concepts of a selected concept–one of primitive concepts of user's keyword*). A *primitive concept node* can be further classified into two types: one for the pair of the same primitive concepts and the other for the pair of different primitive ones. We can easily guess that the pairs of the same primitive concepts affect the degree of similarity of their concepts. However, we do not know which pairs of different primitive concepts affect the degree of similarity on their *concept node*. In order to deal with it, we define the possible *primitive concept nodes* generated by the combination of all different primitive concepts of a keyword and a selected concept. A set of nodes in BN (η') for the concept C_S (selected concept) and C_B (user's keyword) is defined as follows: $\eta' = \{d\} \cup A$ where $d = (C_S - C_B)$, $A = A' \cup A''$,

$$\begin{aligned} A' &= \left\{\left(C_k - C_k^*\right)\middle| C_k = C_k^*, C_k \in [C_{SP}], C_k^* \in [C_{BP}],\ \text{for } 0 \leq k \leq \min(|[C_{SP}]|, |[C_{BP}]|)\right\}, \\ A'' &= \left\{\left(C_s - C_t^*\right)\middle| C_s \neq C_t^*, C_s \in [C_{SP}], C_t^* \in [C_{BP}],\ \text{for } 0 \leq s \leq |[C_{SP}]|, 0 \leq t \leq |[C_{BP}]|\right\} \end{aligned} \tag{4}$$

Here, $|\bullet|$ indicates the number of elements in $\bullet$. d, A', and A'' indicate the *concept node*, *primitive concept node* with the same primitive concepts, and the *primitive concept node* with the different primitive concepts, respectively. The arcs from *primitive concept nodes* to a *concept node* can be defined considering the relations between a concept and its primitive concepts.

Before merging a transformed network with the existing BN, we need to calculate the initial probabilities for the nodes of the transformed network. For this purpose, we define the prior probability (P_P) of a *primitive concept node* $\left(C_s - C_t^*\right)$ as follows.

$$P_P\left(\left(C_s - C_t^*\right)\right) = \begin{cases} 0.9 & \text{if} \left(C_s - C_t^*\right) \in A', \\ \dfrac{\mathrm{ChaSim}\left(C_s, C_t^*\right) + \mathrm{Similarity}(C_S, C_B)}{2} & \text{if} \left(C_s - C_t^*\right) \in A'' \end{cases} \tag{5}$$

such that $C_s \in [C_{SP}]$, $C_t^* \in [C_{BP}]$, and $(C_S - C_B)$ is a *concept node*.

Here, we can assume that the prior probability of a *primitive concept node* in A' is 0.9 because two concepts in the *primitive concept node* are the same. We also assume that the prior probability of a *primitive concept node* in A''

is an average of the degree of character similarity (ChaSim) between two primitive concepts in the *primitive concept node* and the degree of multidimensional similarity (Similarity) between two concepts in the *concept node*. Since the prior probability of a *primitive concept node* in A'' is mainly affected by its character similarity and its parent node (*concept node*)'s similarity, this assumption is reasonable. On the other hand, these prior probabilities can be defined with the probabilities of the same nodes in the existing BN.

The CPT of a *concept node* has various probabilities conditioned by its linked *primitive concept nodes* because the *primitive concept nodes* can have True/False cases. The conditional probability $P(d|\theta^m)$ of a *concept node* d, (C_S-C_B), for a condition θ^m is calculated as follows.

$$P(d|\theta^m) = \frac{N_{A'|\theta^m} + \left|T_{[C_{DP}]|\theta^m}\right|}{|[C_{\mathrm{MIN}}]|} \tag{6}$$

where $[C_{DP}]$ is a set of primitive concepts that are elements of $\{[C_{MIN}]-([C_{BP}] \cap [C_{SP}])\}$. $[C_{MIN}]$ is a set of primitive concepts such that $[C_{MIN}]=[C_{BP}]$ if $|[C_{BP}]|\leq|[C_{SP}]|$ and $[C_{MIN}]=[C_{SP}]$ if $|[C_{BP}]|>|[C_{SP}]|$. θ^m is a condition included in the set of conditions for $d(\chi_d)$ such that $\theta^m \in \chi_d$, $1 \leq m \leq 2^{|A|}$. It indicates there is one among all the cases where *primitive concept nodes* of d have true or false. $N_{A'|\theta^m}$ is the number of true cases for A' under a condition θ^m. $T_{[C_{DP}]|\theta^m}$ is a set of primitive concepts which belong to $[C_{DP}]$, included in the nodes which are known as true under a condition θ^m.

6.2 Bayesian network update

When a new BN is merged with the existing BN which might include isolated subnetworks, the structures of the existing BN and the probabilities of nodes in the BN are updated. Here, we have to consider the following four merging cases since the probability of a Bayesian node is differently calculated depending on the cases. The merging is performed node by node so that the cases are defined for node merging, which finally results in BN merging.

Case 1. If there is no identical node between new BN and the previous BN, we make the updated BN by adding the new BN as the isolated network. In this case, there is no link between the nodes of the new BN and the ones of the previous BN.

Case 2. For a *primitive concept node* in the new BN, if the same *concept node* exists in the previous BN, we merge two nodes into one node by linking the arcs attached to *primitive concept node* in the new BN to the *concept node* of the existing BN. In this case, the probability of the merged node is given with the greater value among the probability values of two nodes, and then the CPTs of *concept nodes* are changed accordingly.

Case 3. For a *primitive concept node* in the new BN, if the same *primitive concept node* exists in the previous BN, then two nodes are merged into one *primitive concept node* in the updated BN by linking the arcs attached to the *primitive concept node* in the new BN to the *primitive concept node* of the existing BN. In this case, the probability of the merged node is given with the greater value among the probability values of two nodes, and then the CPTs of *concept nodes* are changed accordingly.

Case 4. For a *concept node* in the new BN, if the same *primitive concept node* exists in the previous BN, then two nodes merge into one node by linking the arcs attached to the *concept node* in the new BN to the *primitive concept node* of the existing BN. In this case, the probability of the merged node is given with the greater value among the probability values of two nodes, and then the CPTs of *concept nodes* are changed accordingly.

