IEICE TRANS. INF. & SYST., VOL.E103–D, NO.5 MAY 2020

PAPER Special Section on Data Engineering and Information Management

Estimating Knowledge Category Coverage by Courses Based on Centrality in Taxonomy

Yiling DAI[†a)], ***Nonmember***, **Masatoshi YOSHIKAWA**[†], ***and*** **Yasuhito ASANO**[†*], ***Members***

SUMMARY The proliferation of Massive Open Online Courses has made it a challenge for the user to select a proper course. We assume a situation in which the user has targeted on the knowledge defined by some knowledge categories. Then, knowing how much of the knowledge in the category is covered by the courses will be helpful in the course selection. In this study, we define a concept of knowledge category coverage and aim to estimate it in a semi-automatic manner. We first model the knowledge category and the course as a set of concepts, and then utilize a taxonomy and the idea of centrality to differentiate the importance of concepts. Finally, we obtain the coverage value by calculating how much of the concepts required in a knowledge category is also taught in a course. Compared with treating the concepts uniformly important, we found that our proposed method can effectively generate closer coverage values to the ground truth assigned by domain experts.

key words: *knowledge category coverage, course, taxonomy, centrality*

1. Introduction

The movement of providing Massive Open Online Courses (MOOCs) emerges from distance education and bursts into popularity in 2012. As is visioned, everyone should be able to access to the course materials on any subject, anywhere, and anytime. Albeit with the difficulty to realize this ideal condition, the current MOOC platforms have brought unprecedented mutual freedom to educators and learners. Belanger and Thornton [1] report that their first MOOC reached around 12,000 students, more than half of whom actually interacted with the course materials. Although only 313 students completed the course successfully, it is noteworthy that those students represent at least 37 different countries. It is hardly ever for an instructor in a brick-and-mortar university to reach students with such diverse backgrounds. Meanwhile, the learners are faced with various choices of courses offered by different institutions. For example, we have 56 choices on the subject of database in just one of the current MOOC platforms[**], which are designed and oriented under diverse educational purposes. As a result, it is undoubtedly a difficult task to select the proper course that satisfies one's learning need.

Categorization is an effective way to manage information. Taking "Database" for example, it is such a broad subject that we normally break it down into topics like "Relational Database", "Distributed Database", and "Data Mining", to name a few. With these topics, we can tackle the subject by focusing on one aspect of it at one time. In this study, we term the topics as knowledge categories. We presume the user of MOOC has already targeted on some knowledge categories, then it would be helpful if he/she knows how much the knowledge in the categories is covered by the courses. For example, suppose the user is interested in learning "Relational Database". As shown in Fig. 1, we can rank the courses based on the degree to which they cover the knowledge of "Relational Database". It is straightforward that Course A serves the user's need best since it covers the knowledge of this category with a highest percentage 92%. Additionally, if we are given the absolute amount of knowledge that is required in each category (i.e., the length of each hollow bar in Fig. 1), we can compare the course knowledge compositions as well. As shown in Fig. 1, we obtain an overall impression that Course A and B put an emphasis on "Relational Database" and touch some knowledge of "Distributed Database". In contrast, Course E teaches intensively the knowledge of "Distributed Database" and "Data Mining". In this study, we take the first step— estimating the knowledge category coverage as our goal.

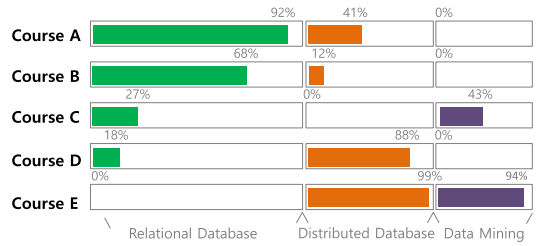

Fig. 1 An illustration of knowledge category coverage and course knowledge composition. The hollow bar indicates the total amount of required knowledge by the category. The colored bar and the percentage value represent how much of the knowledge in a category is covered by a course.

Analyzing the course content has been a research interest of education-related communities. Researchers [2]–[4] attempt to understand how the course content distributes over predefined knowledge categories[***]. As mentioned before, our goal in this study— estimating knowledge category

Manuscript received June 26, 2019.
Manuscript revised November 3, 2019.
Manuscript publicized December 26, 2019.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
[*]Presently, with the Faculty of Information Networking for Innovation and Design, Toyo University, Tokyo, 115–0053 Japan.
a) E-mail: daiyiling@db.soc.i.kyoto-u.ac.jp

DOI: 10.1587/transinf.2019DAP0002

[**]https://www.edx.org/course?search_query=database, accessed June 19, 2019.

[***]Though it may be called as "topic", "knowledge", "academic learning standard", or "knowledge area" in previous research, we unify them into the term "knowledge category" for consistency.

coverage— can be extended to acquire course knowledge composition. Other researchers [5]–[9] endeavour to gauge whether or to what extent a knowledge category is covered by course materials. However, either they employ a manual method or they don't define a concept of coverage. To the best of our knowledge, our study is the first to give a definition of knowledge category coverage and propose a semi-automatic[†] method to estimate it.

To estimate the knowledge coverage of a category by a course, we first model the knowledge category and the course as sets of concepts. Then, we define the coverage as the degree to which the concepts required in a knowledge category are also taught in a course. The key of estimating the coverage is to quantify the importance of concepts to the set, since the importance of the concepts is influenced by the existence of other concepts in the set. We resort to a taxonomy to capture the relationships among concepts and then utilize the idea of centrality to estimate how important a concept is to a set. When applying centrality to our method, we make a special effort to assign larger values to more important concepts without undervaluing less important concepts.

