Expert Systems with Applications 42 (2015) 913–928

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A reference ontology for profiling scholar's background knowledge in recommender systems

Bahram Amini [a,*], Roliana Ibrahim [a], Mohd Shahizan Othman [a], Mohammad Ali Nematbakhsh [b]

[a] Dept. of Information Systems, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia
[b] Dept. of Computer Engineering, University of Isfahan, Iran

ARTICLE INFO

Article history:
Available online 1 September 2014

Keywords:
Profiling
Recommender system
Ontology
DBpedia
Scholars

ABSTRACT

The profiling of background knowledge is essential in scholar's recommender systems. Existing ontology-based profiling approaches employ a pre-built reference ontology as a backbone structure for representing the scholar's preferences. However, such singular reference ontologies lack sufficient ontological concepts and are unable to represent the hierarchical structure of scholars' knowledge. They rather encompass general-purpose topics of the domain and are inaccurate in representing the scholars' knowledge. This paper proposes a method for integrating of multiple domain taxonomies to build a reference ontology, and exploits this reference ontology for profiling scholars' background knowledge. In our approach, various topics of Computer Science domain from Web taxonomies are selected, transformed by DBpedia, and merged to construct a reference ontology. We demonstrate the effectiveness of our approach by measuring five quality-based metrics as well as application-based evaluation against the developed reference ontology. The empirical results show an improvement over the existing reference ontologies in terms of completeness, richness, and coverage.

1. Introduction

A recommender system for scholars recommends scientific articles to researchers based on their research interests or background knowledge, which is represented in user profiles (Amini, Ibrahim, Othman, & Rastegari, 2011). A user profile encompasses various characteristics of user's context, and represents a precise model of users' preferences, which is interpreted as user's needs. Thus, in scholars' recommender systems, it is essential to capture rich knowledge from scholar's context, and profile what scholars know in line with their research. The effectiveness of recommender systems depends highly on the completeness and accuracy of profiles (Gauch, Speretta, Chandramouli, & Micarelli, 2007). In recent years, Semantic Web technologies (Dolog & Nejdl, 2007), particularly ontology-based approaches, have been proven to be successful for user profiling (Gauch, Speretta, & Pretschner, 2007; Liao, Hsu, Chen, & Chen, 2010), which greatly improve the recommendation process (Uchyigit, 2009). Most profiling approaches exploit a domain ontology and train the scholars' preferences using various contextual information (Liao et al., 2010; Yujie & Licai, 2010). Example of such approaches are (Duong, Uddin, Li, & Jo, 2009; Middleton, Roure, & Shadbolt, 2009; Sieg & Burke, 2007), which employ a pre-built taxonomy such as Open Directory Project (ODP), Yahoo Directory, or WordNet. Those taxonomies serve as reference ontology which formally represent the scholars' preferences (Schiaffino & Amandi, 2009).

However, one possible pitfall in profiling scholars is to use a poor reference ontology, which inadequately capture the scholar's knowledge, and consequently, lead to inaccurate recommendations. The problem with using such pre-built taxonomies is that they are general-purpose directories (Gauch, Speretta, Chandramouli, et al., 2007); the number of topics is insufficient; and the quality of topics is poor, i.e., a significant part of scholar's knowledge could not be mapped to those taxonomies. For example, about 85% of ODP subjects are useless for profiling scholars' knowledge. For illustration, Table 1 represents a small part of ODP hierarchy, showing that many subjects are irrelevant to Computer Science (CS) domain. In addition, many ODP's topics (Mohammed, Duong, & Jo, 2010) have little resemblance with the popular items of scholar's knowledge, because ODP provides many generic terms including Freeware, Browser, Texture, Devices, Client, etc., that are clearly not a research topic in CS domain.

Moreover, such taxonomies do not support "part-of" relationship. In ODP, for instance, the subject "Grid Computing" is neither a sub-part of "Parallel Computing", nor a sub-part of any

* Corresponding author.
E-mail addresses: avbahram2@live.utm.my (B. Amini), roliana@utm.my (R. Ibrahim), shahizan@utm.my (M.S. Othman), nematbakhsh@eng.ui.ac.ir (M.A. Nematbakhsh).

http://dx.doi.org/10.1016/j.eswa.2014.08.031
0957-4174/

Table 1
A sample of ODP subjects, indicating a great number of subjects are useless. Useful entries marked by asterisks.

ODP subject entries	Useful
Computers/Artificial_Life/Particle_Swarm/Conferences	No
Computers/Artificial_Life/Particle_Swarm	Yes*
Computers/Artificial_Life/Particle_Swarm/Papers	No
Computers/Artificial_Life/Particle_Swarm/People	No
Computers/Artificial_Life/People	No
Computers/Artificial_Life/Publications	No
Computers/Artificial_Life/Publications/Journals	No
Computers/Artificial_Life/Publications/Papers	No
Computers/Artificial_Life/Software	Yes*
Computers/Artificial_Life/Research_Groups	No

higher-level topic of CS. Therefore, it is important to develop a reference ontology which captures as many scholars' knowledge items as possible and supports accurate profiling.

In this paper, we develop a reference ontology by bringing together some Web taxonomies in CS domain and create profiles for a group of scholars by real data. The idea of merging several taxonomies takes the advantage of topic aggregation as well as topic cross-validation. The merge process is complex because the concepts of source taxonomies are in different granularity, different structure, ambiguous, and partly incompatible. Table 2 represents a typical heterogeneity among some entries of ODP compared to Wikipedia. As shown, a topic is represented in different syntax: "Particle_Swarm" in ODP versus "Particle Swarm Optimization" in Wikipedia. Additionally, the organization of topics is different: the term "Particle_Swarm" is a sub-topic of two different super-topics, i.e., "Computer/Artificial_Life" and "Artificial Intelligence/Machine Learning/Evolutionary Algorithms". Besides, the source taxonomies are not complete by nature, because they are under updating and the domain knowledge changes over time.

To validate our work, we first empirically measure the reference ontology against the quality factors, including concept coverage, completeness, and richness. Finally, we demonstrate how new reference ontology outperforms the current ontology-based profiling approaches. The rest of the paper is organized as follows: Section 2 overviews the related work. We describe our method in Section 3. Section 4 presents a prototype of our method. Section 5 deals with the evaluation and results. Finally, Section 6 discusses the findings, draws conclusion, and suggests areas of future research.

2. Related work

Ontology is a conceptualization framework of a domain by human- and machine-understandable format containing concepts, relationships, attributes, and axioms (Guarino & Giaretta, 1995). Ontological representation of user profiles enables inferences about the user interests that are widely used in recommender systems (Middleton et al., 2009).

This section reviews the related works from two dimensions: the works that merge domain ontologies regardless of the underlying application, and the profiling approaches that employ a singular ontology as reference ontology.

2.1. Merging ontologies

There are many works for ontology integration, varied in methodological and theoretical frameworks, and tools for automation. They integrate ontologies on the basis of linguistic level (meaning of concepts), instance level, relation or structural level, semantic level, or a combination of them (Chen, Bau, & Yeh, 2011). The similarity of ontological concepts, depending on the target application, is measured from naïve lexical level to semantic level. The automation degree is also varying from manual to high automatic alignment and merging. Selected examples of such approaches are ATOM (Raunich & Rahm, 2011), HCONE-merge (Kotis, Vouros, & Stergiou, 2006), Chimaera (McGuinness, Fikes, Rice, & Wilder, 2000), OBSERVER (Mena, 2000), and ONIONS (Gangemi, Steve, Giacomelli, Medica, & Biomediche, 1999). HCONE-merge employs the reasoning services of description logics (DL) to automatically with varying degree align and merge ontologies. It discovers the intended meaning of ontological concepts by mapping them to WordNet senses using the Latent Semantic Indexing (LSI) approach. These approaches, however, merge ontologies of non-technical domains, do not consider the intended application of the ontology, and do not take into account the role and semantic of concepts in the target ontology.

Fareh, Boussaid, and Chalal (2013) merge OWL ontologies through semantic enrichment of merging concepts by WordNet. This approach enrich ontologies by a set of metadata including annotation over the underlying concepts with synonyms and homonyms via WordNet, or semantic enrichment by human expert, and generating a thesaurus for each ontology to build the global thesaurus. It focuses on computing structural as well as semantic similarity between ontological concepts, and based on a weighted combination of similarity computing methods.

The very similar works to our approach are MeMO and SMART. MeMO (De Araujo, Lopes, & Loscio, 2010) is an automatic clustering-based approach for merging multiple ontologies. It calculates the similarity of ontologies to determine what pairs of ontologies to be merged first, and employs a hierarchical divisive clustering method to merge similar parts of ontologies in a progressive combination of alignments. However, this approach depends on the correspondence of ontology alignments obtained by ontology matching, lacks of supporting big taxonomies, where there are many similar clusters of ontologies to be joined. Also, PROMPT (Noy, Musen, & Informatics, 2000) and formerly called SMART (Noy & Musen, 1999) is a semi-automatic approach which merges

Table 2
A sample of two source taxonomies representing the structural and naming heterogeneity among subjects in ODP and Wikipedia. Bold texts show similar subjects in different wordings.

ODP	DBpedia
Computers/Artificial_Life/**Particle_Swarm**	Artificial Intelligence
Computers/Artificial_Life/Particle_Swarm/Papers	
Computers/Artificial_Life/Particle_Swarm/People	
Computers/Artificial_Life/People	Machine Learning
Computers/Artificial_Life/Publications	Evolutionary Algorithms
Computers/Artificial_Life/Publications/Journals	Genetic Algorithms
Computers/Artificial_Life/Publications/Papers	Ant Colony Optimization
Computers/Artificial_Life/Research_Groups	**Particle Swarm Optimization**
Computers/Artificial_Life/Software	Bees Algorithm

ontologies in a syntactical and semantic levels without any assumption about the structure of participant ontologies.

In addition, Fareh et al. (2013) compare the state-of-the-art of ontology merging approaches, and conclude that most approaches employ syntactic similarity to identify the correspondence among ontological concepts and rarely take into consideration the semantic similarities. Recently, the taxonomy-based similarity (Huang, Milne, Frank, & Witten, 2012) as well as scheme-based similarity (Xinglin, Qilun, Qianli, & Guli, 2012) are employed which measure the similarity upon the structure (i.e., the relationship among concepts) and property restrictions (i.e., class, type, and range) of ontological concepts.

However, the common weaknesses among these methods involve fully matching concepts using syntax similarity, overlooking the target application of the ontology, engaging external knowledge of the domain or extreme human intervention, and missing the global view of the target ontology.

2.2. Profiling approaches

Ontologies in the context of personalized systems typically represent user preferences as a set of weighted concepts, interrelationships with "is-a" or "has-part", and sophisticated features such as logical reasoning. Concepts are usually extracted from pre-existed or reference ontologies, which are general purpose taxonomies (Gauch, Speretta, Chandramouli, et al., 2007). Example of successful approaches are (Duong et al., 2009; Middleton et al., 2009; Sieg & Burke, 2007), which employ a pre-built taxonomy such as Open Directory Project (ODP), Yahoo Directory, or WordNet. Those taxonomies serve as a reference ontology which formally represents the scholars' preferences (Schiaffino & Amandi, 2009).

The work presented in Mohammed et al. (2010) engages ODP as a reference ontology to model users' preferences by utilizing the terms of user's queries. It employs three levels of ODP's hierarchy with a minimum of five indexed documents associated with each subject. This approach focuses only on a limited scope of user keywords and does not consider more sophisticated scholar's knowledge such as publications and reading articles.

Sieg, Mobasher, and Burke (2007b) propose ODP as a reference ontology and engage ODP's topics and associated Web pages as training data for learning the users' interest. Similarly, in Sieg, Mobasher, and Burke (2010), ontological concepts are collected from Amazon's Book Taxonomy. It contains the book's title, book's categories, URL, and editorial review. In Trajkova and Gauch (2004), the top three levels of ODP hierarchy have been employed as a reference ontology for profiling scholars' interests. These works support neither "part-of" relation nor hierarchical structure of scholars' interests.