6.3 Probability calculation of *BNSim*

The OntSE sets the probability of the newly added *concept node* in a BN as 1 because a user has decided that their concepts are matched and considered a new evidence. The changed probability is then propagated through the BN model, and as a result, the probabilities of some of the nodes related to the new *concept node* are affected. This can be explained by the propagation of the BN that calculates the posterior probability by the Bayes rule. In this study, we omit this calculation process because the calculation can be easily obtained by several available tools (e.g. Microsoft BN Editor [24]). The updated probabilities are saved in the MOL, and they are used to calculate the BNSim for the next mapping.

As shown below in Eq. 7, we can calculate the BNSim between a selected concept, C_S, and a keyword, C_B, by considering $\left[C'_{SP}\right]$ and $\left[C'_{BP}\right]$.

$$\mathrm{BNSim}\left(\left[C'_{SP}\right],\left[C'_{BP}\right]\right) = \sum_{i=1}^{m}\sum_{j=1}^{k} p_{ij}/N, \tag{7}$$

where $\left[C'_{SP}\right]$ $\left(\left[C'_{BP}\right]\right)$ is a set of primitive concepts in C_S (C_B), which are not the same with primitive concepts in C_B (C_S); p_{ij} denotes the probability of *primitive concept node*, $(C_i - C_j^*)$, such that $C_i \neq C_j^*$, $C_i \in \{\left[C'_{SP}\right]\}$, and $C_j^* \in \{\left[C'_{BP}\right]\}$ for $1\leq i\leq m$ and $1\leq j\leq k$, $m = \left|\left[C'_{SP}\right]\right|$, $k = \left|\left[C'_{BP}\right]\right|$. N is the number of p_{ij} satisfying the above conditions. Here, if there is no $(C_i - C_j^*)$ in the BN, let p_{ij}=0.5.

7 Case study

In this section, we perform a case study that shows the benefit of the proposed OntSE approach. The case study is confined within a small domain about digital products, of which ontology has been stored into the OntSE in the form of domain ontology (IOL), a taxonomy (TOL), and a BN with matching results (MOL), as shown in Table 3 and Figs. 2 and 3, respectively. Figure 3 shows the probabilities of the nodes in the existing BN.

The main scenario of the case study is as follows: A user inputs a keyword, "HighCellPhone," with its description. Then, the OntSE helps the user to find the semantically same concepts of the keyword from the ontology repository through the procedure of the modules as we explained in the previous sections. After that, the OntSE updates the ontology repository with user's keyword and matching result. The more detailed explanations with examples are as follows.

7.1 Ontology building

7.1.1 Defining UO

With the help of the template of Table 1, the user defines the UO of the keyword, "HighCellPhone," and other necessary descriptions as shown in the left table of Fig. 4.

Table 3 Sample ontology repository

ID	Concept name	Definition
1	CellPhone	Telephone ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Mobile
2	WirelessPhone	Telephone ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Portable
3	InternetPhone	Telephone ∩ ∀ communication.Internet
4	MobileInternetPhone	InternetPhone ∩ ∀ HasAttribute.Mobile
5	BraunTubeTelevision	Television ∩ ∀ HasDisplay.BraunTube
6	LCDTelevision	Television ∩ ∀ HasDisplay.LCD
7	ProjectionTelevision	Television ∩ ∀ HasDisplay.Screen
8	DVDPlayer	MediaPlayer ∩ ∀ HasMedia.DVD
9	VideoPlayer	MediaPlayer ∩ ∀ HasMedia.VHS
10	VideoGamePlayer	GamePlayer ∩ ∀ HasMedia.DVD
11	MobileComputer	Computer ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Mobile
12	PDA	MobileComputer ∩ ∀ HasCharacter.Handy
13	SmartPhone	Telephone ∩ Computer ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Movable
14	VideoPhone	CellPhone ∩ ∀ Transfer.Information ∩ ∀ hasContents.Video
15	PortableTelevision	BraunTubeTelevision ∩ ∀ HasCharacter.Handy ∩ ∀ broadcasting.Wireless
16	DMB	LCDTelevision ∩ ∀ HasCharacter.Handy ∩ ∀ broadcasting.Wireless
17	PortableDVDPlayer	DVDPlayer ∩ ∀ HasCharacter.Handy ∩ ∀ HasDisplay.LCD
18	PortableMediaPlayer	MediaPlayer ∩ ∀ HasCharacter.Handy ∩ ∀ HasDisplay.LCD
19	NextGenerations VideoGamePlayer	VideoGamePlayer ∩ ∀ communication.Internet
20	PortableVideoGamePlayer	VideoGamePlayer ∩ ∀ HasCharacter.Handy ∩ ∀ HasDisplay.LCD ∩ ∀ communication.Internet
21	ComContent	DigitalContent ∩ ∀ transferredBy.CommunicationDevice
22	AudioContent	ComContent ∩ ∀ hasCharacteristics.Voice
23	VideoContent	ComContent ∩ ∀ hasCharacteristics.Video
24	ImageContent	ComContent ∩ ∀ hasCharacteristics.Image
25	TextContent	ComContent ∩ ∀ hasCharacteristics.Text
26	MobileAudioContents	AudioContent ∩ transferredBy.MobileDevice
27	MobileVideoContents	VideoContent ∩ transferredBy.MobileDevice
28	MobileImageContents	ImageContent ∩ transferredBy.MobileDevice
29	MobileTextContents	TextContent ∩ ∀ transferredBy.CommunicationDevice
30	3GCellPhone	CellPhone ∩ ∀ has.(TransmissionSpeed ∩ ∀ HasValue.Fast)
31	4GCellPhone	3GCellPhone ∩ ∀ transmit.(HDVideoContent ∩ 3DVideoContent)
32 (Query)	HighCellPhone	Cell phone which has high transmissionSpeed and transmits VideoContent.CellPhone ∩ ∀ has.(TransmissionSpeed ∩ ∀ HasValue.Fast) ∩ ∀ transmit.VideoContent