Compared with treating all the concept uniformly important, our centrality-based computation method produces closer coverage values to the ground truth assigned by domain experts. The main contributions of this study are twofold: 1) Our study is the first one to define a concept of knowledge category coverage and to estimate it in a semi-automatic manner. 2) We construct a taxonomy and utilize the idea of centrality to differentiate the importance of concepts in a set. Moreover, our method is elaborated to weight more important concepts without underestimating other concepts.

The rest of the paper proceeds as follows: In Sect. 2, we summarize related works on course content analysis, document relatedness estimation, and centrality in text processing. Then, we clarify our problem in Sect. 3 and propose our method in Sect. 4. In Sect. 5, the experiment procedures and results are reported. Section 6 presents our discussion on the results. Finally, we conclude our work and state the future work in Sect. 7.

2. Related Work

2.1 Course Content Analysis

The community of education has a long-standing interest in understanding how knowledge is organized and conveyed in academic programs and courses. Researchers have investigated whether the academic programs or courses fulfill the requirements established by domain experts, regardless of by manual or automatic methods. We separate these works into two groups based on what types of information they aim to extract.

The first group of research [2]–[4] focuses on how the course content distributes on a predefined set of knowledge categories. For instance, Bain et al. [2] manually scrutinize textbooks and count the pages spent on the knowledge categories in the domain of accounting information system. A statistical model is adopted in [4] to predict the distribution of computer science courses over some predefined knowledge categories. These works look at the composition of course content rather than the coverage of a knowledge category, which is the main difference with our work.

The second group of research attempts to understand whether and to what extent a knowledge category is covered by academic programs or individual courses. For example, Lennox and Diggens [5] interview the school staffs on whether their curricula touch on the ideal knowledge summarized by domain experts. Contractor et al. [6] tackle the problem of detecting the most related knowledge category for a given piece of course materials in an automatic manner. Both of these works only evaluate whether the knowledge category of interest is covered or not. Other research takes a further step to inspect the extent to which knowledge categories are covered. For instance, Macdonald and Fougere [7] use 5-point Likert scale to review how a textbook covers the categories about the subject of software piracy. Ishihata et al. [8] conduct a survey on how the informational science and engineering departments cover the core knowledge categories in this domain. They obtain the teaching hour of each department spending on each category from the questionnaire and then divide it by the required hour of the category to compute the coverage. What they achieved is close to our goal in this study, however, we address the problem in a semi-automatic way by processing the texts of courses and knowledge categories. Lastly, Kawintiranon et al. [9] utilize information retrieval techniques to estimate how a course is associated with a knowledge category. The association score they extract is actually the ratio of how many keywords in the knowledge category also appears in the course content. In this sense, it is similar to the concept of coverage in our study. However, their association score gets larger when the keyword appears more frequent in the course content. As a result, their association score does not strictly fall into the range of $[0, 1]$, which is different from what we attempt to estimate in this study.

2.2 Graph-Based Document Relatedness Estimation

Relatedness (or similarity) of documents is an important metric in information retrieval and it has received intensive attention. One stream of research in this area utilizes the knowledge graph to represent a document, thus the relatedness can be captured from the graphical perspective. It would seem that our work falls in a branch in this stream of research. However, our work is independent from those works for two reasons:

- What we aim to estimate, as called "coverage", is distinct from "relatedness". The relatedness of a document d_1 and another document d_2 derives from the re-

[†]We treat this method as semi-automatic for the reason that one step of the method— the construction of taxonomy is conducted in a manual process.

lated and the unrelated information of d_1 and d_2 (Refer to [10] for a detailed clarification.). Thus, if any of d_1 and d_2 contains more unique information, the relatedness become less. In contrast, the coverage of d_1 by d_2 is decided by the common information of d_1 and d_2, and all the information of d_1. In other words, no matter how much unique information d_2 contains, the coverage remains unchanged unless d_1 is unchanged.

- Theoretically, the coverage of d_1 by d_2 can be approximated by computing the relatedness of d_1 and $d_1 \cap d_2$ (common information of d_1 and d_2). However, previous methods [11]–[13] lack the quantification of the total information contained in a document, which is essential for estimating coverage. For example, Schuhmacher and Ponzetto [11] estimate the relatedness of d_1 and d_2 by inverting the cost of converting the graph of d_1 to the graph of d_2. However, this approach only captures what is unrelated (the cost to edit the differences of two graphs) but not what is related (the identical part of two graphs). Thus, it is unsuitable to estimate coverage in this type of approach.

2.3 Centrality in Text Processing

Centrality has been applied in text processing tasks mainly for document summarization [14]–[17] and other tasks such as keyword extraction [15], [18], topic identification [19], and term weighting [20] etc.

Some of these works use centrality score as a feature in further computation. For example, Xie [18] utilizes centrality measures as the features in a supervised model— decision tree to predict the noun phrases that should be included in the abstract of a document. Rousseau and Vazirgiannis [20] adopt centrality scores as term weights to represent a document, which is then used to retrieve the proper document for a query. Other works use the centrality score directly to select important sentences/words to represent a document [14]–[17], [19]. All of these works utilize degree centrality, which results in a sentence being considered important if it has a larger number of direct neighbors. In this study, we value the indirect connections between vertices as well. Therefore, we adopt another type of centrality and it will be further explained in Sect. 4.3.2.

The construction of the graph used to compute centrality plays a key role in applying centrality in such tasks. Some of the works [15], [18], [20] add edges based on the co-occurrences of sentences or phrases. This is built upon the assumption that a sentence can represent a document better if it appears together with more sentences. While in other works [14]–[16], an edge indicates two sentences are similar to each other. The meaning of edges is defined more specifically in [19] and [17]. Coursey and Mihalcea [19] model the relationship between two phrases if one is mentioned in the document of the other one. Rashidghalam et al. [17] adopt the relationships (e.g., derive, is-a, part-of, and related etc.) existing in the BabelNet ontology as the meanings of edges in their graph. Our definition of the edge is closer to the ones in the last two works and the details will be explained in Sect. 4.2.