In Liao, Kao, Liao, and Chen (2009), an ontology-based recommender system for Chinese digital libraries is proposed. It uses a traditional cataloging scheme as a reference ontology which is known as the Library of Congress Classification (LCC). The reference ontology is extracted from LCC, and the user's favorite terms are organized by using both the borrowing records of individual users and the notes keyed by librarians. The system is linked to check-information system, which collects user information from online interactions. The cataloging information as well as loan information are used as sources of user interests. However, this approach is limited to the area of scholar's information up to the book's title, cataloging information, and borrowing keywords.

Duong et al. (2009) propose an ontology-based recommender system for Web information. It engages ODP as a reference ontology, and improves it by automatically collecting users' information from the Web. It also exploits Lexico-syntactic pattern and WordNet to extract hyponyms of documents' features for updating profiles. It employs ontology matching algorithm to find similar users for collaborative recommendation and updating users' profiles. To learn the profiles, it collects text documents from Web including blogs, publications, and homepages, and creates Vector Space Model of documents in tf*idf to train ODP subjects. User interests described in feature vectors are then compared with the document's vectors under each leaf concepts in ODP. In fact, for each concept in ODP, associated documents are represented as feature vectors in tf*idf. For non-leaf concepts, the set of all documents associated to its immediate child concepts are considered. Finally, similar document vectors are used to learn the profiles. To update the profiles, the top N similar documents, using a threshold value, is searched. For each similar document, the uncommon keywords are added to the target ontology. The similarity over ontology is measured using the Sorenson-Dice relation of paired topics.

The work in Kodakateri, Gauch, Luong, and Eno (2009) exploits a subset of ACM Computing Classification System (CCS) to create the users' profile of CiteSeerX users. ACM taxonomy has been used as a reference ontology to categorize both visited and unvisited documents of the CiteSeerX collection. The user profiles are a list of ACM concepts and their corresponding weights, in decreasing order based on the weight. This taxonomy is limited to three levels and does not contain fine grained topics of Computer Science.

Liang, Yang, Chen, and Ku (2008) propose a query expansion personalization system for Internet users. It builds user profile using the semantic-expansion approach, and uses browsing history of users- the articles that users study on the Web. It firstly apply information retrieval method on reading documents to represents the documents by Vector Space Model (Manning, Raghavan, & Schütze, 2009), and then, adjust the weights using the explicit user's scores as well as temporal tag of each document. Next, it groups the similar keywords together- the keywords with the same meaning and synonyms, to become a concept. Finally, it expands the concepts by applying Spreading Activation (SA) method (Salton & Buckley, 1988) on the networked semantic tree to extend the users' query and document personalization. A semantic tree is a collection of relevant concepts and two types of relationships, "is-a" and non-"is-a". Each concept represented as a node, whereas link between two nodes represents a relation. Proposed semantic-expansion network includes 56 semantic trees, 265 concepts, and 470 keywords. In Spreading Activation (SA) process, the weights of primary concepts are spreading over the semantic tree and the weights of connected concepts are updated as follows: initially, the activation value (or interest score) of concepts are set to the VSM weights, and other concepts in the semantic tree set to zero. Having used a threshold value and a spreading distance, the weights of adjacent concepts are updated. The threshold is the minimum activation value required for a concept to activate its adjacent concepts. SA stops if the activation value of adjacent concepts is below the threshold or the maximum activation distance is reached.

The work presented in Yang, Hsu, and Lu (2010) proposes an ontology-based information recommendation for scholar domain in the fields of Fuzzy Theory, Artificial Intelligence, and Neural Network. The system exploits a domain ontology to enrich and disambiguate keywords of users' queries in search interface. It maintains the Web page retrieved by a Web crawler in a backend database and classifies the Web pages based on the shared keywords. Once a user queries the system, it searches in backend database to find corresponding Web page. If not found, a Web crawler is activated to collect relevant Web pages by Google search engine, and proceeds by classifier to classify the Web pages using the domain ontology.

Sugiyama and Kan (2010) engage recent publications of scholars and the neighboring articles (i.e., those articles which are cited by and referenced in other articles) and uses Cosine similarity for

finding similarity between candidate articles and scholar's profiles. It uses tf*idf as feature vector and an enhanced weighting scheme using Google's Page Rank method, which smooth the term frequency of target papers with cited and reference in scholars' publications. It tests the system on 15 senior and 13 junior researchers (Ph.D. and master students) having publication in DBLP in the field of Information Retrieval and Language Processing.

However, these reference ontologies lack sufficient ontological concepts of CS domain and do not support the top-down or general to specific relationship of concepts which is necessary in profiling scholars' background knowledge. These shortcomings hamper the exploitation of such ontologies as a reference ontology for modeling scholars' background knowledge.

3. Methodology

Our method consists of two phases: First, it constructs a reference ontology by engaging multiple taxonomies from the Web; and second, it creates profiles for a group of researchers that implements a function-based evaluation of the reference ontology (Gangemi, Catenacci, Ciaramita, & Lehmann, 2005).

3.1. Preliminaries

To avoid ambiguity, we define some popular terms. We exploit three terms interchangeably, that are "subject", "topic", or "concept", each one refers to the "components" of scholar's knowledge. As an example, "Unsupervised Learning" and "Swarm Optimization" are two topics/concepts of a scholar's knowledge. Taxonomy is a hierarchy of concepts, and ontology is a collection of concepts with "part-of" relations.

Scholars' knowledge is represented as a hierarchy of concepts (or hierarchy for short) (Sharman, Kishore, & Ramesh, 2007). The hierarchy supports the "Part-Of" relation explicitly, where each concept subsumes its relevant sub concepts, and each concept is a composition of some more granular knowledge items in the domain. In other words, the hierarchy supports general to specific relationship among ontological concepts. Each concept has zero or many sub-concepts called children, and a super-concept is called parent of its children. A concept with zero sub-concepts is called a leaf. Concept with no parent is called root. Each concept is annotated with a tuple [Cp, Wt], where Cp is a concept, Wt represents the weight of Cp within the knowledge area of respective scholar. The set of concepts, starting from leaf concepts up to and including the root build up the components of scholars' knowledge. Fig. 1 illustrates a typical structure of scholars' knowledge, containing a set of interconnected concepts, ordered as a hierarchy. As shown, the respective scholar has significant knowledge in "Artificial Intelligence", which is detailed by topics "Data Mining", "Knowledge Representation", "Neural Networks", etc. The weight associated with each topic demonstrates the degree of which the scholar has knowledge in that topic.

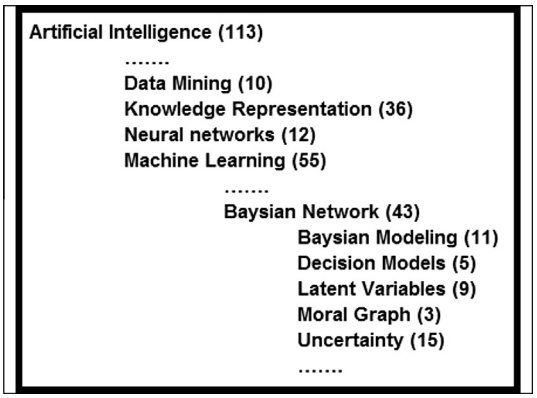

Fig. 1. A conceptual structure of a scholar's background knowledge in CS domain.

3.2. Framework

To develop a reference ontology for CS domain, we employ the following Web directories, which provide sufficient topics of the domain. In addition, DBpedia (Bizer et al., 2009) has been employed for concept transformation and disambiguation (Navigli, 2009) in the merge process:

- Open Directory Project (ODP) (www.dmoz.org) (Henderson, 2009)
- ACM/IEEE Computing Curriculum 2008 (www.acm.org/education) (Seidman & McGettrick, 2008)
- Best of the Web directory (BOTW) (www.botw.org) (Mahan, 2008)
- Wikipedia (www.wikipedi.org)

Fig. 2 represents the structure of subject extraction, transformation, and integration. As shown at the left side, subjects are extracted from base directories. Each directory provides different subject hierarchy in different granularity. Thus, concept selection, disambiguation, (Gal & Shvaiko, 2009), and transformation are proposed. At the right side, the validation process (or profiling) is applied, i.e., the academic knowledge of a number of researchers are extracted and mapped to the reference ontology to develop respective scholars' profiles. This part deals with ontology mapping, which populates the ontology with the concepts derived from scholar's knowledge. In the following subsection, components of the framework are described.

3.2.1. Characteristics of source taxonomies

We deal with two types of heterogeneity among the source taxonomies, which influence the integration and profiling process, including schematic (naming) and structural conflicts. Structural conflict refers to differences in the organization of information (Keet, 2004), while schematic conflict deals with differences in syntax of information. To highlight the diversity and heterogeneities among taxonomies, the significant properties of taxonomies which influence the merge process are described as follows:

- **Open Directory Project (ODP):** It is a human-edited directory on the Web, containing various topics including commercial, news, products, science, technology, events, music, and other similar topics. The main goal of ODP is "resource discovery" on the Web, and assists in locating descriptive information about general-purpose topics. Currently, ODP contains 786,225 subjects, varying from three to five abstraction levels (Henderson, 2009). The portion of CS topics contains 8471 general entries.
- **ACM/IEEE Computing Curriculum 2008:** It provides a standard, full course based, and community accepted terminologies for describing the body of scholars' knowledge (Liao et al., 2010). This curriculum is a pseudo-hierarchical arrangement of CS subjects as well as abstractions of courses. It currently contains 14 major areas, 161 units, and 1152 cores as well as associated concepts (Cassel, Clements, & Davies, 2008). Fig. 3 represents a partial view of the curriculum, representing two units of "Intelligent Systems", called "Basic Search Strategies" and "Machine Learning", their associated cores, and corresponding concepts. It shows, for example, if one wants to learn basic search strategies, he/she must learn associated cores and concepts in the blue boxes including problem spaces, problem solving by search, etc.

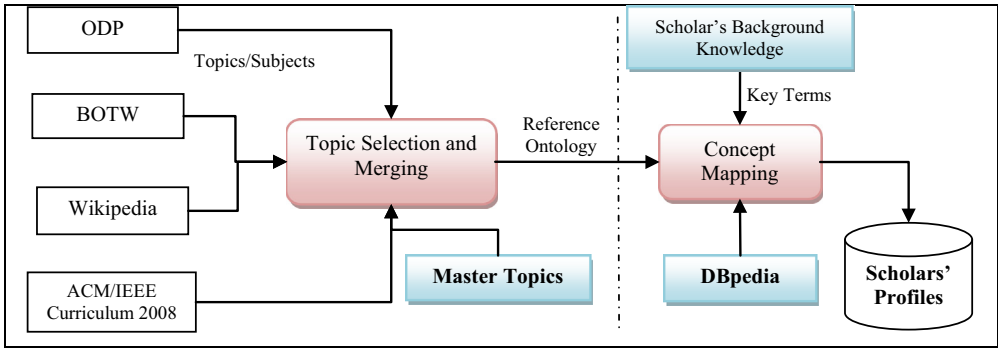

Fig. 2. The structure of construction a reference ontology (left side), and profiling scholar's background knowledge (right side).