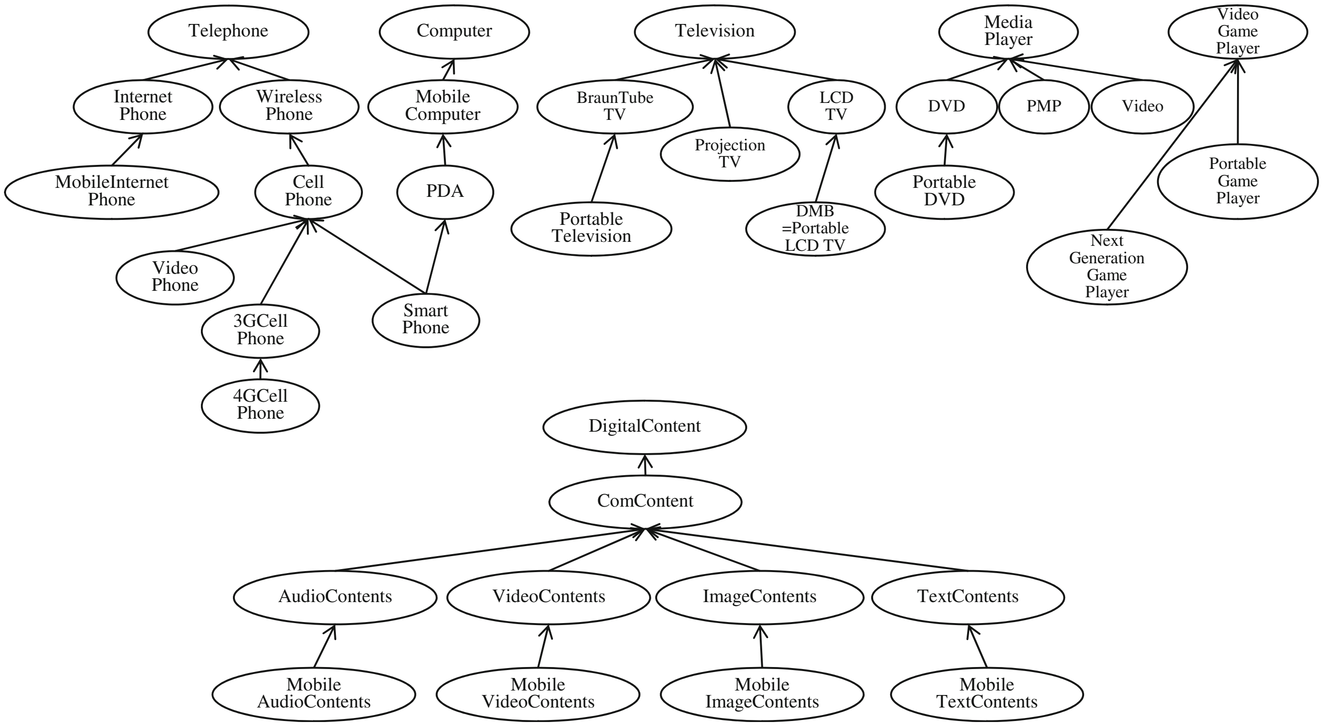


Fig. 2 Taxonomy in the TOL

7.1.2 Transformation of UO into IO

Figure 4 shows an example of a UO, "HighCellPhone," and a transformed IO. After transformation, concepts, relations, a concept hierarchy, relation's domain and range, and primitive concepts are formulated. Both a UO and an IO are stored as new ones in the UOL and the IOL.