Table 1 A part of the knowledge categories in CS2013.

KA	*KU*	*Topic*
Information Management	Information Management Concepts	· Information systems as socio-technical systems...
	Database Systems	· Approaches to and evolution of database systems...
	Data Modeling	· Data modeling...
	Indexing	· The impact of indices on query performance...
	Relational Databases	· Mapping conceptual schema to a relational schema...
	Query Languages	· Overview of database languages...
	Transaction Processing	· Transactions...
	Distributed Databases	· Distributed DBMS...
	Physical Database Design	· Storage and file structure...
	Data Mining	· Use of data mining...
	Information Storage and Retrieval	· Digital libraries...
	Multimedia Systems	· Standards (e.g., audio, graphics, video)...

3. Problem Formalization

3.1 Knowledge Category

Domain knowledge categorization is used as a reference to manage knowledge. With the diverse backgrounds of MOOCs, a standard domain knowledge categorization becomes especially helpful. According to our preliminary survey, there exist curriculum guidelines which attempt to categorize the knowledge that an academic program should include. Some examples are, "Curriculum Guidelines for Undergraduate Programs in Statistical Science" (by American Statistical Association) [†], "ASM Curriculum Guidelines for Undergraduate Microbiology" (by American Society for Microbiology) [††], "Computer Science Curricula 2013" (by ACM/IEEE-CS) [21], etc. Among these existing knowledge categorizations, we select "Computer Science Curricula 2013" (henceforth, CS2013) as an instance of the knowledge categorization for the reasons that: *a*) it covers a wide range of knowledge in the domain and organizes it into a category structure with more than one level; and *b*) the authors are more familiar with the domain of computer science. In CS2013, the knowledge is dubbed as *Topics*, and

[†]http://www.amstat.org/asa/education/Curriculum-Guidelines-for-Undergraduate-Programs-in-Statistical-Science.aspx, accessed June 11, 2019.

[††]https://www.asm.org/index.php/guidelines/curriculum-guidelines, accessed June 11, 2019.

then grouped into *Knowledge Units* (*KUs*) and *Knowledge Areas* (*KAs*). Table 1 shows the structure and some instances of knowledge categories in CS2013.

3.2 Problem Definition

In this study, we adopt the term "concept" to refer to a technical term, denoted as c. Since we use the course syllabus as a textual representation of the course content, we denote a course as s to avoid a duplicate notation with concept. Besides, we denote a knowledge category as k. Both of s and k are defined as a set of concepts. That is to say, given a syllabus s and a knowledge category k, we aim to estimate the ratio that the concepts required in k are covered by s, which is denoted as $cov(k|s)$.

4. Methodology

4.1 Intuition

Intuitively, the coverage of k by s can be captured in Eq. (1). With the denominator being the total knowledge that is required in k and the numerator being the knowledge that is both required in k and taught in s, the result provides us a ratio that can be comprehended as the knowledge coverage of k by s. Then, our goal is to estimate the two items in Eq. (1), namely, the required knowledge and the required and taught knowledge.

$$cov(k|s) = \frac{\text{The knowledge required in } k \text{ and also taught in } s}{\text{The knowledge required in } k} \quad (1)$$

Since we have already constrained k and s to the form of a concept set, we need to quantify how important the concepts are to the whole set. Suppose we have a concept set C_1 = {"Relational Model","Transaction Processing","Concurrency Control"} and a concept set C_2 = {"Relational Model","SQL","Relational Algebra"}. Although the concept "Relational Model" is required by both C_1 and C_2, it is not equally important to these sets. This is the underlying relationship among concepts that makes them behave differently when they are combined with different concepts. In this study, we model the concepts and their relationships as a taxonomy. Figure 2 demonstrates an example of the taxonomy of some database-related concepts. An edge indicates that the concept being pointed is a part of the other concept. As we can see, both "Relational Model" and "Transaction Processing" are important and relatively independent concepts in the domain of database. Therefore, it is likely that C_1 requires broader and shallower knowledge of "Relational Model". While in C_2, "SQL" and "Relational Algebra", two sub-concepts of "Relational Model" are also required, which indicates more concentrated and deeper knowledge of "Relational Model" is required.

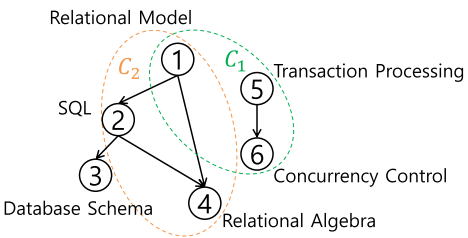

Fig. 2 An example of the taxonomy of concepts.

Based on the above intuition, we propose a method consisting of three steps— I) Taxonomy Construction, II) Concept Importance Computation, and III) Concept Importance Aggregation. Firstly, we construct a taxonomy which embeds the relationships among concepts. Then, we utilize the idea of centrality in the taxonomy to compute the importance of concepts to a concept set. When the concept importance is computed, we then simply aggregate the importance values of the concepts that are contained in k or s as the amount of required knowledge or taught knowledge, respectively. Figure 3 depicts the overall framework of this study and we will explain the main method by steps in the following sections.

4.2 Taxonomy Construction

This step corresponds to step I in Fig. 3, in which a taxonomy of concepts is constructed. We denote the taxonomy as a directed acyclic graph $G = <V, E>$, where V is a set of concepts and $E = \{(c_i, c_j)|$if learning c_j is necessary to understand $c_i\}$ is a set of directed edges. As mentioned in the example in Fig. 2, there should be an edge from "Relational Model" to "SQL", since it is inevitable to learn the knowledge of "SQL" to understand "Relational Model". We define the edge this way deliberately for the computation of concept importance and the reason will become clearer in the next section.