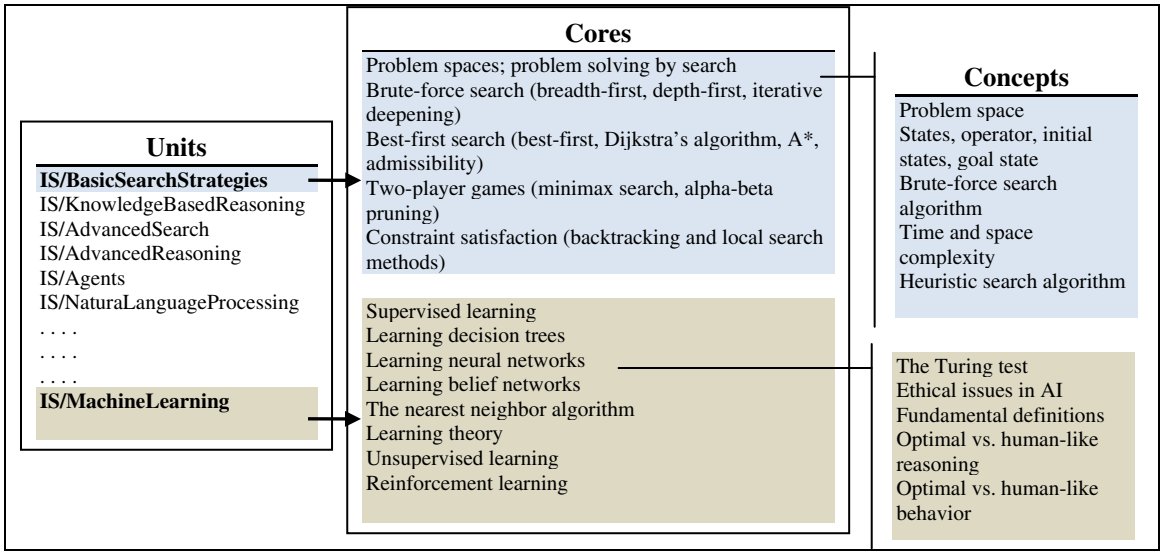

Fig. 3. A subject hierarchy extracted from ACM/IEEE curriculum 2008 (Cassel et al., 2008).

Table 3
A partial view of Wikipedia topics in AI field.

Knowledge engineering (8)	Knowledge representation (7)	Logic programming (12)	Machine learning (24)
Artificial intelligence			
Knowledge acquisition	Library classification	Automated theorem proving	Active learning
Documentation structuring	Ontology	Constraint programming	Algorithmic inference
Knowledge modeling	Semantic Network	Inductive logic programming	Bayesian networks
Frame language	Thesauri	Reasoning system	Classification algorithms
Ontology	Semantic desktop	Abductive logic programming	Cluster analysis
Decision support system	Argument mapping	Constraint logic programming	Knowledge discovery
Semantic reasoner	Classification systems	Situation calculus	Pattern recognition
		Stable model semantics	Data mining
		Unification	Decision trees
		Warren abstract machine	Dimension reduction
		Well-founded semantics	Document classification
		Yale shooting problem	Ensemble Learning

- **Wikipedia:** It is an expert agreement encyclopedia and a rich source of knowledge about the latest topics of CS domain (Medelyan, Milne, Legg, & Witten, 2009), which provides the "part-of" relation among the topics. Wiki topics are assigned to categories, which provide a broader sense of topics. The categories range from abstract topics such as "Computational Problems" to real world applications such as "Computer Graphics". At present, the number of top-level categories of Wikipedia is 17. Each category contains varied sub categories between 3 and 31. The dump of 2013 contains 594 distinct categories. As an illustration, Table 3 represents 4 of 14 subcategories of "Artificial Intelligence". Each topic/category has a number of sub topics, shown in parenthesis beneath the topics. Particularly, the field "Artificial Intelligence" has 14 sub-topics at the second level, 106 topics in the third level, and 195 topics in the last level.

Table 4
Top categories of BOTW hierarchy, only items with asterisk (*) are applicable for profiling scholar's knowledge.

Algorithms*	Data formats*	Mobile computing*
Anti-Virus	Data storage	Multimedia*
Artificial intelligence*	Education	Open source
Artificial life*	Electronic books	Parallel computing*
Bulletin board systems	Emulators	Performance and capacity
CAD and CAM*	Encyclopedia	Security*
Chats and forums	Ethics	Shopping
Companies	Graphics*	Speech technology
Computer law	Hacking	Supercomputing*
Computer Science	History	Systems*
Consultants	Home automation	Usenet
Data communications*	Human–computer interaction*	Virtual reality*

- **BOTW:** It is a Web categorization system, containing various subjects including art, finance, regional, health, computer, organization, and many other diffused topics. As depicted in Table 4, the "computer" category involves 36 subjects in the top level. However, it has many irrelevant topics: only 15 of 36 topics (marked by asterisk) are useful for profiling the scholars' knowledge in CS domain. The other topics are commercial applications or documentary information about the computers.
- **DBpedia:** It provides a structured and "real community agreement" of Wikipedia information. The information is organized by eleven RDF tags including, among others, "rdfs:Label", "rdf:Category", and "dcterms:Subject" (Mendes, Jakob, & Bizer, 2012). The "rdf:Category" is represented by two SKOS vocabularies[1] including "skos:Concept" and "skos:Broader", which are associated with a category name in Wiki page and its super categories. Currently, DBpedia consists of 415,000 categories in different disciplines (Bizer et al., 2009), providing a rich source of information for concept transformation and mapping.

3.2.2. Concept selection method

As discussed earlier, many topics in the Web taxonomies are neither in CS domain nor useful for profiling of scholars' knowledge. This leads us to design "selection criteria" to filter out outlier topics. Thus, we create a list of "controlling vocabulary" – the range of valid topics in CS domain, which assists in selecting CS vocabulary to be augmented to the reference ontology under construction. To do so, we identify the high-level topics of CS domain which appear in theme of CS's conferences, and categorize into major/subordinate topics. Overall, CS topics span into wide range of areas from theoretical studies to practical issues of computing (Shaw, Aho, & Bennett, 2004). Accordingly, if a candidate topic in the source directories is relevant to any major or subordinate topic, it will be selected to augment with the reference ontology.

3.2.3. Merging ontologies

The process of merging taxonomies is difficult, because we deal with domain concepts in different vocabularies as well as potential mismatch due to ambiguity problem (Degemmis, Lops, & Semeraro, 2006). In order to merge taxonomies, we enhance SMART algorithm (Noy & Musen, 1999) to fit into our method by incorporating Wikipedia as a conflict resolver. The SMART algorithm is a semi-automatic approach, which merges ontologies in syntactical and semantic levels without any assumption about the structure of participant ontologies. To improve the accuracy of merging, a semantic enrichment based on a thesaurus is applied (Degemmis et al., 2006).

In SMART method, the schematic or structural heterogeneities among involving taxonomies are resolved by matching the participant terms based on the syntactic similarities. Syntactic similarity means how similar are two terms based on their syntax (Mohsenzadeh, Shams, & Teshnehlab, 2005). In our approach, the semantic similarity is included. The semantic similarity refers to how two terms share the same meaning in a given context (Sánchez, Isern, & Valls, 2012). In order to measure the similarity between two terms, a voted similarity is computed. It computes the similarity between a term of source taxonomy (ResTax) and a term in the reference ontology (RefOnt) by following two methods:

(1) **Syntactic similarity:** It looks for identical term sense such as common substrings, common suffixes, and common prefixes. A normalized version of the Levenshtein's distance algorithm (Haldar & Mukhopadhyay, 2011) is employed. The similarity is a value in [0..1], where 1 indicates full term match and 0 for full difference. Thus, two given terms are syntactically similar if similarity value is greater than a threshold value. Eq. (1) computes the syntax similarity $SSM(\mathbf{S,T})$ between two terms **S** and **T**, where S ϵ ResTax and T ϵ RefOnt:

$$SSM(S,T) = max\left(0, \frac{min(|S|,|T|) - ED(S,T)}{min(|S|,|T|)}\right) \quad (1)$$

where **ED(S,T)** estimated the number of edits (character edition such as insert, delete, or replacement) required to change **S** to **T**. Lower case and upper case are also treated similar. As an example, if S = "dataset" and T = "Data Set", then SSM(s,t) = max{0,(7 − 1)/7} = 6/7 = 0.86.

(2) **Semantic similarity:** This measurement engages Wikipedia to find semantic relations between **S** and **T**. If both frequently appear in the same Wiki page, or one term appears frequently in the body page of another term, there is a semantic relation between **S** and **T**. More precisely, $\mathbf{Sem_{Sim}(S,T)}$ in Eq. (2) measures the degree of similarity based on the Lesk algorithm (Lesk, 1987), which takes into consideration the shared (or overlapping) words in a context. The context may be either a corpus of documents or a common knowledge base. The basic Lesk algorithm employs glosses found in Oxford dictionary, and a modified version of Lesk (Banerjee & Pedersen, 2002) takes the advantage of WordNet as a shared context. In our approach, we use Wiki glosses (the first paragraph of Wiki pages) and Wiki full-texts (second large paragraph) of Wikipedia (Strube & Ponzetto, 2006), which offer a rich context of technical terms of CS domain as shared context.

$$Sem_{Sim}(S,T) = Tanh\left(\frac{Overlapping(S,T)}{Length(S) + Length(T)}\right) \quad (2)$$

where **Overlapping(S,T)** is the count of shared words/terms in the gloss parts of both **S** and **T** in Wiki pages, and **Length(S)** is the total number of words/terms appeared in gloss part of **S**.

[1] http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/

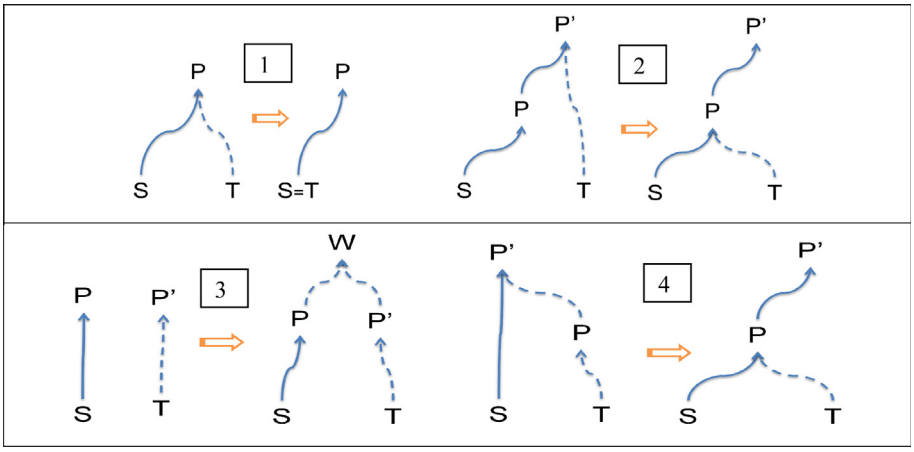

Fig. 4. Four states of relationship between S and T, respective parents P and P', and common parent W from Wikipedia.

We revise the core steps of SMART algorithm using Wikipedia in Algorithm 1. It repeatedly accepts a taxonomy (ResTax) as input and merges to the reference ontology (RefOnt). In order to position a concept in RefOnt, Fig. 4 illustrates the possible states and positioning decision visually. However, expert validation is also involved in positioning of terms for complicated cases when no matched Wiki page, in terms of semantic and hierarchy, has been found. This guaranties the quality of final artifact and decreases the chance of un-satisfiable concepts (terms with no parent) in the reference ontology.