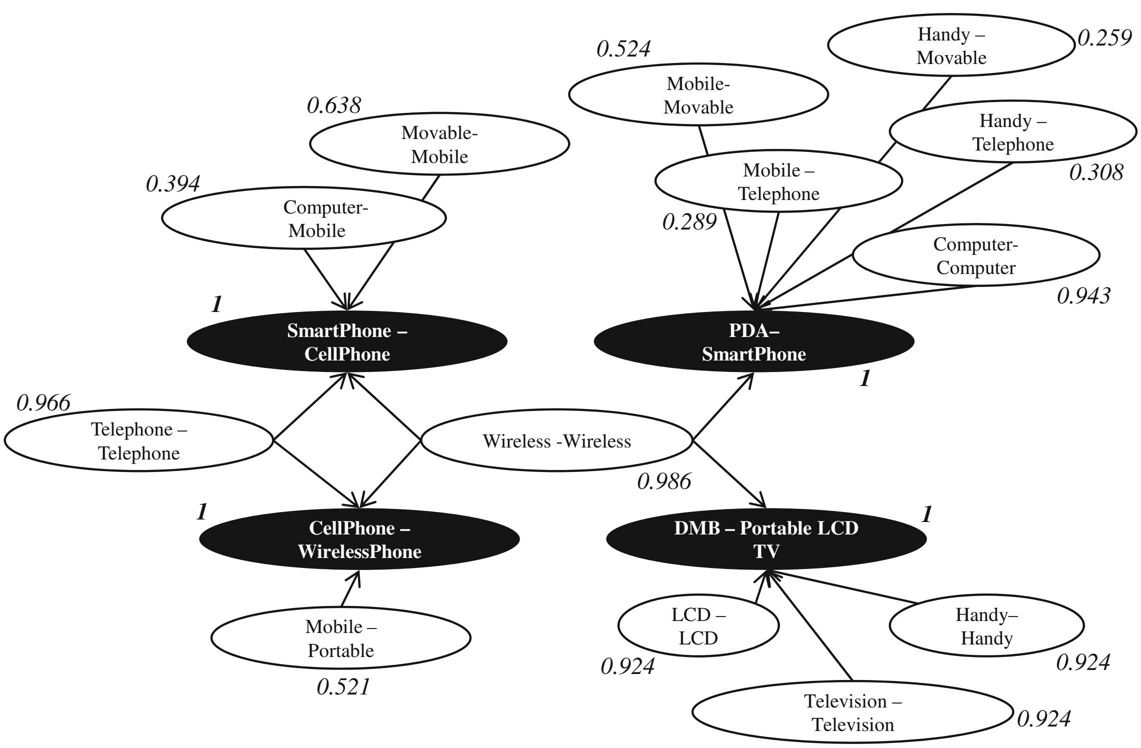


Fig. 3 BN in the MOL

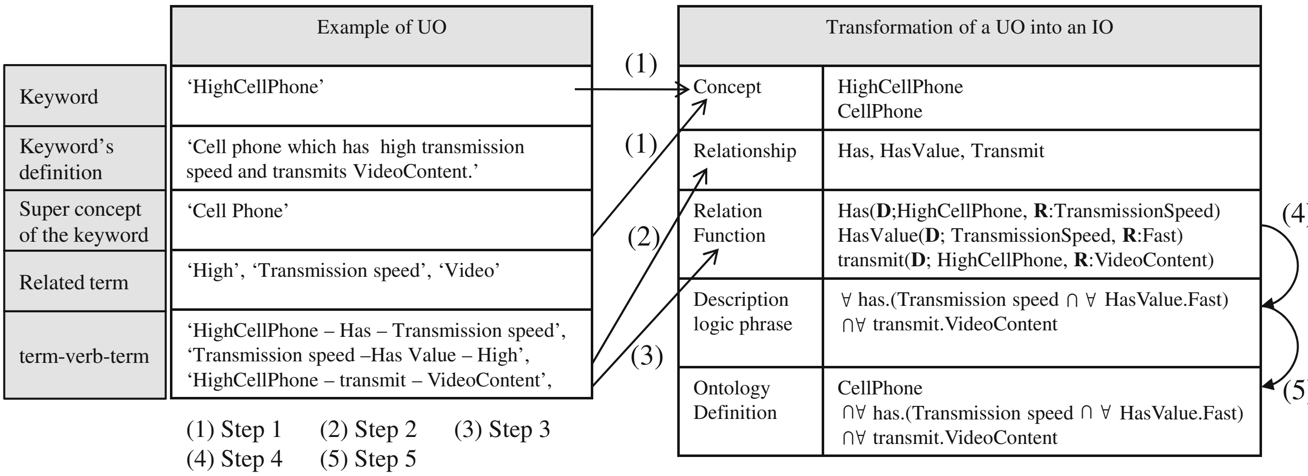


Fig. 4 Transformation of a UO into an IO

7.2 Ontology mapping

7.2.1 Taxonomy update

Normalization and classification Figure 5 (a) shows the result of normalization of the keyword, "HighCellPhone." During this normalization, the "CellPhone" in the keyword's subconcepts (primitive concepts) is replaced with the definition of "CellPhone" in IOL. The procedure for the normalization of the "4GCellPhone" as one candidate concept for mapping is shown in Fig. 5 (b). In the taxonomy update of the "HighCellPhone," it is not the case that taxonomy reasoning can be automatically done with the existing taxonomy. Thus, in this case, additional information provided by the user is required for the taxonomy update. If a user provides the information that the concept "Video content" includes "3DVideo content" and "HDVideo content," since the definition of "4GCellPhone" is a subset of the definition of "HighCellPhone," the latter concept can be defined as a superconcept of the former concept by the subsumption algorithm [4]. As a result, TOL will be updated with inserting the normalized keyword "HighCellPhone" as shown in Fig. 6.