The quality of the taxonomy plays a significant role in the computation of the concept importance. Therefore, we are cautious to construct a taxonomy with reliable edges. Based on our definition of the edge, we consider the textbook is a valuable recourse to extract the relationships between two concepts. If we treat the chapter title as the concept to be explained, then the concepts being mentioned in

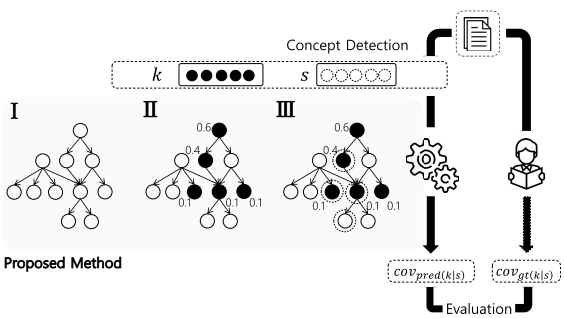

Fig. 3 The overall framework of this study. The white circles represent concepts and the directed edges indicate relationships between concepts. To separate the concepts appearing in different types of documents, filled circles are used for knowledge categories and dotted circles used for syllabi. $cov_{pred}(k|s)$ is the knowledge coverage of k by s estimated by our proposed method while $cov_{gt}(k|s)$ is the one assigned by domain experts.

the chapter can be treated as the necessary concepts in understanding the chapter title. When given a set of concepts and a set of textbooks, we propose Algorithm 1 to establish edges for those concepts.

Algorithm 1: Establish edges from textbooks

```
   Input  : V: set of c that is given, B = {b_1, b_2, ··· , b_n}: set of
            textbooks
   Output: E_1, E_2, ··· , E_n
1  E_1, E_2, ··· , E_n ⟵ ∅
2  for i ⟵ 1 to n do
3      for c ∈ V do
4          if c shows in the index of b_i then
5              P_c ⟵ the pages p where c appears
6              for p ∈ P_c do
7                  T ⟵ the titles t where p appears and the
                   corresponding levels of the titles l_t.   // l_t
                   is the level of t in the table of
                   content of the textbook.  A lower
                   level has a greater l_t value
8                  sort T based on l_t in descending order
9                  for t ∈ T do
10                     if t shows as c' ∈ C then
11                         E_i ⟵ E_i ⋃{(c', c)}
12                         break
```

In Algorithm 1, for every concept in the initial set, if it appears as a terminology in the index of the textbook, we then check whether the titles of the chapters where it appears are also concepts in the initial set. If so, an edge from the chapter concept to the index concept is established. Line 7 is a guarantee that when there exist multiple levels of chapters of a page, only the strongest edge (from the chapter concept at the lowest level) is included.

4.3 Concept Importance Computation

This step corresponds to step II in Fig. 3, where we firstly introduce two ways to compute the importance of the concepts (Sect. 4.3.1 and Sect. 4.3.2) and then put forward a method to combine the two ways (Sect. 4.3.3).

4.3.1 Uniform Computation Method

A naive way to compute the importance of the concepts in a set is to treat them uniformly important. We denote the importance of a concept c in a knowledge category k computed by the uniform computation method as $Imp_U(c|k)$, and it is generated as $Imp_U(c|k) = 1$.

4.3.2 Centrality-Based Computation Method

As we have discussed before, the concepts in a set are not uniformly important since they represent knowledge of different depths and widths. For instance, in the concept set C = {"Relational Model", "SQL","Relational Algebra"}, the knowledge of "SQL" and "Relational Algebra" contributes to the understanding of "Relational Model". Thus, it is possible that C requires profound knowledge of "Relational Model" with special interests in the knowledge of "SQL" and "Relational Algebra". How can we differentiate the importance of concepts? Recall our definition of the edges in the taxonomy— a concept has an edge to another concept if it can be better understood by learning the other concept. Therefore, we consider the concepts that have more access to other concepts in the taxonomy are more important in a set. This attribute is called "centrality" and used to find out the most "central" member in the context of social network analysis. The meaning of "central" depends on how it is defined in specific applications and different graph properties are used to compute centrality.

Based on our assumption that the importance of a concept is decided by the extend to which it could be understood by learning other concepts in the set, more important concepts should have more and intimate access to other concepts. Closeness centrality [22] serves our need in the sense that it treats a vertex as more central if it is closer to all the other vertices. A shorter total distance indicates more direct paths towards other concepts, thus, better understood and more important.

(1) The original closeness centrality.

A general equation used to compute the closeness centrality of v, $C(v)$, is shown in Eq. (2), where $dis(v, u)$ is the length of the shortest path from v to u, and $n-1$ is the possible smallest total distance of v to other vertices. Thus, $C(v)$ presents the inverse average distance of v towards other vertices, which is normalized to the range [0, 1].

$$C(v) = \left(\frac{\sum_{u \neq v} dis(v,u)}{n-1}\right)^{-1} = \frac{n-1}{\sum_{u \neq v} dis(v,u)} \tag{2}$$

(2) Dealing with disconnected graphs.

The original equation for computing closeness centrality is meaningless if the graph is not connected. Since all the total distance of a vertex towards other vertices becomes infinity even if there is only one vertex not connected to any vertices. This results in all the vertices having a centrality of zero, which underestimates the importance of the connected vertices. In the following, we explain some variants [23]–[26] to deal with disconnected graphs and how we choose the proper one to solve our problem.

(a) Large-value-replaced closeness centrality [23]. In this model, the distance to unreachable vertices are replaced by a large value instead of infinity. In Eq. (3), m in the second item in the denominator is the number of unreachable vertices of v and β is a parameter to modify this value (which is commonly set to the diameter of the graph). Note that the first item in the denominator only counts the distance to reachable vertices of v.

Table 2 An example of using variants of closeness centrality for disconnected graphs.