Algorithm 1. A revision of SMART algorithm using Wikipedia as conflict resolver

1. *Input*: RexTax, a resource taxonomy.
2. Generate two lists of terms (or ontological concepts) from ResTax and RefOnt.
3. For each S ∈ ResTax do: {scan RefOnt to find a parent}
   For each T ∈ RefOnt,
   if *SSM(S,T)* > $Threshold_1$ then ignore S; {S and T are equivalent};
   else Position (S,T)
   if Sem_{Sim}(S,T) > $Threshold_2$, then Position (S,T)
   else store T in a candidate list (CL) with associated similarity score;
   Select a term $T_{highest}$ with highest similarity score from CL, Position(S,$T_{highest}$)
   Else add S to RefOnt as a main topic under the root.
   Tag S as assigned.
4. End

{——————— Position Nodes S and T ——————}
- Input: Node S, T
- Output: RefOnt
//Four states are recognized
Begin {
1. Both S and T are equivalent. Then S and T will have the same parent P, i.e., ignore S.
2. Concept S has a parent P whose parent is P' that is also parent T, then T becomes the child of P and P the child of P'.
3. Concepts S and T has two disjoint parent P and P', then lookup in Wikipedia to find a common parent W for P and P'.
4. Concept S has parent P' which is also parent P that is parent T, assign P to parent S and T and assign P' to be parent P.

}

3.3. Extracting scholars' knowledge

In order to measure the performance of our reference ontology, we extract scholars' knowledge and construct scholars' profiles in a real scenario. The source knowledge is a corpus of information including "reading articles" as well as "publications" of scholars. In order to extract the key terms from the corpus, the C-value/NC-value method (Frantzi, Ananiadou, & Mima, 2000), which is a combination of linguistic and statistical approaches is employed. This approach considers the semantic relations between words, and extracts the important multi-words (terms) in a corpus. This approach is a domain-independent method, which exploits contextual information in recognition of technical terms, and outperforms similar document indexing methods such as tf*idf and Glossex (Piao, Forth, Gacitua, Whittle, & Wiggins, 2010). The algorithm includes two related linguistic and statistical parts as follows (Frantzi et al., 2000):

- The linguistic part:

  {Identifies the candidate terms in a corpus}

  - Tokenizes the text by Penn Treebank Tokenization method (Marcus et al., 1994).
  - Assigns tag (e.g. noun, adjective, verb, preposition, determiner, etc.) to words with Brill's rule-based part-of-speech tagger (Eric, 1992).
  - Applies linguistic filters, i.e., the linguistic patterns on useful words to be extracted such as terms containing the following patterns:
    - $Noun^{+}Noun$,
    - $(Adj|Noun)^{+}Noun$,
    - $((Adj|Noun)^{+}|((Adj|Noun)^{*}(NounPrep)^{?})(Adj|Noun)^{*})Noun$
  - Removes popular stop-words, non-domain terms, and high frequencies words, i.e., the words not expected to occur as real terms such as great, numerous, several, year, good, etc.
- The statistical part:

  {Assigns a termhood value, C-Value, to a candidate term Z}

  - If **Z** is not a nested term, then the frequency of **Z** and its length is computed. The longer the term, the more important it likely is. Termhood is a function of length **Z**.
  - Otherwise (**Z** is a nested term), subtract the weight of involving "shorter terms" (substrings) from the frequency of **Z**. The formula is given in Eq. (3), where **Z** is a candidate term, **f(Z)** is the frequency of **Z** in the corpus, $\mathbf{T_Z}$ is the set of candidate terms containing **Z**, $\mathbf{P(T_Z)}$ is the number of such candidate terms.

$$C-value = \begin{cases} \log_2|Z| . f(Z) & Z \text{ is not nested} \\ \log_2|Z| . (f(Z) - \frac{1}{P(T_Z)} \sum_{BT_Z} f(B)) & \text{otherwise} \end{cases} \quad (3)$$

– NC-value incorporates context information to C-value. By definition, a context word appears near a term/word in the given corpus. Typically, context words are adjectives, nouns, and verbs that either precede or suffix a shorter term. Eq. (4) represents the NC-value relation, where **Z** is the candidate term, $\mathbf{C_Z}$ is the set of context words of **Z**, $\mathbf{f_a(B)}$ denotes the frequency of **B** (context word) in the corpus, **weight(B)** is the weight of **B**. The constant value **α** and **β** characterize the context information, where **α + β = 1**. Appropriate values rank the most important terms and filter out unused or "noise" terms. The greater **α** lessens the neighboring terms in the corpus.

$$NC-value = \alpha * C-value(Z) + \beta * \sum_{BC_z} f_Z(B) * weight(B)$$
$$weight(w) = \frac{t(w)}{n} \quad (4)$$

where **weight(w)** denotes the weight of **w**, that is a context word, **t(w)** is the number of terms the word **w** appears with, **n** is the total number of terms considered.

The key terms are represented by feature vectors: each vector's entry is a tuple **[T, W]**, where **T** is a term and **W** is corresponding to C-Value/NC-value. However, this method produces outlier terms, i.e., the terms that cannot fully characterize or index a document (Amini, Ibrahim, Othman, & Selamat, 2014). Thus, a feature selection method is always applied (Chen, Huang, Tian, & Qu, 2009) to select most suitable items. In our approach, we apply the following steps to prune uninformative terms:

- ***Step 1:*** Reorder the list of terms based on the weight **W** decreasingly.
- ***Step 2:*** Non positive terms, the terms with negative or zero weights, are removed from lists. According to (Eq. (3)), if $f(Z) < \frac{1}{P(T_Z)} \sum_{BT_Z} f(B)$ then the weight of compound term is zero or negative. This means that the sub parts of a compound term appear more frequent than the longer compound term. For example, the frequency of "ad hoc" is more than frequency of "ad hoc network". Thus, "ad hoc" has negative weight, and it is a good candidate to prune. This implies that the longer terms bear more knowledge than the smaller ones.
- ***Step 3:*** The bottom 25% of each term list are truncated, as they appear frequently in the corpora and contain "non-technical" terms such as "Wide Variety", "Future Work", and "Generalized Approach" (Amini et al., 2014).
- ***Step 4:*** The terms that are general, i.e. non-discriminant, neutralized, uninformative, which give no sense of conception in the domain are removed, i.e., the terms that can be changed without affecting the topic. We employ domain-specific vocabulary of Computer Science[2] to exclude the un-matching words (Kozakov, Park, Fin, & Drissi, 2004). Examples of such neutralize terms are "several examples", "large number", "results show that", and "experimental results".
- ***Step 5:*** Variant terms based on the syntactic similarity are aggregated. The variant terms which appear in different forms are inflectional variants (a singular or a plural variant), compound variants (data set versus dataset), orthographic variants (terms with special characters such as hyphens such as object-oriented design), case-sensitive, misspellings (behavior versus behaviour), overlapping, and abbreviations (Kozakov et al., 2004). As an example, the term "artificial neural network" and "neural networks" are overlapping, and thus can be aggregated to the longer term. The score is set to the maximum value of respective scores. We follow a dictionary lookup approach using Wiktionary[3] to refine the candidate terms.

3.4. Construction of scholar's profiles

In order to build scholars' profiles, the key terms which represent the scholars' knowledge are mapped to their corresponding concepts in the reference ontology. Although the extracted terms are descriptors of individual scholar's knowledge, they may contain non-topical terms which do not belong to CS domain. This is a conflict in concept level, where for a given term, its semantically equivalent term from an expert consensus knowledge base is replaced (Nguyen, 2006). This issue naturally happens because two or more knowledge items refer to the same real word object, but are represented by different ontological concepts. For example, "Data Encryption" and "Injection Attack" all refer to a single knowledge item "Data Encryption" but in different syntax. Thus, they are mapped to the more general term "Data Encryption" in the reference ontology.

Therefore, we map feature vectors by transforming each key term to its equivalent concept in the reference ontology based on syntactic and semantic relatedness measurements (Medelyan et al., 2009). The mapping uses DBpedia as an indexing mechanism, which assigns similar terms, either grammatical variations, synonymy, or semantically relevant (Salahli, Gasimzade, & Guliyev, 2009) to the ontological concepts. In fact, DBpedia provides "relevant concept", which is the possible meaning of a given term under the standard terminology. For instance, "Fuzzy Systems" and "Fuzzy Logic" are semantically relevant, because the former refers to the broader concept "Fuzzy Logic". In DBpedia data set, it is tagged by skos:Broader, and both concepts belong to the same Wiki category tagged by rdf:Category.

3.4.1. Interface to DBpedia

The DBpedia knowledge base has two joint components: "Linked Data" and the "Applications" (Bizer et al., 2009). The "Linked Data" publishes RDF data on the Web, which relies on URI resource identifiers and HTTP protocol to retrieve resource descriptions. The DBpedia resource identifier[4] returns RDF descriptions when browsing by a Semantic Web agent (e.g., Gruff[5]), and returns an HTML view of the same information in the Web browser. We employ nine different attributes which provide alternative explanations (relevant meaning) of a term in DBpedia. Table 5 summarizes the prefix, attribute names, and brief specification of the attributes. These attributes help in mapping ambiguous terms to semantically equivalent terms in DBpedia. Fig. 5 represents the RDF graph of resources with aforementioned attributes. In order to acquire related topics of a given term, we simply traverse the graph by querying the RDF graph using the prefix and attribute names.

3.4.2. Mapping with DBpedia

In order to map the key terms by mediation of DBpedia, we first create a list of pairs $\mathbf{(T_i, D_i)\ \{i = 1,2,..,n\}}$, where $\mathbf{T_i}$ is a candidate term corresponding to scholar's knowledge, and $\mathbf{D_i}$ is an "unassigned" concept (term) to be assigned by DBpedia. The term $\mathbf{T_i}$ is looked up in RDF graph against different tags including Label, Subject, Category, and Broader. Once any match is found, corresponding tag which is a DBpedia data is assigned to $\mathbf{D_i}$. If no match has been found, the term is treated as "noise" and ignored from the input list. The mapping algorithm is given in Algorithm 2.

[2] http://www.springerreference.com

[3] http://en.wiktionary.org/wiki/Wiktionary:Main_Page

[4] Such as http://DBpedia.org/resource/category:Machine_Learning

[5] http://www.franz.com/agraph/gruff/

Table 5
The types of RDF resource for transforming a term to its corresponding term in DBpedia.

Prefix	Attribute	Specification
Dcterms	Title	Specifies a name by which the resource (URI) is formally known
Dcterms	Subject	Represents key phrase, classification code, or topic or a resource
foaf	PrimaryTopic	Specifies the primary topic of a URI associated with a subject
RDFS	Label	Provides a human-readable version of a resource's name (alternative title)
SKOS	Broader	Defines broader topics of a given URI, which are the superior topic of a term
SKOS	Subject	The resource identifier which is linked to an object with a predicate like ⟨Subject, Predicate, Object⟩
SKOS	PrefLabel	A short human readable label of a resource
RDF	Category	Represents the category name of a subject (label)
OWL	SameAs	Links multiple information about a particular topic

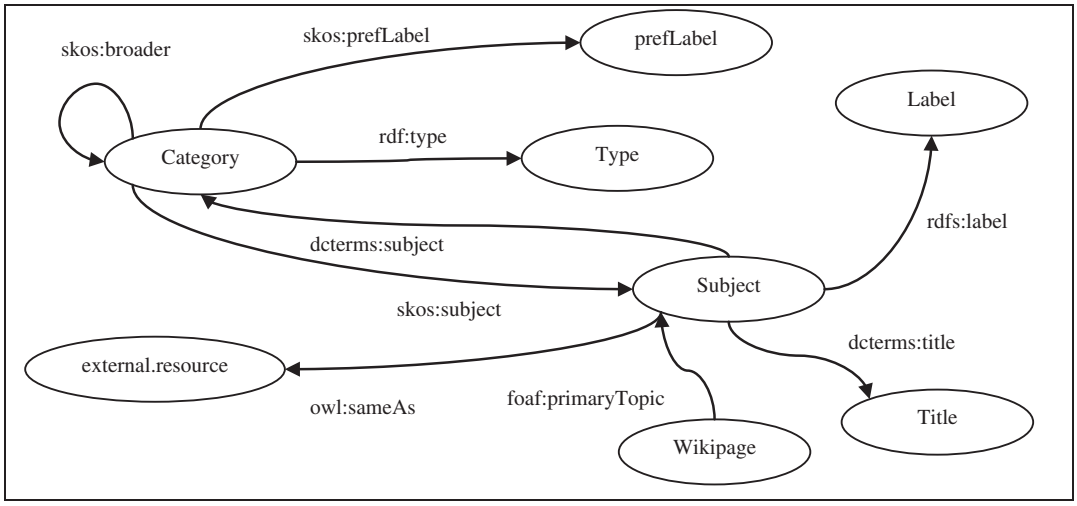

Fig. 5. RDF graph of DBpedia's resources for acquiring related topics of a given term.