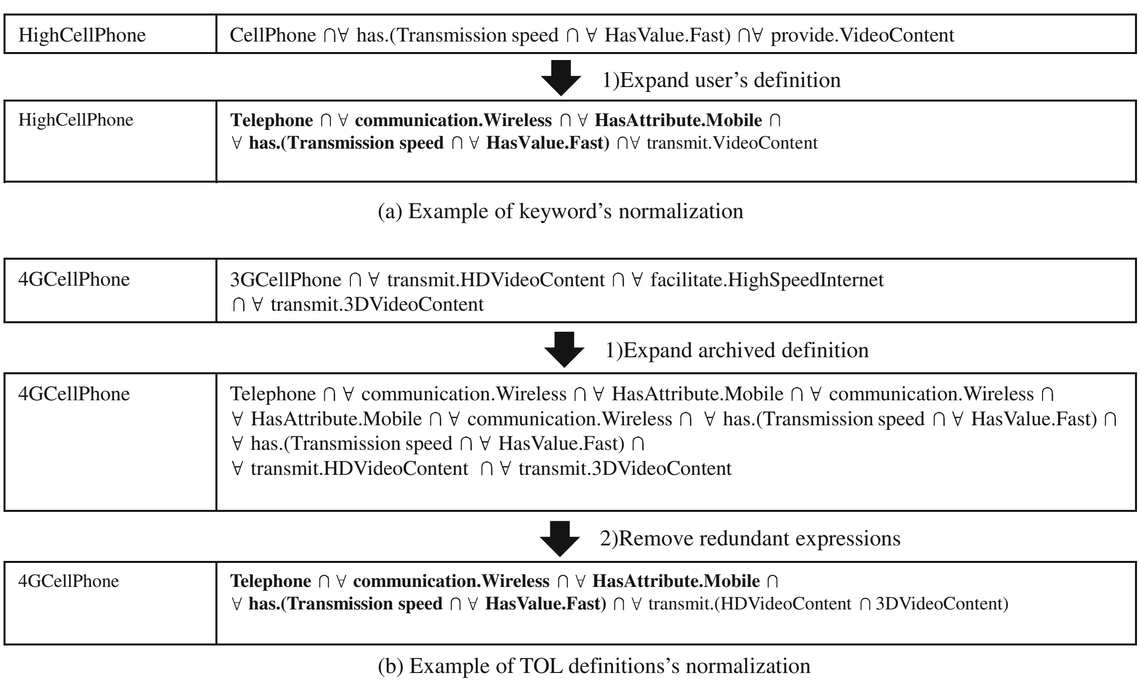


Fig. 5 Example of normalization

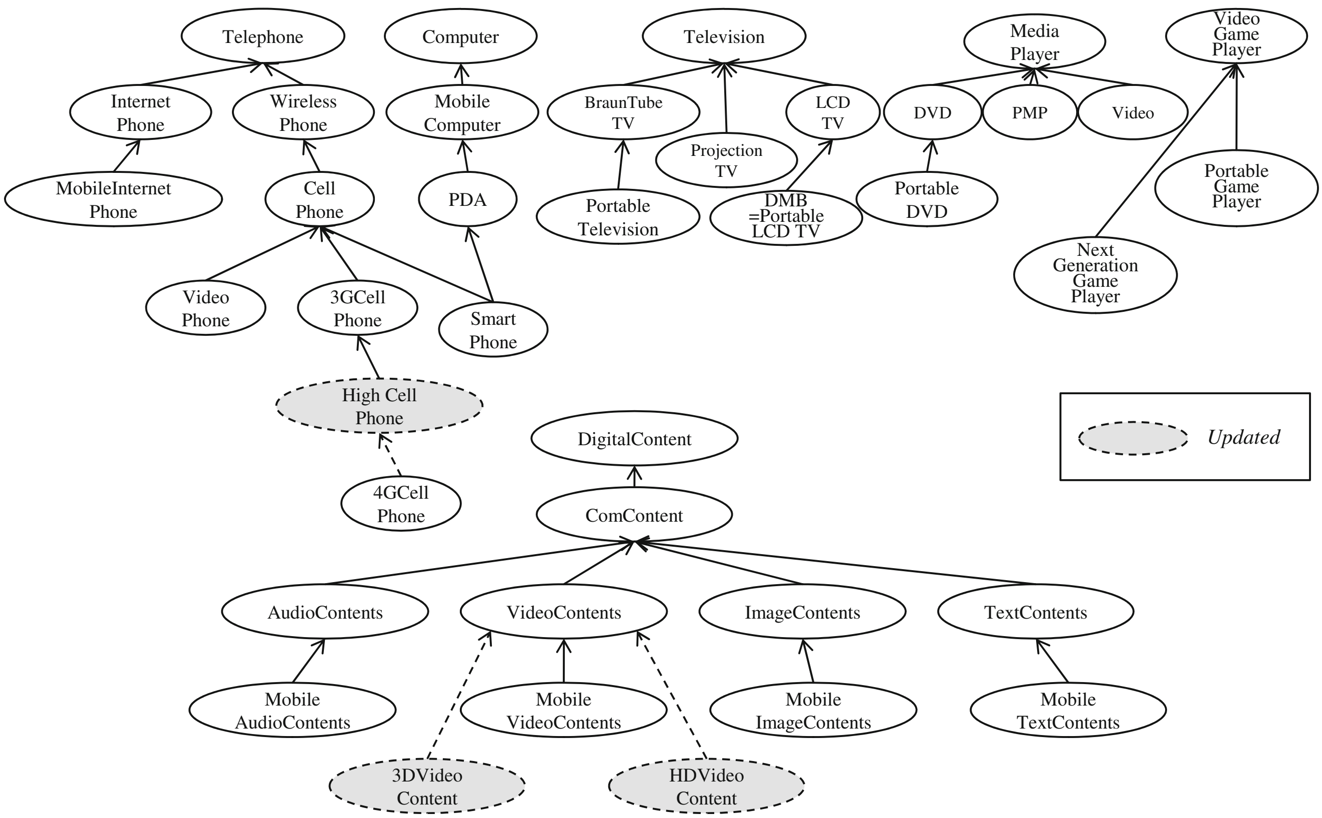


Fig. 6 Taxonomies after updating

7.2.2 Multidimensional similarity calculation

(1) Character similarity: we can calculate the degree of character similarity between C_A="4GCellPhone" and C_B="HighCellPhone" by referring Eq. 2 as follows. Here, since $\langle C_A \rangle$ indicates the number of characters "4GCellPhone," $\langle C_A \rangle = 11$. In the same way, we can get $\langle C_B \rangle = 13$. The edit distance between them is 4. Then, by Eq. 2, we can get the following character similarity value.

$$\mathrm{ChaSim}(C_A, \mathrm{C}_B) = \max(0, \frac{\min(11,13) - 4}{\min(11,13)}) = 0.63$$

(2) Taxonomy similarity: the degree of taxonomy similarity between "4GCellPhone" and "HighCellPhone" is 0.8 according to Table 2 because two concepts have a direct hierarchical relationship as shown in Fig. 6 under the condition that the concept "Video content" includes "3DVideo content" and "HDVideo content." As another example, the degree of taxonomy similarity between "PDA" and "Cell Phone" is 0 because they have no taxonomical relationship.

(3) Definition similarity: to calculate the degree of definition similarity between "4GCellPhone" and "HighCellPhone," their ontological definitions should be considered. The followings show their ontological definitions.

4GCellPhone	Telephone ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Mobile ∩ ∀ has.(TransmissionSpeed ∩ ∀ HasValue.Fast) ∩ ∀ transmit.(HDVideoContent ∩ 3DVideoContent)
HighCellPhone	Telephone ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Mobile ∩ ∀ has.(TransmissionSpeed ∩ ∀ HasValue.Fast) ∩ ∀ transmit.VideoContent

From these definitions, we know N_R=5, because each concept has five common primitive concepts: Telephone, Wireless, Mobile, TransmissionSpeed, and Fast. We also know that $N_R^{'} = 0$ because there is no similar primitive concept that has different relation. To calculate the degree of definition similarity, we need the degree of BN similarity, BNSim, which has been explained in Section 6.3. In this study, we assume that we can get $\mathrm{BNSim}\left(\left[C_{AP}^{'}\right], \left[C_{BP}^{'}\right]\right) = 0.5$. Note that here $\left[C_{AP}^{'}\right] = \{$HDVideoContent, 3DVideoContent} and $\left[C'_{BP}\right] = \{\text{VideoContent}\}$. We also assume that δ=0.9. Then, we have the degree of definition similarity (refer to Eq. 3) calculated as follows:

$$\text{DefSim}(C_A, C_B) = \frac{2\times 5}{7+6} + 0.9 \times \frac{2\times 0}{7+6} + \frac{0.5\times 2}{7+6} = 0.68$$

Based on the above results, if we assume that α, β, and γ in Eq. 1 are 0.3, 0.3, and 0.4, respectively, the degree of multidimensional similarity between C_A and C_B is calculated by Eq. 1 as follows,

$$\text{Similarity}(C_A, C_B) = 0.3 \times \frac{11-4}{11} + 0.3\cdot 0.8 + 0.4\cdot 0.68 = 0.70$$

In this study, let the threshold value of multidimensional similarity be 0.65. Then, since the Similarity (C_A, C_B) is over the threshold value, the OntSE proposes the concept "4GCellPhone" as a matching candidate for the keyword, "HighCellPhone." After trying ontology mapping for all concepts in the IOL, the OntSE shows all candidate concepts to the user, with ranked similarity values. The user then decides the best matched concept of the keyword among candidates. Here, we assume that the user selects "4GCellPhone" as the best matched concept of the keyword "HighCellPhone."

7.3 Ontology updating

After the user decides the matched concept of the keyword, the user selection result should be considered into the existing BN in order to update the ontology for information searching of future users. The following subsections explain the ontology update based on BN.