Concept	v	$C_{LV}(\beta = 6)$	C_H
"Relational Model"	1	0.31	0.50
"SQL"	2	0.25	0.40
"Database Shema"	3	0.17	0.00
"Relational Algebra"	4	0.17	0.00
"Transaction Processing"	5	0.20	0.20
"Concurrency Control"	6	0.17	0.00

$$C_{LV}(v) = \frac{n-1}{\sum_{u \neq v} dis(v,u) + m\beta} \tag{3}$$

(b) Harmonic closeness centrality [24], [25]. As shown in Eq. (4), this model computes the centrality of a vertex by summing up its inverse shortest distance to other vertices, which is then normalized by the possible maximum total inverse distance $(n-1)$. When there is no path from v to u, $dis(v,u) = \infty$, which results in a zero in the summation.

$$C_H(v) = \frac{1}{n-1} \sum_{u \neq v} \frac{1}{dis(v,u)} \tag{4}$$

(c) Components-based closeness centrality [26]. In this model, the centrality of vertices are computed independently inside each connected components, and then normalized by the relative size of this component to the whole graph. However, in each connected component, this model cannot cope with directed graphs that may still have disconnected pairs of vertices. For this reason, we exclude this variant from the candidate models to compute concept centrality.

Table 2 shows the closeness centrality values of the exemplar taxonomy (in Fig. 2) by using different models explained in (a) and (b). As we can see the distributions of two models, large-value-replaced closeness centrality tends to generate closer values for all vertices. In this example, we set β to the possible smallest large value, namely the number of vertices in the graph, as the replacement of infinity. If we enlarge the value of β, the values of all the vertices will get closer and closer, which is not desirable in our problem setting. On the contrary, harmonic closeness centrality succeeds to differentiate more important vertices (i.e., 1, 2, and 5) and less important vertices (i.e., 3, 4 etc.). Therefore, we adopt harmonic closeness centrality as our centrality-based method to compute the importance of concepts in a set.

4.3.3 Combined Computation Method

Since our taxonomy is directed, the leaf vertices will be underestimated during centrality computation (see the last column in Table 2). Although we consider leaf concepts are less important than the concepts in the upper levels, they should not be ignored completely. Thus, it is effective to combine the uniform computation method and centrality-based computation method to achieve a balanced importance of the concepts. We introduce a parameter to modify the trade-off between the differentiation of concept importance and the preservation of importance of less important concepts. To sum up, the importance of a concept to a set is computed by

$$Imp(c|k) = \alpha \cdot C_H(c|(k,G)) + (1-\alpha) \cdot Imp_U(c|k), \tag{5a}$$

$$C_H(c|(k,G)) = \frac{1}{|k|-1} \sum_{c' \in k \setminus c} \frac{1}{dis(c,c')}. \tag{5b}$$

Note that, when computing $dis(c,c')$, it may involve other concepts contained in the whole taxonomy G but not in k. And when $\alpha = 0$, it is identical to use uniform computation method, which is considered as our baseline. We tuned the value of α from $[0,1]$ in the experiment and found that the best performance appears when α is valued between $[0.85, 0.95]$.

4.4 Concept Importance Aggregation

In step II, we have computed the importance of a concept to a concept set. On one hand, the required knowledge of k can be obtained by summing up the importance values of its concepts. On the other hand, the required but also taught knowledge is simplified as the sum of the importance values of concepts that appear both in k and s. The ultimate equation used to compute the knowledge coverage of k by s is

$$cov(k|s) = \frac{\sum_{c \in k \cap s} Imp(c|k)}{\sum_{c \in k} Imp(c|k)}. \tag{6}$$

This step is notated as step III in Fig. 3.

5. Experiment

5.1 Dataset

There are 18 *KAs* in CS2013 and we only pick one *KA* to test the effectiveness of our proposed method. The reasons are:

- It is non-trivial to generate a taxonomy of domain knowledge correctly and automatically. To make sure the emphasis of this study falls on the step of computing concept importance, we adopt a balanced strategy to generate a reliable taxonomy which only allows us to implement it on a limited number of *KAs*.
- It is costy to generate ground truth for the dataset. We resort to domain experts to assign the knowledge coverage of *KUs* by courses. This requires the domain experts being considerably familiar with a domain. It is practically hard for us to reach domain experts across the extensive scope of computer science.

Among the 18 *KAs* defined in CS2013, we then chose "Information Management" as our preliminary dataset. Firstly, the authors are more familiar with this *KA*, which leads to a more insightful result analysis. Secondly, information management is viewed as a microcosm of computer

Table 3 Statistics of the documents of k and s.

	k	s
Number of documents	12	26
Average word count of the documents	50.42	253.96

science [27] and we think it is proper to choose a representative *KA* in this domain to try out our proposed method. Then, we collected 26 syllabi of courses that are related to information management. Among them, 7 courses are online courses and 19 are courses being provided in brick-and-mortar universities. Table 3 gives the basic information of the documents of k and s.

Regarding the ground truth, we asked two domain experts [†] to assign the knowledge coverage of all the pairs of k and s after reading the documents of 12 *KUs* and 26 courses. In detail, they were required to follow these instructions:

1. Read through the documents of 12 *KUs* to make sure you understand what knowledge is required. It may be helpful to form an image and keep in mind of what sub-topics you will teach and how much time you will spend in order to teach the required knowledge.
2. Read the course syllabi one by one and assign the percentage values while referring to your comprehension of the required knowledge. For the syllabi without explicit indications on how much time is spent on each topic in it, you may judge from the overall content of the course and estimate the volume of its content by treating it as a regular one-semester course.
3. Adjust the coverage values you have assigned to make sure they are judged under the same criterion whenever necessary.

The correlation coefficient of the coverage values collected from two experts is 0.855 ($p < 0.0005$). Therefore, We consider their assignment as reliable and then took the average coverage values as the final ground truth.