Algorithm 3 populates the reference ontology with key terms extracted from the scholars' knowledge. Each concept in the reference ontology is represented by a tuple **[C_p,W_t]**, where **C_p** is a concept and **W_t** is its associated weight. Moreover, each key term (scholars' knowledge item) is represented by tuple **[T, W]**, where **T** is a term and **W** is its associated weight. The function **Similarity(T,S)** computes an average similarity using syntactic and semantic similarities between input terms **T** and **S**.

Algorithm 2. Key terms of scholars' knowledge are transformed by DBpedia

Create a list of pair (Ti, Di),where Ti is a candidate term, and Di is Null
For each pair (Ti, Di), lookup Ti in DBpedia graph
do case {
rdfs:Label = Ti then Di ← foaf:PrimaryTopic
dcterms:Subject = Ti then Di ← dcterms:Subject
rdf:Category = Ti then Di ← rdf:Category
skos:Broader = Ti then Di ← dcterms:Subject
otherwise: Di ← "NT" {Noise Term}
end case

Once all key term are mapped, the weight of adjacent concepts in the reference ontology are normalized using a variation of Spreading Activation (SA) approach (Salton & Buckley, 1988). The procedure is as follows: it spreads the weight of input node (concept **O_i**) toward the adjacent upper levels in the hierarchy. The spreading weight first affects the sibling (adjacent concepts) of **O_i** and repeats through the hierarchy one link at a time to reach the maximum level, **MaxLevel**. In the hierarchy tree, **MaxLevel** is 1 for the root concept, and 2, 3, 4, etc. for downward levels, respectively.

Algorithm 3. The process of term mapping and ontology population

– *Input: a set of terms featured by [T, W], reference ontology RefOnt*
– *Output: populated reference ontology- the scholar's profile*
1- For all ontological concept [C_p, W_t], set $W_t \leftarrow 0$
2- For each term T ϵ feature vectors do:
a. Scan RefOnt to find a concept O_c
– If *Similarity* $(T,O_c) >$ *Threshold* then $O_c.W_t \leftarrow W$, $O_c.C_p \leftarrow T$
Else *ignore*(T)
b. Normalizing RefOnt: *NormalizeHierarchy* (O_c)
{*Spreading Activation*}
c. Remove T from input list
3- Normalize tree (O_c)
4- End.
{————— Normalize Tree Function —————}
- Input: Node O_c
- Output: Normalized RefOnt
- Const MaxVal:K, Indicates maximum valid weight in RefOnt
1- For each upper level node O_u connected to O_c
a. Set $O_u.W_t \leftarrow O_u.W_t + \Sigma\ O_j.W_t$, where O_j is a sibling of O_c
b. *level* = *level* + 1
c. If level < Maxlevel *then*$\{O_c \leftarrow O_u$, *loop* $(a)\}$
2- If $O_c >$ *MaxVal* then $O_c.W_t \leftarrow O_c.W_t * 0.1$ for all O_c in RefOnt

Table 6
Controlling vocabularies: the main topics and sub-topics of World CS conferences in 2012.

Main topics	Sub-topics
1 Algorithms and theory	1.1 Algorithms 1.2 Geometric algorithms 1.3 Other specialized subtopics
2 Programming languages	
3 Concurrent, distributed and parallel computing	3.1 High-performance computing
4 Software engineering	4.1 Formal methods
5 Operating systems	
6 Computer architecture	6.1 Computer hardware 6.2 Real-time and embedded systems 6.3 Computer-aided design
7 Computer networking and networked systems	7.1 Wireless networks
8 Security and privacy	8.1 Cryptography
9 Data management	
10 Artificial intelligence	10.1 Automated reasoning 10.2 Computer vision 10.3 Natural language processing
11 Computer graphics	
12 Human–computer interaction	
13 Computational biology	

4. Implementation and analysis

This section describes the details of constructing a reference ontology for CS domain and creating scholars' profiles for evaluation purpose.

4.1. Capturing Topics from Source Taxonomies

In order to capture appropriate topics/subjects from the source taxonomies, we first created a controlling vocabulary, i.e., the core topics of Computer Science domain. Thus, we simply crawled the Web using a general purpose Web Crawler[6] and extracted a list of ten international CS conferences in 2012, and retrieved the conference themes. Accordingly, we identified 13 main topics and 13 corresponding sub-topics, falling into four main areas including: (1) Theory of computation, (2) Algorithms and data structures, (3) Programming methodology and languages, and (4) Computer elements and architecture. Table 6 represents the title of main topics in controlling vocabulary.

In order to capture the subjects from source taxonomies, we first focused on ODP as a primary hierarchy (RefOnt), and pruned irrelevant topics using the above controlling vocabularies. Since ODP contained a great number of subjects that were not a kind of CS research topic, e.g. Groups, FAQ and Help, Journals, etc, we applied an intuitive filter iteratively in five stages to sift irrelevant topics, i.e., the topics that frequently appear in CS context but do not pertain to the scholars' domain as "academic knowledge".

Table 7 depicts the filter description, examples, and the number of resulting subjects after applying each filter. As shown, engaging filters lessened the number of concepts to 747 (about 8% of the original size) and made the final topics most relevant to scholars' knowledge. This slight percentage of subjects stressed that further taxonomies need to be merged with RefOnt.

In order to merge other taxonomies with RefOnt, topics and their hierarchical relationships from BOTW and Wikipedia directories are extracted, pre-processed, and transferred to Excel sheets. In this stage, crosschecking (cross validation), inclusion or exclusion of topics for three topmost levels of RefOnt is carried out. We engaged Protégé 4.2 ontology editor, which automatically imports concepts from the sheets and made individual sub-ontology from each taxonomy. We finally merged the sub-ontologies one-by-one to RefOnt using SWOOP framework (Kalyanpur, Parsia, Sirin, Grau, & Hendler, 2006) and the revised SMART algorithm. This framework assisted in semi-automatic merging of ontologies, and controlled the inconsistencies, duplicates, and structural properties of the developing ontology.

To merge ACM/IEEE curriculum's topics with RefOnt, a heuristic filter is designed to exclude irrelevant topics from the curriculum. Considering the "controlling vocabulary", the most ineffective topics are recognized as follows:

–Undergraduate Subjects	–Introductions	–Programming Skills
–Definitions	–Applications	–Hardware

These topics are either irrelevant or general with respect to scholars' knowledge in CS domain. After pruning, the remaining areas and cores that were 54% of the total were selected and transferred to an Excel sheet. Finally, the sheet was imported to Protégé 4.0 and a sub-ontology was created. The sub-ontology was then merged with RefOnt in SWOOP. Fig. 6 represents a partial view of the final reference ontology, which consists of 2035 concepts in 15 top levels. It also includes 7 maximum levels, 2 minimum levels, and 4.4 levels on average. The maximum branching factor (siblings) is 29, and the average branching factor per concept is 5.6.

4.2. Constructing scholars' profiles

This section deals with the profiling of scholars' knowledge with our reference ontology. The goal is to develop the functional characteristics of the reference ontology in practice. It involves selecting a sample scholars' data set, feature extraction, and profiling.

4.2.1. Data set

We employed a part of data sets in which developed in our previous study (Amini, Ibrahim, & Othman, 2013). In fact, we collected a corpus of scholarly information from 25 volunteer researchers at Faculty of Computing[7], UTM, who studied in any sub-fields of Computer Science. They were invited to supply relevant articles in line with their own research topics including "reading papers" and "publications". We collected 20 to 25 normal articles from each researcher, 8–20 pages in length, which produced 625 PDF files in total. Each file is pre-processed and converted to plain text by removing uninformative parts such as images, tables, references, acknowledgment, algorithm, footnotes, authors' affiliation, and header/footer, because these parts do not contain domain-relevant information.

4.2.2. Feature extraction

We applied the C-Value/NC-Value algorithm on the corpus using the Jate 1.11 toolkit (Zhang, Brewster, & Ciravegna, 2008), which is an open source tool in Java for Linux platform. The number of terms that were extracted by was 550 to 630 per scholar. The term lists were partly ineffective since they contained outliers or non-domain terms such as "Defect Size" and "Test Image". Thus, five filters given in Section 3.3 were applied to prune the outlier terms. The final output of this phase was 25 lists of key terms with associated scores (C-values/NC-value with $\alpha = 0.8$, $\beta = 0.2$). Table 8 represents the top 10 key terms for two scholars. As shown, scholar 1 has the highest knowledge in the topics "bp algorithm", "neural networks", and "edge detection", bearing more interest in "Evolutionary Computation", while scholar 2 is interested in "vision systems", "industrial vision", and "image processing".

[6] www.webcrawler.com

[7] www.comp.utm.my

Table 7
Five types of heuristic filters applied to the ODP directory.

Filter No	Irrelevant subjects	Examples	Resulting subjects
1	Hardware components, specific software, education	Hard disk, peripheral, storages, drivers, home automation, personal tools, freeware, organizations, training resources	2303
2	Internet services, commercial tools, organizations	Chat, groups, messengers, broadcasting tools, domain names, weblogs, web hosting, wikis, trademark, tools, brands, markets, mailing lists, ERP, service providers, software foundations	1925
3	Projects, programming, information technologies	Open source, platforms, compilers, methodologies, agents, networking, engineering, etc.	1327
4	Applications, products	Programming languages, tools, development kits, technologies, libraries, operating system, compilers, DBMS, cookies, plug-in, protocols	998
5	Non CS topics	Directories, ethics, history, help, QA, country name, management, configuration, administration, etc.	747

4.2.3. Profiling

In this section, we transformed the candidate terms to their corresponding ontological concepts in the reference ontology. Since the syntax and semantic of key terms compared to the concepts in the reference ontology were different, DBpedia was engaged to mediate (transform) the mismatches. In order to retrieve information from DBpedia, the Virtuoso Faceted Browser Web service at <http://dbpedia.org/fct/> has been employed. It explores DBpedia's URIs and provides text search and article indexing on DBpedia in an integrated interface. This service gives sufficient flexibility and freedom to navigate the information space. Table 9 represents a sample of output terms. The first column is $\mathbf{T_i}$ - the terms of a scholar's knowledge, and the second column represents various RDF tags that have been located for $\mathbf{T_i}$. The last column is the type of assignments chosen by Algorithm (2), which has been matched with each $\mathbf{T_i}$. It should be noted that there is no assignment for two $\mathbf{T_i}$ including "FULL DUMP" and "GRAPHICAL SIMULATION". In addition, there are two assignments for **Ti** = "Data Collection". Thus, unassigned terms are not mapped to the reference ontology, because they are uninformative and have no corresponding concept in the reference ontology.