7.3.1 Network construction

The two matched concepts, "4GCellPhone" and "HighCellPhone," and their primitive concepts are transformed into a network form which has seven nodes as shown in Fig. 7. With our notations, it can be described as follows:

C_S 4GCellPhone
C_B HighCellPhone
$[C_{SP}]$ {Telephone, Wireless, Mobile, Fast, HDVideoContent, 3DVideoContent, TransmissionSpeed}
$[C_{BP}]$ {Telephone, Wireless, Mobile, Fast, VideoContent, TransmissionSpeed }
d (4GCellPhone– HighCellPhone)
A' {(Telephone–Telephone), (Wireless–Wireless), (Mobile–Mobile), (Fast–Fast), (TransmissionSpeed–TransmissionSpeed)}
A'' {(HDVideoContent–VideoContent), (3DVideoContent–VideoContent)}

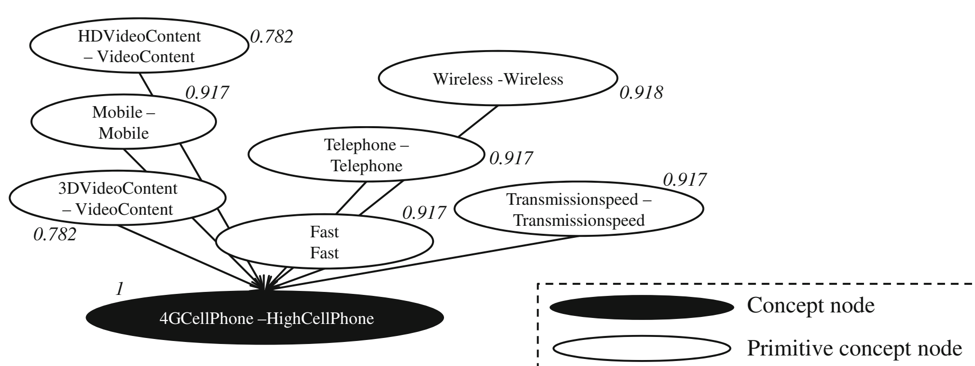
The nodes in the transformed network (Fig. 7) indicate the pairs of two concepts and two primitive concepts. The black-shaded node indicates the *concept node* (pair of two concepts), and the other seven nodes indicate the *primitive concept nodes* (pairs of two primitive concepts). Each *primitive concept node* has prior probability and the *concept node* has a CPT as explained in Section 6. Figure 7 shows the probabilities of the nodes. For example, the prior probability of "(HDVideoContent−VideoContent)" node is calculated by the average of ChaSim and Similarity (refer to Eq. 5) as follows.

$$\begin{aligned}&\text{ChaSim}(\text{HDVideoContent} - \text{VideoContent})\\ &\quad = \text{Max}\left(0, \tfrac{\text{Min}(14,12) - \text{ed}(\text{HDVideoContent},\text{VideoContent})}{\text{Min}(14,12)}\right)\\ &\quad = \text{Max}\left(0, \tfrac{12-2}{12}\right) = 0.83\end{aligned}$$

$$\begin{aligned}&\text{Similarity}(\text{4GCellPhone} - \text{HighCellPhone})\\ &\quad = 0.3\cdot\tfrac{11-4}{11} + 0.3\cdot 0.8 + 0.4\left[\tfrac{2\cdot 5}{7+6} + 0.9\cdot\tfrac{2\cdot 0}{7+6} + \tfrac{2\cdot 0.5}{7+6}\right]\\ &\quad = 0.70\\ &P_P(\text{HDVideoContent} - \text{VideoContent}) = (0.83 + 0.70)/2\\ &\quad = 0.765\end{aligned}$$

Likewise, P_P(3DVideoContent−VideoContent) has 0.765, and the probabilities of "(Telephone−Telephone)" and "(Wireless−Wireless)" are defined with the probabili-

Fig. 7 An example of a transformed network from definitions



ties of the same nodes in the existing BN. The CPT of a *concept node* has the probabilities conditioned by its linked *primitive concept nodes*. For example, a *concept node* “(4GCellPhone-HighCellPhone)” is linked with seven *primitive concept nodes* as shown in Fig. 7. So, the CPT of the *concept node* has 2^7 (i.e., 128) conditional probabilities. If all *primitive concept nodes* are set as true cases, it indicates the condition θ^1, and if all *primitive concept nodes* are set as false cases, it indicates the condition θ^{128}.

The following shows an example for how to calculate the conditional probability of the CPT under a condition θ^4 (refer to Eq. 6):

(Telephone–Telephone)=True, (Fast–Fast)=True,
(HDVideoContent–VideoContent)=False,
(Wireless–Wireless)=True,
(Mobile–Mobile)=True,
(3DVideoContent–VideoContent)=False,
(TransmissionSpeed–TransmissionSpeed) = True

$$[C_{MIN}] = [C_{BP}] = \{\text{Telephone}, \text{Wireless}, \text{Mobile}, \text{Fast}, \text{VideoContent}, \text{TransmissionSpeed}\},$$
$$|[C_{MIN}]| = 6$$
$$[C_{BP}] \cap [C_{SP}] = \{\text{Telephone}, \text{Wireless}, \text{Mobile}, \text{Fast}, \text{TransmissionSpeed}\}$$
$$[C_{DP}] = \{\text{VideoContent}\}$$
$$N_{A'}|_{\theta^4} = 5$$
$$T_{[C_{DP}]}|_{\theta^4} = \{\quad\}$$
$$\left|T_{[C_{DP}]}\middle|\theta^4\right| = 0$$
$$P\left((\text{4GCellPhone} - \text{HighCellPhone})\middle|\theta^4\right) = \frac{5}{6} = 0.83$$

After building a BN with the matched concept as explained the above, the OntSE merges it into the existing BN and updates the probabilities of Bayesian nodes (similarities between two concepts) with the mapping result. This procedure includes (1) BN update and (2) probability calculation for BNSim.

7.3.2 Bayesian network update

Figure 8 shows the updated BN. The dotted-lined ellipses represent newly added nodes, and the solid-lined ellipses show the nodes in the existing BN. Since the “(Telephone−Telephone)” and “(Wireless−Wireless)” *primitive concept nodes* were already in the existing BN as shown in Fig. 3, two *primitive concept nodes* are merged into one for each case as shown in Fig. 8 (case 3 in Section 6.2). With the newly structured BN and user selection result, the probabilities of nodes are updated by the propagation of BN. As the result of user selection, the matched *concept node* can be regarded as an evidence in the BN. Thus, the probability of the *concept node* “(4GCellPhone−HighCellPhone)” is changed as 1 as shown in Fig. 8, which affects the probabilities of other *primitive concept nodes* that are directly connected to the *concept node* in the BN. For example, the probabilities of all *primitive concept nodes* of the “(4GCellPhone−HighCellPhone)” *concept node* slightly increase compared to the cases of Fig. 7. The interesting observation is that the probabilities of “(Telephone−Telephone)” and “(Wireless−Wireless)” slightly increase from the previous values, obtained from the existing BN as shown in Fig. 3. It is caused by the fact that two nodes shared *primitive concept nodes* between existing *concept nodes* and newly added *concept node*. Along with this, we can see that the probabilities of some *primitive concept nodes* such as “(Computer−Mobile)” and “(Mobile−Portable)” are slightly decreased according to the property of BN.