5.2 Concept Detection

Our proposed method is based on the assumption that the knowledge categories and course syllabi are given as sets of concepts. Therefore, the concept detection is a preprocessing of the documents we have collected. Specifically, we adopt several existing tools to detect the concepts that are defined in the knowledge base Wikipedia. Wikipedia is an online encyclopedia that can be edited and updated by massive users. It has become a valuable knowledge resource for tasks in various fields such as information retrieval, knowledge engineering, and natural language processing, to name a few. For a given document, Wikification is a process in which the phrases and their corresponding Wikipedia articles are detected and extracted. In this study, we treat the titles of Wikipedia articles as concepts and adopt four Wikifiers to convert documents of k and s into concept sets. Among the four Wikifiers, one is our original tool[††] and the other three are developed in previous research [28]–[30]. We then took the union of the detected concepts by all the four Wikifiers as the final concept sets[†††]. We accept the detected Wikipedia articles as the given sets of concepts. No special process is conducted to find potentially missing concepts since it is beyond the scope of this study. Table 4 reports the numbers of concepts detected in the process of Wikification.

[†]One of them is the author of this paper, and the other one is not.

Table 4 Statistics of the Wikification results of k and s.

	k	s
Average number of concepts	25.83	98.54
Total number of concepts	264	1436

Table 5 Evaluation on the Wikification results.

	# of unrelated concepts	Average centrality of unrelated concepts
IM01	6	0.000
IM03	2	0.000
IM04	8	0.024
IM06	1	0.000
IM07	3	0.000
IM10	4	0.000

There remains a concern that including multiple Wikifiers may increase the noisy concepts that are not actually related to this document. However, we expect the negative effect of these concepts can be alleviated when projecting them on the taxonomy. To verify this, we randomly selected six of the *KUs* and asked two evaluators (both of them are PhD candidates and one is the first author.) to check whether the detected Wikipedia articles are related to the document or not. Three levels were used to rate a Wikipedia article—related, somewhat related, and not related. We then computed the average centrality values of the Wikipedia articles that are considered as not related by at least one of the evaluators. Table 5 reports that the noisy Wikipedia articles get rather low centrality values, which implies that the centrality computation succeeds to suppress the importance of wrongly detected Wikipedia articles.

5.3 Taxonomy Construction

In the experiment, we set the union of the concepts detected in all the knowledge categories as V, and followed Algorithm 1 to extract $E = E_1 \cup E_2 \cup E_3$ manually from three classic textbooks [27], [31], [32] in the domain of information management. Note that we include all the edges that can be found in at least one of the textbooks, since the number of valid edges decreases dramatically if we raise the threshold

[††]We first use NLTK package to extract noun phrases from the documents. Then, the noun phrases are used as query to search related Wikipedia articles in the Bing search engine.

[†††]In preliminary experiments, we tried to use the number of wikifiers by which a concept is detected as an indicator of the reliability of the concept. However, the performance didn't improve significantly. Therefore, we do not include this factor in this study.

to two. We also removed one edge that causes a cycle in the taxonomy[†]. Consequently, we obtained a taxonomy of 264 vertices and 245 edges.

5.4 Evaluation Framework

In this study, our main concern is how the coverage values can help a user to make an informed decision of learning. Therefore, the predicted $cov(k|s)$ value should be as close to the ground truth as possible. We evaluate the result in two scales.

5.4.1 Evaluating All Pairs

In this evaluation scale, we require every individual $cov(k|s)$ being comparable to each other. That is to say, we evaluate whether the $cov(k|s)$ values of all pairs of *KUs* and courses are in consistent with the ones of the ground truth. Two metrics are adopted:

- **Pearson Correlation Coefficient (Pearson, thereafter)** [33] is used to reflect whether the predicted coverage values is "propotional" to the ones of the ground truth. We denote the set of knowledge categories and the set of syllabi as K and S, respectively. The ground truth of the coverage of the i^{th} k by the j^{th} s is denoted as $cov_{gt}^{(i,j)}$ and the prediction of our proposed method as $cov_{pred}^{(i,j)}$. Then, Pearson value is computed as

$$P = \frac{\sum(cov_{gt}^{(i,j)} - cov_{gt})(cov_{pred}^{(i,j)} - cov_{pred})}{\sqrt{\sum(cov_{gt}^{(i,j)} - cov_{gt})^2 \sum(cov_{pred}^{(i,j)} - cov_{pred})^2}}, \tag{7}$$

  where cov_{gt} is the average value of $cov_{gt}^{(i,j)}$, and cov_{pred} is the average value of $cov_{pred}^{(i,j)}$.
- **Mean Squared Error (MSE, thereafter)** is used to check whether the predicted coverage values have a small deviation from the ground truth. As shown in the following equation, the errors that have a larger difference from the ground truth get larger penalties.

$$MSE = \frac{\sum(cov_{pred}^{(i,j)} - cov_{gt}^{(i,j)})^2}{|K||S|} \tag{8}$$

5.4.2 Evaluating by Course

In this evaluation scale, we focus on the coverage estimation inside every course. This is driven by the consideration that syllabi may be written in different styles or levels of detailedness even the courses cover a *KU* to a similar degree. We treat a syllabus as a vector in the space of its knowledge coverage with the *KUs*, which is denoted as $\overrightarrow{cov_s(K)} =< cov(k_1|s), \ldots, cov(k_n|s) >$, where $n = |K|$. Then we compute the cosine similarity between the predicted $\overrightarrow{cov_s(K)}$ and the ground truth.

[†]One cycle (Transaction processing⇌Concurrency control) is found in our dataset. Since coping with the cycles is not essential in this study, we simply removed the edge from Concurrency control to Transaction processing to avoid the cycle.