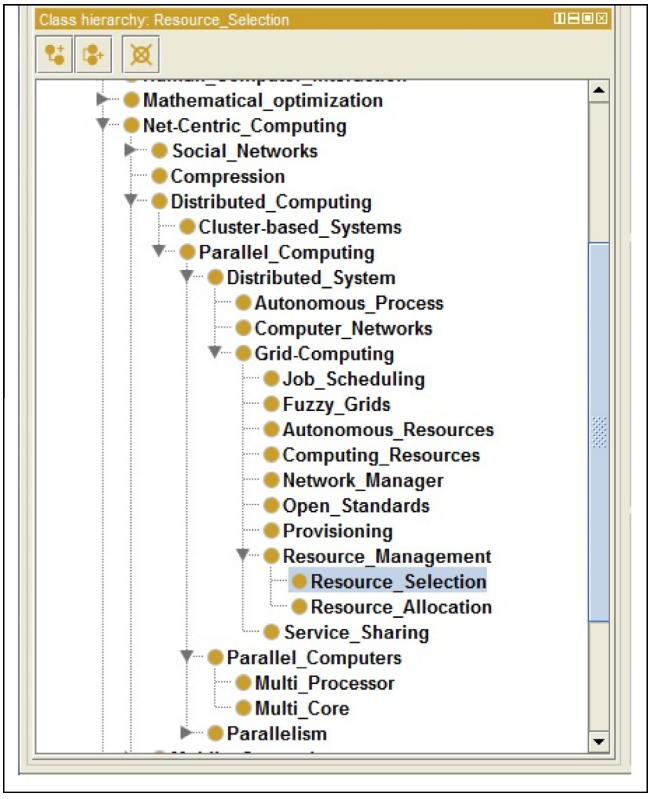

Fig. 6. A partial view of final reference ontology (RefOnt) in CS domain, integrated from four Web taxonomies.

Table 8
A partial listing of top 10 key terms extracted from two scholar's corpora.

Scholar 1		Scholar 2	
Key term	Score	Key term	Score
Bp algorithm	26.33	Vision system	39.44
Neural network	21.25	Measurement process	39.18
Edge detection	18.00	Industrial vision	37.29
Vector torque	18.00	Ceramic tile	36.88
Service selection	16.43	Straight line	35.43
Inertia weight	15.00	Machine vision	32.28
Local search	14.89	Image processing	31.74
Heuristic algorithm	14.50	Vision system	27.50
Cloud model	13.00	Dimensional metrology	27.11

Fig. 7 depicts a profile and an example mapping of key terms to ontological concepts by DBpedia. As shown, the key terms which are marked in color, e.g., "BP Algorithm" and "Neural Networks" are mapped to "Back Propagation Algorithm" and "Neural Networks", respectively. Similarly, the term "PSO Algorithm" which contains abbreviations, is mapped to "Particle Swarm Optimization". A good feature of Virtuoso Facet Browser is that it automatically finds abbreviations of technical terms, which frequently appear in CS domain. Additionally, the score of target ontological concepts are assigned with the terms' score (C-value/NC-value of input terms).

5. Evaluation

Generally, ontology evaluation is important when the ontology is constructed from different heterogeneous knowledge sources, which in turn affects the usefulness of users' profiles (Tartir, Arpinar, & Sheth, 2010). It is also difficult to compare the user profiles with one another because each one engages a different reference ontology. Thus, we adopt the function-based approaches, which evaluate the functional characteristics of a reference ontology in practice, and data-driven approaches, which measure the

Table 9
A sample of term mapping/transformation by DBpedia.

Terms (T_i)	DBpedia terms (D_i)	Type of assignment
Ad hoc networks	Wireless network	L
Authentication phase	Network admission control	S
Authentication process	Network admission control	L
Congestion control	Congestion control	P
Performance analysis	Network performance	L
Data collection	Data management	P, B
Data processing	Data processing	B
Data packets	Data packet	L
DSR simulation	Dynamic source routing	B
Full dump	<NT>	<NT>
Graphical interface	Graph connectivity	L
Graphical simulation	<NT>	<NT>

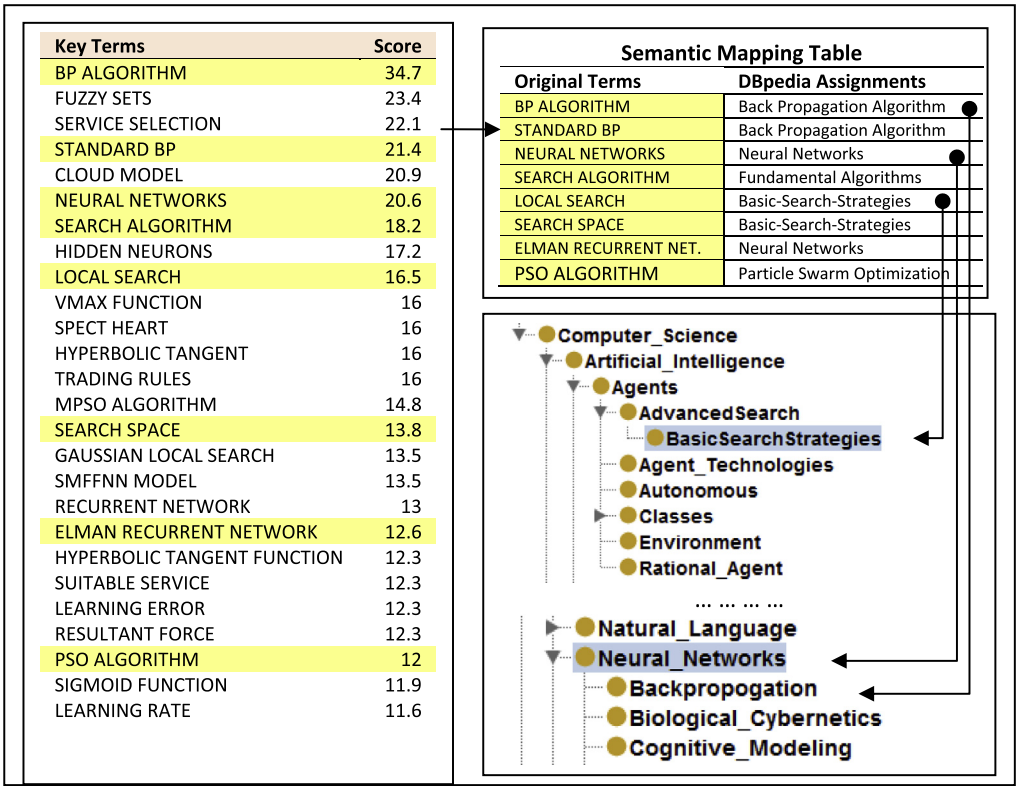

Fig. 7. A section of profile, representing the mapping of key terms to their corresponding ontological concepts by DBpedia.

contextual and structural aspects of an ontology (Gangemi et al., 2005). The generic metrics including cost of construction, type of methodology, building tool, ease of maintenance, etc., are of no concern. Our goal is to demonstrate how accurately scholars' knowledge is profiled using our reference ontology. Therefore, the following dimensions were measured:

1. Concept coverage
2. Accuracy of profiles
3. Degree of concept specificity
4. Ontology richness
5. Structural validation

To deal with the first measurement, we evaluated the percentage of matched scholar's knowledge (terms that successfully mapped to the ontology), i.e. how many knowledge terms are mapped to the reference ontology. The second measure concerns the accuracy of reference ontology, indicating the percentage of mapped scholar's knowledge into the ontology. The third one measures how the ontology models the CS domain. The fourth one measures the amount of conceptual information retained in the ontology. Finally, the fifth one aims to check the ontology against the structural properties such as concept consistency and correctness.

Table 10
The percentage of concept coverage for 25 scholars in CS domain.

Scholar ID	Matched%	Scholar ID	Matched%	Scholar ID	Matched%
1	87	10	80	19	88
2	94	11	73	20	89
3	88	12	83	21	89
4	78	13	81	22	34
5	81	14	79	23	23
6	90	15	88	24	16
7	85	16	83	25	18
8	80	17	92	–	–
9	89	18	89	–	–

- Concept coverage

In order to determine how well the scholars' knowledge has been represented by the reference ontology, we simply counted the key terms which mapped to the correct ontological concepts. Then, the ontology was penalized for unassigned key terms. Table 10 represents the percentage of covered knowledge in terms of concept for 25 participant scholars. As shown, the matching percentage for the first 21 scholar's ID is greater than or equal to 73% and the average is 85%.

In addition, to demonstrate that the reference ontology truly represents the CS domain, we incorporated four scholars of multi-disciplinary fields to the experiment to treat as negative examples, including wireless communication, business modeling, and information technology disciplines. As shown, for scholar's ID 22 to 25, the matched percentage is decreased substantially to less than 34%. This downfall indicates that our reference ontology captures the knowledge items of CS domain decidedly.

However, for each CS scholar, un-mapped concepts exist due to two reasons: First, the ontology resources, including ODP, BOTW, etc., are not complete by nature, since each one is developed for a particular functionality and do not cover the entire areas of Computer Science. In addition, they grow up over time, which indicates CS areas are growing over time. Second, the interdisciplinary concepts such as "Quality Control", "Manufacturing Process", "Supply Chain", and the similar are not belong to CS domain, though they have been extracted as key terms in the feature extraction phase.

- Accuracy of profiles

Accuracy of profiles means how well the terms of scholars' knowledge are captured by, and distributed across, the reference ontology. Thus, two affecting factors are measured: (1) the percentage of mapped terms, which have been mapped to the ontological concepts, and (2) the levels (depth) of hierarchy which

Table 11
The percentage of concept coverage on different depth of reference ontology.

Scholar ID	Matched level (%)			Scholar ID	Matched level (%)		
	3	5	7		3	5	7
1	53	83	87	14	63	71	79
2	62	89	94	15	70	85	88
3	60	76	88	16	57	66	83
4	67	71	78	17	54	79	92
5	64	77	81	18	65	73	89
6	61	78	90	19	77	81	88
7	50	81	85	20	71	88	89
8	64	78	80	21	75	84	89
9	59	77	89	22	27	34	34
10	67	76	80	23	04	13	23
11	66	71	73	24	01	07	16
12	59	78	83	25	02	12	18
13	63	74	81	–	–	–	–

capture the terms. More levels in the ontology force more processing time and complexities (Trajkova & Gauch, 2004), while lower levels are prone to generality and inaccurate mapping. Thus, we measured the accuracy of scholars' profiles in different levels. We calculated the concept coverage for 3, 5, and 7 levels, and compared the matched percentages- the percentage of knowledge terms that are mapped to the reference ontology.

Table 11 lists the percentage of matched terms for different levels of scholars' profiles. When considering only 3 levels, the concept coverage is relatively low since concepts spread on level 1 through 3, which comprise broad and general topics of CS domain. The chart in Fig. 8 also compares the ontological fitness for three different depths. As shown, the percentage of coverage at depth 3 is medium compared to other depths. The coverage is also increased at depth 5, but improved slightly at depth 7. It indicates that as we incorporate more knowledge levels and concepts after level 5, a few factors such as concept coverage and accuracy are raised. A trade-off between the maximum depth of the reference ontology and the tolerable complexity was made. In this study, the maximum depth of 5 has been chosen, meaning that no significant knowledge items are lost without engaging higher levels.

We also compare our reference ontology (at level 5) with ACM and ODP ontologies. Fig. 9 depicts the comparison. As shown, our reference ontology outperforms the other ontologies in terms of accuracy when mapping the knowledge items of 21 first scholars to the reference ontology. For scholars' ID 22 through 25, the status is reverse because it does not encompass general and non-CS topics.