7.3.3 Probability calculation of BNSim

The reason that we update the BN based on user selection is to provide the next user with more appropriate values of similarities of concepts. It is closely related to recalculating the BNSim. For example, let us assume that the next user inputs a keyword “Visual Communication CellPhone.” Its IO is as follows.

VisualCommunicationCellPhone : =Telephone ∩ ∀ communication.Wireless ∩ ∀ HasAttribute.Mobile ∩ ∀ has.(TransmissionSpeed ∩ ∀ HasValue.Fast) ∩ ∀ transmit.(HDVideoContent ∩ MessageContent)

When the OntSE calculates the degree of similarity between a “HighCellPhone” and the new keyword, the updated mapping probabilities of BN are used for the BNSim. The new keyword’s ontological definition includes a “HDVideoContent” concept, and a “HighCellPhone”

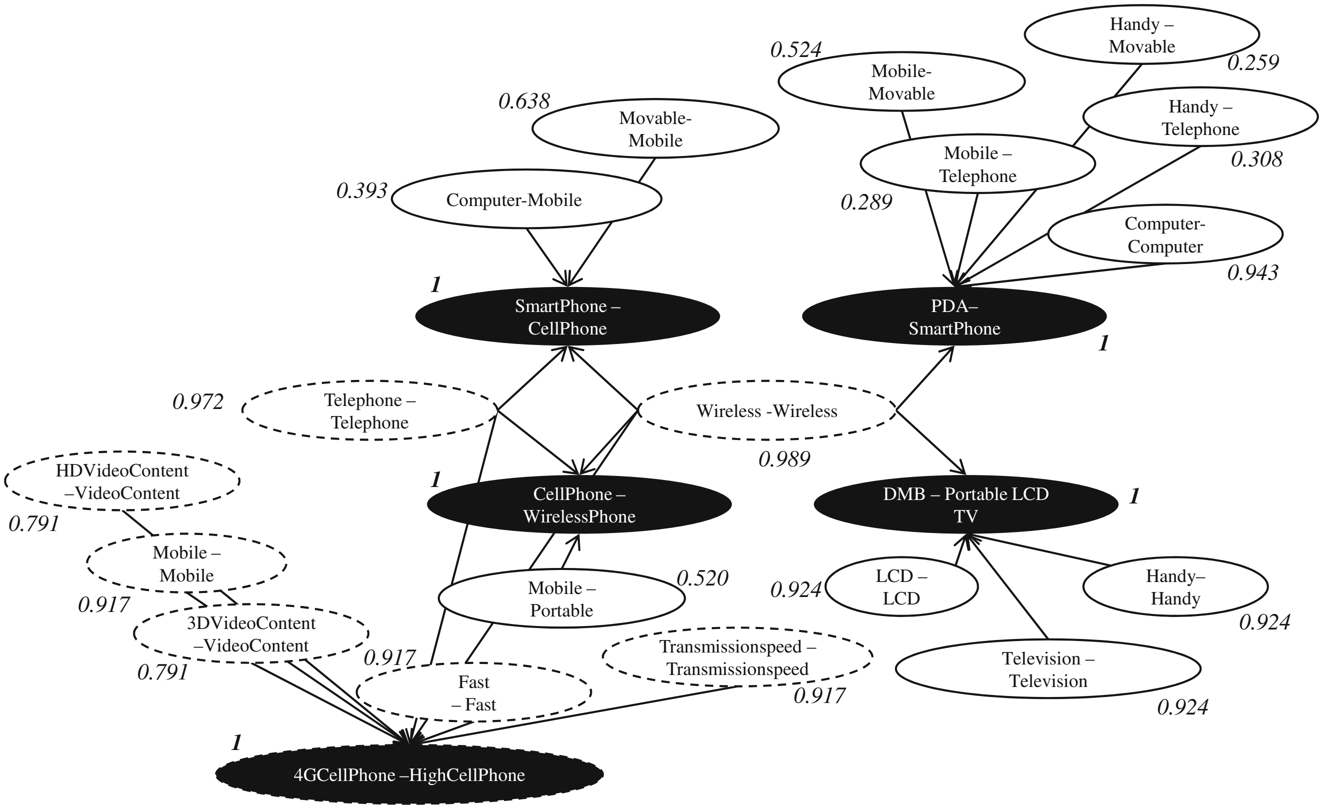


Fig. 8 An example of updated BN

ontological definition includes "VideoContent" concept. Since we know the probability of "(HDVideoContent–VideoContent)" in the updated BN, the *BNSim* can be calculated as follows:

$$
\begin{aligned}
&C_S : \text{HighCellPhone}, \\
&C_B : \text{VisualCommunicationCellPhone} \\
&\left[C'_{SP}\right] = \{\text{VideoContent}\}, \\
&\left[C'_{SP}\right] = \{\text{HDVideoContent}, \text{MessageContent}\} \\
&\left(C_i - C_j^*\right) = (\text{VideoContent} - \text{HDVideoContent}), \\
&\qquad\qquad\quad (\text{VideoContent} - \text{MessageContent}) \\
&BNSim\left(\left[C'_{SP}\right],\ \left[C'_{SP}\right]\right) = (0.781 + 0.5)/2 = 0.64.
\end{aligned}
$$

Here, we need to put the value of p_{ij} (VideoContent–HDVideoContent) as 0.781 instead of 0.765, the value used for previous matching because the value is updated by user selection result.

8 Concluding remarks

8.1 Results

In this study, we have proposed the OntSE, which is especially for a user to search information within documents in various enterprise information systems. With the keyword description input by a user, the proposed approach finds semantically same concepts, and, with the collaboration with the user, some selected concepts can be used for searching relevant documents. The proposed approach has three distinguished features. First, it is based on an ontology mapping to find concepts (terms) which are semantically similar with user's intention. Second, the OntSE updates ontology repository with the descriptions of user's keywords. The updated taxonomy and accumulated ontology can be used for later users while they input the new UO. Third, the user's historical matching decision is utilized to help a later user's search. A BN is built and updated to calculate the probabilities of matched concepts for this. As shown in Section 7, a case study has been conducted to demonstrate the procedures and features of this methodology. Results from our case study show that OntSE provides an effective way to help users in information searching.