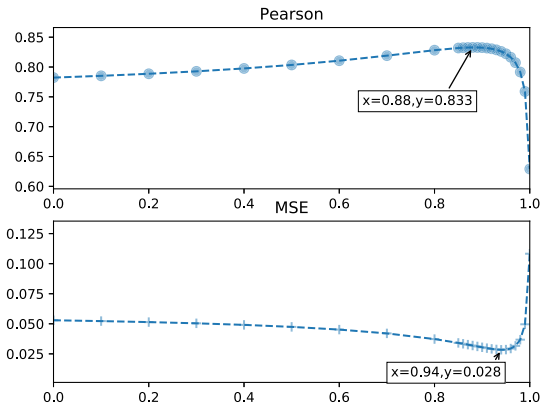

Fig. 4 The results of evaluating all pairs in scatter plots. The x axis represents the values of α and the y axis represent Pearson and MSE values, respectively. The y value in the box is the maximum Pearson or minimum MSE value, and x value is its corresponding α value.

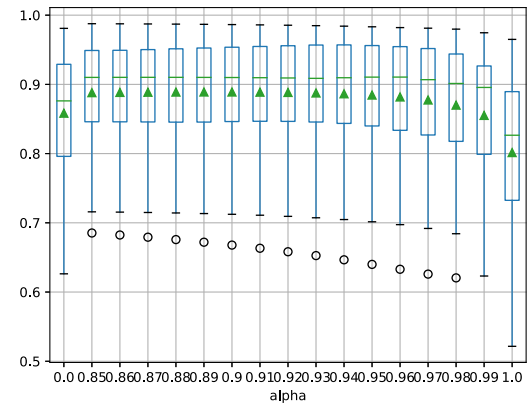

Fig. 5 The result of evaluating by course. The x axis represents the values of α and the y axis represents the cosine similarity values.

5.5 Results

5.5.1 Evaluating All Pairs

The only parameter in our method is α, which indicates the extent to which the harmonic closeness centrality is utilized in the computation of concept importance. Figure 4 is the scatter plot of the α values and their corresponding Pearson and MSE values. $\alpha = 0$ represents the baseline experiment in which uniform importance values of concepts in k and s are used to compute the coverage. When inspecting the α values other than zero, both Pearson and MSE values reach a peak at some relatively high value and then drop dramatically when α equals 1. In detail, Pearson reaches its peak when α is valued of 0.88, and MSE reaches its peak when α is valued of 0.94. Overall, the method performs best on both evaluation metrics when α is valued in the range $[0.88, 0.94]$. This proves that the idea of using centrality to

compute the importance of a concept is valid in the estimation of knowledge category coverage. Moreover, the combination of uniform computation method and centrality-based computation method plays a significant role in applying the idea of centrality to solving our problem.

5.5.2 Evaluating by Course

In Fig. 5, we plot the box-plots of the cosine similarity values of 26 courses for the experiments with α valued in $[0, 0.85 - 0.99, 1]$. As can be observed, the mean value (represented in green triangles in the figure) of cosine similarity has an obvious rise from the baseline method ($\alpha = 0$), starts to fall from where $\alpha = 0.94$, and finally drops when $\alpha = 1$. Similar trends can be found on median, the first-quartile, the third-quartile, the minimum and the maximum values of cosine similarity. We conclude that our proposed method is valid to estimate the knowledge category coverage for a course when choosing the appropriate parameter to combine the uniform and centrality-based computation methods.

6. Discussion

In this section, we investigate on the result in more depth. Specifically, we check over the most important concepts of each *KU* and the performance on different *KUs*. In Table 6, we list the five most important concepts with their importance values when $\alpha = 0.88$. Regarding Table 7, we treat a *KU* as a vector in the space of its knowledge coverage by the syllabi and then compute the cosine similarity of the

Table 6 The five most important concepts of *KUs* ($\alpha = 0.88$).

KU1: Information Management Concepts		*KU2*: Database Systems		*KU3*: Data Modeling	
Database	0.208	Database management system	0.419	Relational database	0.306
Computer data storage	0.148	Database	0.403	XML	0.296
Information	0.120	Database transaction	0.183	Data model	0.237
Sociotechnical system	0.120	Transaction processing	0.167	Data modeling	0.237
Information retrieval	0.120	Relational database management system	0.151	Semi structured data	0.237
KU4: Indexing		*KU5*: Relational Databases		*KU6*: Query Languages	
Computer data storage	0.289	Relational model	0.513	SQL	0.503
Database	0.234	Database design	0.472	Database	0.495
SQL	0.169	Database normalization	0.419	Relational database	0.417
Database index	0.149	Relational database	0.325	Query language	0.291
Index database	0.149	Data integrity	0.261	Stored procedure	0.169
KU7: Transaction Processing		*KU8*: Distributed Databases		*KU9*: Physical Database Design	
Database transaction	0.462	Database	0.384	Computer data storage	0.357
Transaction processing	0.413	Distributed database	0.294	Database index	0.323
Computer data storage	0.364	Parallel database	0.266	Index (database)	0.323
Concurrency control	0.218	Query optimization	0.221	Database	0.289
Data buffer	0.120	Computer data storage	0.180	Database design	0.120
KU10: Data Mining		*KU11*: Information Storage and Retrieval		*KU12*: Multimedia Systems	
Data mining	0.309	Information storage and retrieval	0.153	Information retrieval	0.120
Data	0.120	Document management system	0.136	Digital library	0.120
Interactive visualization	0.120	Inverted index	0.136	User interface	0.120
Cluster analysis	0.120	Information retrieval	0.133	Data compression	0.120
Algorithm	0.120	Information security	0.133	Multimedia	0.120

Table 7 The result of evaluating by *KUs*.