- Degree of specificity

The specificity of an ontology with respect to a domain is measured by "domain relevance" measure (Velardi, Fabriani, & Missikoff, 2001), which is the number of domain terms captured by an ontology for user profiling. In Table 10, the column "Matched Level" shows that our reference ontology was able to capture a majority of scholar's knowledge, ranging from 73% to 94% for the

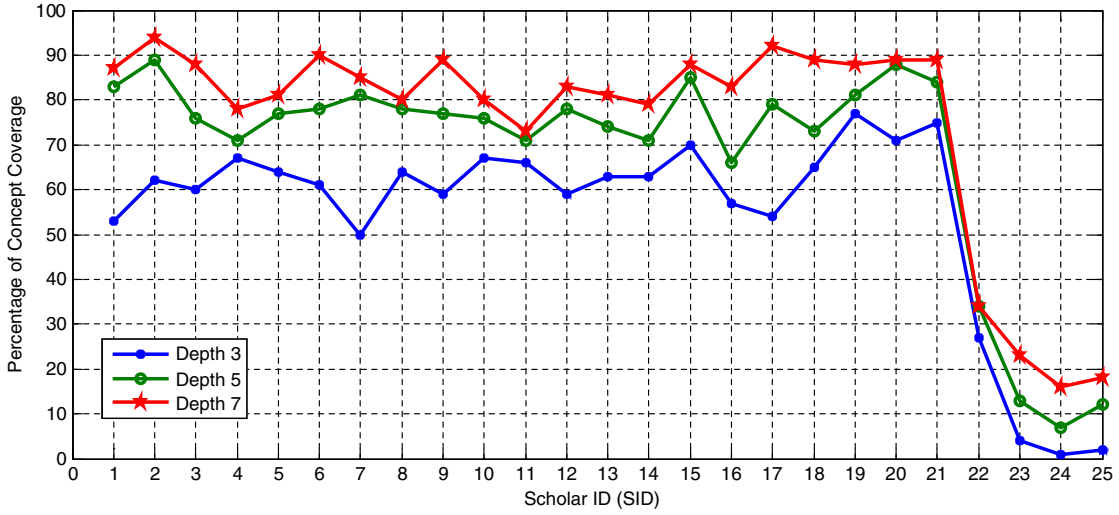

Fig. 8. The visualization of concept coverage on depths 3, 5, 7 for 25 scholars (SID 1–21 are CS, but SID 22–25 are non-CS scholar).

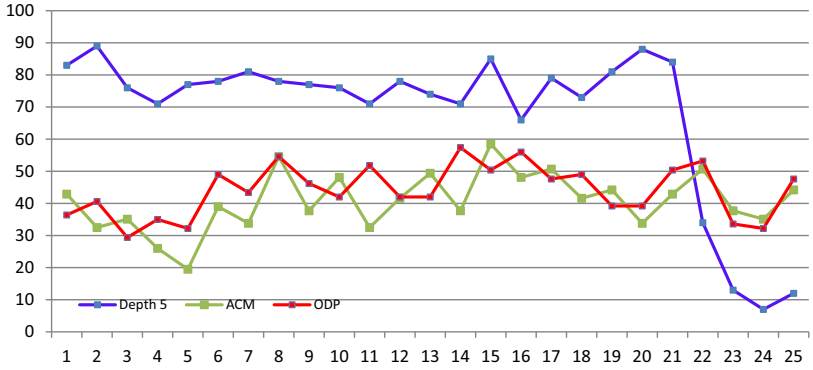

Fig. 9. The comparison of concept coverage of our reference ontology (on depth 5) with ACM and ODP ontologies, for 25 scholars (SID 1–21 are CS, SID 22–25 are non-CS).

Table 12
The comparison of different reference ontologies in terms of ontology richness.

Approaches	Type of ontology	Total topics	Top topics	Average levels	Average sibling
Sieg and Burke (2007)	ODP	4470	11	3.0	7.4
Sieg, Mobasher, and Burke (2007a)	ODP	563	11	6.0	4.0
Mohammed et al. (2010)	ODP	NA	11	3.0	NA
Liao et al. (2009)	LCC	853	13	2.0	3.5
Kodakateri et al. (2009)	ACM CCS	1470	11	3.2	5.0
Trajkova and Gauch (2004)	ODP	2991	13	3.0	6.1
Our approach	**Merged**	**2035**	**15**	**4.4**	**5.6**

top 21 scholar's ID in the known areas of Computer Science. Nevertheless, for scholar's ID 22 through 25, which are non-CS fields such as Information Technology, it captures very few knowledge terms in different levels. In particular, for those fields of study that contain less common topics with Computer Science, the match percent is significantly low (1% to 13%). This analysis proves that our reference ontology fits solely with the Computer Science domain at an average specificity of 83.5%.

- Ontology richness

Ontology richness is inspired by the question of how conceptual knowledge is distributed over the reference ontology. It is measured based on the amount of relational information spread over the ontology compared to the number of concepts, i.e., the average number of ontology relations per concept. The relation is measured by total concepts, the number of topmost concepts, average number of levels, and the average siblings or branching factor (BF) of concepts. Therefore, richness is measured by comparing the richness properties of ontology with some base (golden) ontologies (Aquin & Schlicht, 2009).

Therefore, we compared our reference ontology with six related works. Table 12 outlines the distribution of classes and the relationships across different ontologies. As described in Section 2, the ontological concepts in the reviewed works are too general and pertaining to low-level terms of CS domain. However, our reference ontology contains a higher number of top topics (15), higher number of average levels (deepness) compared to the number of topics (4.4), and a higher average branching factor compared to the levels (5.6). Additionally, the total number of topics is 2,035, which is an outstanding improvement for a reference ontology in CS domain.

- Structural validation

The primary goal of structural validation is to prevent the scholar's recommender system from using an inconsistent or incorrect reference ontology (Obrst, Ashpole, Ceusters, Mani, & Smith, 2007). Structural validation aims to check the ontology against structural and syntactic requirements such as consistency and correctness. It is a type of diagnostic task over ontology properties and attributes. To perform such evaluations, SWOOP tool was employed for automatic detection of possible inconsistency in the reference ontology. The SWOOP's reasoner reliably assisted in detecting and fixing the structural errors.

Having engaged this tool, we first performed automatic diagnosing test over the ontology to understand the cause and source of problematic concepts. Then, some amendments regarding the cost and benefits were applied. We repeated these cycle until a final checking showed that no syntactical problems such as cycles in the hierarchy, disjoint partitions, un-satisfiable concepts (concepts with no parent), and redundant concepts under a particular parent were exists.

Finally, we requested five domain experts in CS domain and ontology engineering to assess our reference. Since the number of topics comparing to the time needed for verification was high, they were asked to check randomly some areas of top topics downward to the leaf to verify any inconsistency.

6. Discussion, further works, and conclusion

In this study, we first explained the issues of engaging the traditional reference ontology such as ODP in modeling scholars' background knowledge, and then presented an approach for integrating multiple heterogeneous and free Web taxonomies to construct a new reference ontology for scholar domain. Our approach involves a sound framework and implementation, because it is independent of the source taxonomies. In this section, we discuss the results, suggest further research, and draw conclusion.

6.1. Discussion

Generally, a trivial solution to profiling scholars' knowledge is to engage a popular reference ontology, and afterward, to see whether knowledge items are augmented to it. The disadvantages of such approach are: (1) the difficulty of integrating new concept to the ontology, and more importantly, (2) the resulting profiles do not match the actual scholar's knowledge, which consequently declines the accuracy of the recommender system.

Our approach tackled these issues; firstly, by exploiting domain taxonomies which provided relatively basic concepts of the domain, and secondly, by employing DBpedia which assisted in concept transformation and merging ontologies. Additionally, Section 5 illustrates that our reference ontology highly captures scholar's knowledge in terms of knowledge items, fit enough to Computer Science domain, encompassed purely domain relevant concepts, and contained rich ontological concepts.

Furthermore, the main contribution of our approach lies on integrating free taxonomies including ODP, ACM/IEEE, Wikipedia, and BOTW using DBpedia as a knowledge transformer. We developed a reference ontology suited for profiling the scholars' knowledge. Our approach is comprehensive and flexible enough to support the Computer Science domain, which fully captured the scholar's academic knowledge, exposed higher coverage, and specific to CS domain.

6.2. Future research and extensions

Our approach in creating the reference ontology is independent to the number of resources, and the coverage of scholars' knowledge is 85%. This flexibility and not-full coverage inspire the incorporation of more hierarchical taxonomies to the framework, which in turn, improves the ontology coverage and richness. However, the degree of heterogeneity among the hierarchies is an issue that should be addressed. As more hierarchies are included, the complexity and cost of merging and alignment of various ontological concepts will be increased substantially. This problem leads us to enhanced mediation approaches as well as automatic merging

process such as (Guzmán-Arenas & Cuevas, 2010) to address the large-scale ontology integration.

Another issue originates in the multi-disciplinary property of full researchers, as they participate in different and likely relevant research areas. These multi-disciplinary roles lead to divergent key terms, i.e., some key terms share in different topics with different weights. We will tackle this issue by performing fuzzy clustering (Bozkir & Sezer, 2013) on the knowledge items into relevant collections of key terms. Dealing with multi-state profiles in scholars' recommender system is also a new research path.

6.3. Conclusion

We analyzed four Web taxonomies in Computer Science domain including ODP, BOTW, Wikipedia, ACM/IEEE, and developed a riched reference ontology for profiling scholars' background knowledge. Wikipedia as an expert-agreement knowledge base enabled us to integrate the ontological concepts derived from the Web taxonomies. This study also made non-obvious connections between different taxonomies in the domain by semantically matching concepts from different taxonomy. The merging of ODP with other taxonomies enriched the domain concepts and facilitated cross-validation. To validate our work, five quality-based metrics and an application-based evaluation against our reference ontology is carried out. The result demonstrated that a broader range of ontological concepts compared to the golden ontologies are developed, which provided significant improvement to the state-of-the art of reference ontologies. Statistical evaluation also showed an improvement in terms of ontology coverage, richness, specificity, and completeness.

Acknowledgements

The authors would thank to Dr. Ahu Sieg, the Director of Research and Operations at Renkara Media Group, for her collaboration. This work is also supported by the Research Management Centre at the Universiti Teknologi Malaysia under the Research University Grant Scheme (Vote No. Q.J130000.2528.05H84).

References

Amini, B., Ibrahim, R., Othman, M. S., & Rastegari, H. (2011). Incorporating scholar's background knowledge into recommender system for digital libraries. In *5th Malaysian conference in software engineering (MySEC'11)* (pp. 516–523).

Amini, B., Ibrahim, R., & Othman, M. S. (2013). Data sets for offline evaluation of scholar's recommender system. *ACIIDS 2013, part II, LNAI 7803* (Vol. 2, pp. 158–167). Berlin/Heidelberg: Springer.

Amini, B., Ibrahim, R., Othman, M. S., & Selamat, A. (2014). Capturing scholar's knowledge from heterogeneous resources for profiling in recommender systems. *Expert Systems with Applications, 41*(17), 7945–7957.

Aquin, M., & Schlicht, A. (2009). Criteria and evaluation for ontology modularization techniques. In *Modular ontologies* (pp. 67–89). Springer.

Banerjee, S., & Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. *3rd International conference on computational linguistics and intelligent text processing (CICLing '02)* (Vol. 7, pp. 136–145). London: Springer.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., et al. (2009). DBpedia – A crystallization point for the web of data. *Web Semantics, 7*(3), 154–165.

Bozkir, a. S., & Sezer, E. A. (2013). FUAT – A fuzzy clustering analysis tool. *Expert Systems with Applications, 40*(3), 842–849. http://dx.doi.org/10.1016/j.eswa.2012.05.038.

Cassel, L., Clements, A., & Davies, G. (2008). Computer science curriculum 2008: An interim revision of CS 2001. *Security*, 37–108.

Chen, R.-C., Bau, C.-T., & Yeh, C.-J. (2011). Merging domain ontologies based on the wordnet system and fuzzy formal concept analysis techniques. *Applied Soft Computing, 11*(2), 1908–1923. http://dx.doi.org/10.1016/j.asoc.2010.06.007.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications, 36*(3), 5432–5435. http://dx.doi.org/10.1016/j.eswa.2008.06.054.

De Araujo, F. F., Lopes, F. L. R., & Loscio, B. F. (2010). MeMO: A clustering-based approach for merging multiple ontologies. In *2010 Workshops on database and expert systems applications* (pp. 176–180). IEEE. http://dx.doi.org/10.1109/DEXA.2010.50.