8.2 Limitations

In spite of the usefulness of the OntSE, it has also several limitations to be further considered as future research issues. First, we need to have computational experiments to adjust the parameters or weighting factors in the proposed approach for specific domains. They are expected to be different according to the domains. Due to the broad scope of the proposed framework, it is not easy to show the effectiveness of the OntSE according to the domains. Second, in this study, we do not consider the polysemy (the word with several meanings) case in ontology mapping because of its complexity. However, it must be an interesting and fruitful research issue to deal with the polysemy case in ontology mapping, which will enrich the information searching capability. Third, the OntSE does not consider the sentence analysis to extract terms and verbs from the description of user's keyword in an efficient way. Introduction of sentence analysis will be also helpful for users to define a UO. Finally, an OntSE prototype system can be implemented, and, with several scenarios, one can show that semantically same concepts are retrieved through the proposed approach in an easy manner and that documents which contain keywords and the same concepts are effectively searched.

References

1. Baader F, Calvanese D, McGuinness DL, Nardi D, Pater-Schneider PF (2003) The description logic handbook: theory, implementation, and applications. Cambridge University Press, Cambridge, pp 76–78
2. Blok S, Medin D, Osherson D (2002) Probability from similarity. In: Proceedings of AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning, pp 43–50
3. Bozsak E, Ehrig M, Handschuh S, Hotho A, Maedche A, Motik B, Oberle D, Schmitz C, Staab S, Stojanovic L (2002) KAON: towards a large scale semantic Web. Lect Notes Comput Sci 2455:304–313
4. Brachman RJ, Levesque HJ (2004) Knowledge representation and reasoning. Elsevier, Amsterdam
5. Braga RMM, Werner CML, Mattoso M (2000) Using ontologies for domain information retrieval. In: Proceedings of the 11th International Workshop on Database and Expert Systems Applications, pp. 836–840
6. Chiang RHL, Chua CEH, Storey VC (2001) A smart web query method for semantic retrieval of web data. Data Knowl Eng 38:63–84
7. Dey L, Singh S, Rai R, Gupta S (2005) Ontology aided query expansion for retrieving relevant text. Proc Adv Web Intell 3528:126–132
8. Doan A, Madhavan J, Domingos P, Halevy A (2003) Ontology matching: a machine learning approach. Handbook on ontologies in information systems. Springer, Heidelberg, pp 397–416
9. Fang WD, Zhang L, Wang YX, Dong SB (2005) Toward a semantic search engine based on ontologies. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp 1913–1918
10. Fellbaum C (1998) WordNet: an electronic lexical database. MIT Press, Cambridge
11. Gómez-Pérez A, Fernández-López M, Corcho O (2004) Ontological engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web. Springer, Heidelberg, pp 131–137
12. Ha SB, Park YT (2005) Semantic search system using ontology-based inference. Journal of the Korean Institute of Information Scientists and Engineers: Software and applications 32(3):202–214
13. Helsper EM, van der Gaag LC (2002) Building Bayesian networks through ontologies. Proceedings of the 15th European Conference on Artificial Intelligence. IOS Press, Amsterdam, pp 680–684
14. Hyvonen E, Styrman A, Saarela S (2002) Ontology-based image retrieval. In: Proceedings of the XML Finland 2002 Conference, Towards the semantic web and web services, pp 15–27
15. Jensen FV (1996) An introduction to Bayesian networks. Springer, Heidelberg, pp 7–25
16. Jiang S, Huang T, Gao W (2004) An ontology-based approach to retrieve digitized art images. In: Proceedings of the International Conference on Web Intelligence, pp 131–137
17. Kim HH (2005) ONTOWEB: implementing an ontology-based Web retrieval system. J Am Soc Inf Sci Technol 56(11):197–206
18. Kim W, Choi DW, Park S (2005) Product information meta-search framework for electronic commerce through ontology mapping. Proceedings of Semantic Web: Research and Applications 3532:408–422
19. Kim W, Choi DW, Park S (2008) Agent based intelligent search framework for product information using ontology mapping. Journal of Intelligent Information System 30:227–247
20. Kim KM, Hong JH, Cho SB (2007) A semantic Bayesian network approach to retrieving information with intelligent conversational agents. Inf Process Manag 43:225–236
21. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10:707–710
22. Lee M, Jung M, Choi Y, Yang YS, Suh HW (2006) Ontology-based semantic mapping approach using a Bayesian network for communication in CPC environment. In: Proceedings of IJCC Workshop 2006 on Digital Engineering, Korea, pp 92–100
23. Maedche A, Staab S (2002) Measuring similarity between ontologies. Lect Notes Comput Sci 2473(1):15–21
24. Microsoft Co. (2008) Bayesian Network Editor and ToolKit. Available from http://research.microsoft.com/msbn
25. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
26. Mitra P, Noy N, Jaiswal A (2005) OMEN: a probabilistic ontology mapping tool. In: Proceedings of the Semantic Web—ISWC 2005, pp 537–547
27. Mussi S (2006) User profiling on the Web based on deep knowledge and sequential questioning. Expert Syst 23(1):21–38
28. Pan R, Ding Z, Yu Y, Peng Y (2005) A Bayesian network approach to ontology mapping. In: Proceedings of the Semantic Web—ISWC 2005, pp 563–577
29. Sugumaran V, Story VC (2003) A semantic-based approach to component retrieval. The Database for Advances in Information Systems 34(3):8–24
30. Tang J, Li J, Liang B, Huang X, Li Y, Wang K (2006) Using Bayesian decision for ontology mapping. Web Semantics: Science, Services and Agents on the World Wide Web 4:243–262
31. Vallet D, Castells P, Fernandez M, Mylonas P, Avrithis Y (2007) Personalized content retrieval in context using ontological knowledge. IEEE Trans Circuits Syst Video Technol 17(3):336–346
32. Vallet D, Fernndez M, Castells P (2005) An ontology-based information retrieval model. Lect Notes Comput Sci 3532:455–470
33. Varga P, Meszaros T, Dezsenyi C, Dobrowiecki TP (2003) An ontology-based information retrieval system. Lect Notes in Artif Intell 2718:359–368
34. Zhang J, Peng Z, Wang S, Nie H (2006) Si-SEEKER: Ontology-based semantic search over databases. Knowledge Science, Engineering and Management 4092:599–611