Degemmis, M., Lops, P., & Semeraro, G. (2006). Learning semantic user profiles from text. In *ADMA 2006, LNAI 4093* (pp. 661–672). Berlin/Heidelberg: Springer.

Dolog, P., & Nejdl, W. (2007). Semantic web technologies for the adaptiveweb. *The adaptiveweb, LNCS 4321* (pp. 697–719). Berlin Heidelberg: Springer.

Duong, T. H., Uddin, M. N., Li, D., & Jo, G. S. (2009). A collaborative ontology-based user profiles system. In *ICCCI 2009, LNAI 5796* (pp. 540–552). Berlin/Heidelberg: Springer.

Eric, B. (1992). A simple rule-based part of speech tagger. In *Applied natural language processing* (pp. 152–155).

Fareh, M., Boussaid, O., & Chalal, R. (2013). Merging ontology by semantic enrichment and combining similarity measures. *International Journal of Metadata, Semantics and Ontologies, 8*(1), 65–74.

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: The C-value/NC-value method. *International Journal on Digital Libraries, Springer, 3*(2), 115–130.

Gal, A., & Shvaiko, P. (2009). Advances in ontology matching. *Advances in Web Semantics I, LNCS 4891* (pp. 176–198). Berlin Heidelberg: Springer.

Gangemi, A., Steve, G., Giacomelli, F., Medica, R. I., & Biomediche, I. T. (1999). ONIONS: An ontological methodology for taxonomic knowledge integration. In *Data & knowledge engineering* (Vol. 31, pp. 183–220).

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2005). A theoretical framework for ontology evaluation and validation. In *2nd Italian semantic web workshop, CEUR workshop proceedings*. Trento, Italy.

Gauch, S., Speretta, M., Chandramouli, A., & Micarelli, A. (2007). User profiles for personalized information access. In *The adaptive web, LNCS 4321* (pp. 54–89).

Gauch, S., Speretta, M., & Pretschner, A. (2007). Ontology-based user profiles for personalized search. *Ontologies, integrated series in information systems* (Vol. 14, pp. 665–694). US: Springer. http://dx.doi.org/10.1007/978-0-387-37022-4_24.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarificatio. *Knowledge Acquisition*, 25–32.

Guzmán-Arenas, A., & Cuevas, A.-D. (2010). Knowledge accumulation through automatic merging of ontologies. *Expert Systems with Applications, 37*(3), 1991–2005. http://dx.doi.org/10.1016/j.eswa.2009.06.078.

Haldar, R., & Mukhopadhyay, D. (2011). Levenshtein distance technique in dictionary lookup methods: An improved approach. In *Web intelligence & distributed computing research lab*.

Henderson, L. (2009). Automated text classification in the DMOZ hierarchy, pp. 1–25.

Huang, L., Milne, D., Frank, E., & Witten, I. H. (2012). Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology, 63*(8), 1593–1608.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., & Hendler, J. (2006). Swoop: A web ontology editing browser. *Web Semantics: Science, Services and Agents on the World Wide Web, 4*(2), 144–153. http://dx.doi.org/10.1016/j.websem.2005.10.001.

Keet, C. M. (Marijke) (2004). *Aspects of ontology integration*. Scotland: Napier University.

Kodakateri, A., Gauch, S., Luong, H., & Eno, J. (2009). Conceptual recommender system for citeseerx. In *RecSys'09* (pp. 241–244). NY, USA: ACM.

Kotis, K., Vouros, G., & Stergiou, K. (2006). Towards automatic merging of domain ontologies: The HCONE-merge approach. *Web Semantics: Science, Services and Agents on the World Wide Web, 4*(1), 60–79. http://dx.doi.org/10.1016/j.websem.2005.09.004.

Kozakov, L., Park, Y., Fin, T., & Drissi, Y. (2004). Glossary extraction and utilization in the information search and delivery system for IBM Technical Support. *IBM Systems Journal, 43*(3), 546–563.

Lesk, M. (1987). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *SIGDOC'86 Proceedings of the 5th annual international conference on Systems documentation* (pp. 24–26).

Liang, T., Yang, Y., Chen, D., & Ku, Y. (2008). A semantic-expansion approach to personalized knowledge recommendation. *Decision Support Systems, 45*(3), 401–412. http://dx.doi.org/10.1016/j.dss.2007.05.004.

Liao, I., Hsu, W., Chen, M.-S., & Chen, L. (2010). A library recommender system based on a personal ontology model and collaborative filtering technique for English collections. *Emeral Group Publishing, 28*(3), 386–400. http://dx.doi.org/10.1108/02640471011051972.

Liao, S., Kao, K., Liao, I., & Chen, H. (2009). PORE: A personal ontology recommender system for digital libraries. *Emerald, 27*(3), 496–508. http://dx.doi.org/10.1108/02640470910966925.

Mahan, R. (2008). Best of the web. *Scientific American Mind, 19*, 84–85.

Manning, C., Raghavan, P., & Schütze, H. (2009). *An introduction to information retrieval*. Cambridge University Press. Online.

Marcus, M., Kim, G., Marcinkiewicz, M. A., Macintyre, R., Bies, A., Ferguson, M., & Schasberger, B. (1994). The Penn TREEBANK: Annotating predicate argument structure. In *ARPA human language technology workshop (HLT'94)* (pp. 114–119).

McGuinness, D. L., Fikes, R., Rice, J., & Wilder, S. (2000). An environment for merging and testing large ontologies. In *7th International conference on principles of knowledge representation and reasoning (KR2000)*. Colorado.

Medelyan, O., Milne, D., Legg, C., & Witten, I. H. (2009). Mining meaning from wikipedia. *Journal of Human Computer Studies, 67*(9), 716–754. http://dx.doi.org/10.1016/j.ijhcs.2009.05.004.

Mena, E. (2000). OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases, 8*(2), 223–271.

Mendes, P. N., Jakob, M., & Bizer, C. (2012). DBpedia: A multilingual cross-domain knowledge base. In *Proceedings of the eight international conference on language resources and evaluation LREC'12*, (pp. 1813–1817). Istanbul, Turkey.

Middleton, S. E., Roure, D. De, & Shadbolt, N. R. (2009). Ontology-based recommender systems. In *Handbook on ontologies, international handbooks on information systems* (pp. 779–796). http://dx.doi.org/10.1007/978-3-540-92673-3.

Mohammed, N., Duong, T. H., & Jo, G. S. (2010). Contextual information search based on ontological user profile. In *ICCCI 2010, LNAI 6422* (pp. 490–500).

Mohsenzadeh, M., Shams, F., & Teshnehlab, M. (2005). A new approach for merging ontologies. *World Academy of Science, Engineering and Technology*, 153–159.

Navigli, R. (2009). Word sense disambiguation. *ACM Computing Surveys, 41*(2), 1–69. http://dx.doi.org/10.1145/1459352.1459355.

Nguyen, N. T. (2006). A consensus-based approach for ontology integration. In *IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT 2006 workshops) (WI-IATW06)* (pp. 514–517).

Noy, N. F., & Musen, M. A. (1999). An algorithm for merging and aligning ontologies: Automation and tool support. In *Workshop on ontology management at the 16th national conference on artificial intelligence (AAAI-99)* (pp. 17–27). AAI Press.

Noy, N. F., Musen, M. A., & Informatics, S. M. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. In *17th National conference on artificial intelligence (AAAI-2000)*. Austin, Texas.

Obrst, L., Ashpole, B., Ceusters, W., Mani, I., & Smith, B. (2007). The evaluation of ontologies: Toward improved semantic interoperability. In *Semantic web: Revolutionizing knowledge discovery in the life sciences* (pp. 1–19).

Piao, S., Forth, J., Gacitua, R., Whittle, J., & Wiggins, G. (2010). Evaluating tools for automatic concept extraction: A case study from the musicology domain. In *Proceedings of the digital economy all hands meeting – Digital futures 2010 conference* (pp. 1–3). Nottingham, UK.

Raunich, S., & Rahm, E. (2011). ATOM: Automatic target-driven ontology merging. In *27th International conference on data engineering (ICDE)* (pp. 1276–1279). IEEE.

Salahli, M. A., Gasimzade, T. M., & Guliyev, A. I. (2009). *Domain specific ontology on computer science*. IEEE, pp. 3–5.

Salton, G., & Buckley, C. (1988). On the use of spreading activation methods in automatic information retrieval information retrieval. In *SIGIR '88 proceedings of the 11th annual international ACM SIGIR conference* (pp. 147–160).

Sánchez, D., Isern, D., & Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications, 39*, 7718–7728. http://dx.doi.org/10.1016/j.eswa.2012.01.082.

Schiaffino, S., & Amandi, A. (2009). Intelligent user profiling. *Artificial intelligence, LNAI 5640* (pp. 193–216). Berlin Heidelberg: Springer-Verlag.

Seidman, S., & McGettrick, A. (2008). Computer Science Curriculum 2008: An Interim Revision of CS 2001 Report from the Interim Review Task Force. Security.

Sharman, R., Kishore, R., & Ramesh, R. (2007). Ontologies: A handbook of principles, concepts and applications in information systems. In *Information systems*.

Shaw, M., Aho, A. V., & Bennett, C. H. (2004). *Computer science: Reflections on the field, reflections from the field. Computer*. Washington, D.C: The National Academies Press.

Sieg, A., & Burke, R. (2007). Web search personalization with ontological user profiles. In *CIKM'07* (pp. 525–534). Lisboa, Portugal: ACM.

Sieg, A., Mobasher, B., & Burke, R. (2007b). Ontological user profiles for personalized web search. In *AAAI worksop, intelligent techniques for web personalization (ITWP07)* (pp. 84–91).

Sieg, A., Mobasher, B., & Burke, R. (2007a). Learning ontology-based user profiles: A semantic approach to personalized web search. *IEEE Intelligent Informatics Bulletin, 8*(1), 7–18.

Sieg, A., Mobasher, B., & Burke, R. (2010). Improving the effectiveness of collaborative recommendation with ontology-based user profiles. *ACM*.

Strube, M., & Ponzetto, S. P. (2006). WikiRelate! Computing semantic relatedness using wikipedia. *American Association for Artificial Intelligence (AAAI)*, 1419–1424.

Sugiyama, K., & Kan, M. (2010). Scholarly paper recommendation via user's recent research interests. In *JCDL'10* (pp. 29–38).

Tartir, S., Arpinar, I. B., & Sheth, A. P. (2010). Ontological evaluation and validation. In R. Poli (Ed.), *Theory and applications of ontology: Computer applications* (pp. 115–130). Springer Science+Business Media.

Trajkova, J., & Gauch, S. (2004). Improving ontology-based user profiles. In *Recherche d'information assistee par ordinateur (RIAO '04)* (pp. 380–390).

Uchyigit, G. (2009). Semantically enhanced web personalization. *Web mining appl. in e-commerce & e-services, SCI 172* (pp. 25–43). Berlin Heidelberg: Springer-Verlag.

Velardi, P., Fabriani, P., & Missikoff, M. (2001). Using text processing techniques to automatically enrich a domain ontology. In *FOIS'01 proceedings of the international conference on formal ontology in information systems* (pp. 270–284).

Xinglin, L., Qilun, Z., Qianli, M., & Guli, L. (2012). Text similarity computing based on thematic term set. *International Journal of Advancements in Computing Technology (IJACT), 4*(April), 338–345. http://dx.doi.org/10.4156/ijact.vol4.issue6.39.

Yang, S.-Y., Hsu, C.-L., & Lu, S.-H. (2010). Developing an ontology-supported information recommending system for scholars. *Expert Systems with Applications, 37*(10), 223–228.

Yujie, Z., & Licai, W. (2010). Some challenges for context-aware recommender systems. In *5th international conference on computer science and education (ICCSE)* (pp. 362–365).

Zhang, Z., Brewster, C., & Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In *The sixth international conference on language resources and evaluation, (LREC 2008)*. Marrakech, Morocco.