α	*KU1*	*KU2*	*KU3*	*KU4*	*KU5*	*KU6*	*KU7*	*KU8*	*KU9*	*KU10*	*KU11*	*KU12*
0.0	0.753	0.907	0.971	0.808	0.953	0.945	0.926	0.887	0.885	0.657	0.793	0.627
0.1	0.753	0.907	0.972	0.809	0.953	0.946	0.927	0.887	0.886	0.657	0.793	0.627
0.2	0.753	0.908	0.972	0.809	0.954	0.948	0.929	0.886	0.887	0.657	0.793	0.627
0.3	0.752	0.908	0.972	0.809	0.955	0.950	0.930	0.886	0.888	0.658	0.793	0.627
0.4	0.752	0.908	0.972	0.810	0.956	0.952	0.931	0.885	0.890	0.659	0.794	0.627
0.5	0.751	0.908	0.972	0.810	0.957	0.955	0.933	0.884	0.891	0.660	0.794	0.627
0.6	0.750	0.909	0.972	0.811	0.959	0.958	0.935	0.883	0.893	0.660	0.794	0.627
0.7	0.749	0.909	0.972	0.812	0.960	0.961	0.936	0.880	0.895	0.661	0.794	0.627
0.8	0.746	0.910	0.972	0.814	0.962	0.966	0.938	0.875	0.898	0.661	0.794	0.627
0.9	0.736	0.910	0.971	0.815	0.962	0.970	0.937	0.862	0.898	0.647	0.795	0.627
1.0	0.647	0.895	0.956	0.803	0.956	0.972	0.928	0.806	0.875	0.411	0.726	0.000

In each column, the cell is colored based on which range the difference of its value with the one of the baseline method ($\alpha = 0$) falls in.

−1.000	−0.100	−0.075	−0.050	−0.025	−0.010	0.000	+0.010	+0.025	+0.050	+0.075	+0.100	+1.000

predicted one and the ground truth.

First of all, we find the most important concepts are representative of the *KUs*. A quick verification is to check whether we can infer the main topic of the *KU* merely from the important concepts without knowing the title of this *KU*. As shown in Table 6, it is easy to judge that *KU5* is about relational database design and *KU7* requires the knowledge of transaction processing.

On the other hand, we find that our method is weak to the knowledge categories that contain "isolated" or "general" concepts. For example, the proposed method (when $\alpha \neq 0$) is not working for *KU12* (see Table 7). A potential reason is that this *KU* contains relatively new and inter-disciplinary concepts that are underpresent in classic textbooks. Thus, these concepts get low importance values since they are "isolated" in the taxonomy. Another example is the comparison of *KU1* and *KU10*. Both of the *KUs* contain a very limited number of important concepts (i.e., Database, Computer data storage in *KU1* and Data mining in *KU10*). However, if we compare the performance on these two *KUs* in Table 7, it can be seen that utilizing centrality-based computation method more is decreasing the accuracy of *KU1* while increasing the accuracy (when α is in the range of $[0, 0.8]$) of *KU10*. We analyze this is caused by the different generality of concepts in the domain. That is to say, Database and Computer data storage tend to appear in various *KUs* and they are not that important to *KU1*, while Data mining is unique to *KU10* and it deserves to be highly valued. Although our method is able to differentiate the importance of a concept in different concept sets, further consideration on how to modify the importance values of a "general" concept is needed.

7. Conclusion and Future Work

In this study, we firstly define the knowledge coverage of a knowledge category by a course as the extent to which the knowledge required in the category is also taught in the course. Then, we propose a centrality-based computation method to estimate the concept importance to the knowledge categories, which is then aggregated to estimate the knowledge coverage. The experiment has shown that our method can generate closer knowledge coverage values to the ground truth assigned by human experts, compared to the uniform computation method.

Some future challenges remain. We only experiment on one *KA* in this study, and we expect to extend it to other areas in the domain of computer science once we have upgraded the technique to build a broader taxonomy. In the method, we did not carry out any technique to deal with general concepts that appears in multiple concept sets. It is an interesting task to consider how to modify the importance values of a concept to different sets based on what other concepts are contained in the sets.

References

[1] Y. Belanger and J. Thornton, "Bioelectricity: A quantitative approach duke university's first mooc," https://hdl.handle.net/10161/6216, accessed June 19, 2019.
[2] C.E. Bain, A.I. Blankley, and L.M. Smith, "An examination of topical coverage for the first accounting information systems course," Journal of Information Systems, vol.16, no.2, pp.143–164, 2002.
[3] P. Denny, A. Luxton-Reilly, J. Hamer, and H. Purchase, "Coverage of course topics in a student generated mcq repository," Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE '09, New York, NY, USA, pp.11–15, ACM, 2009.
[4] T. Sekiya, Y. Matsuda, and K. Yamaguchi, "Curriculum analysis of cs departments based on cs2013 by simplified, supervised lda," Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, LAK '15, New York, NY, USA, pp.330–339, ACM, 2015.
[5] N. Lennox and J. Diggens, "Knowledge, skills and attitudes: Medical schools' coverage of an ideal curriculum on intellectual disability," Journal of Intellectual & Developmental Disability, vol.24, no.4, pp.341–347, 1999.
[6] D. Contractor, K. Popat, S. Ikbal, S. Negi, B. Sengupta, and M.K. Mohania, "Labeling educational content with academic learning standards," Proceedings of the 2015 SIAM International Conference on Data Mining, pp.136–144, 2015.
[7] L.E. Macdonald and K.T. Fougere, "Software piracy: A study of the extent of coverage in introductory mis textbooks," Journal of Information Systems Education, pp.325–329, 2003.
[8] K. Ishihata, H. Ohiwa, H. Kakuda, K. Shimizu, T. Tamai, H. Nagasaki, H. Nakazato, T. Nakatani, T. Hikita, T. Miura, T. Minohara, K. Wada, and O. Watanabe, "Investigation on the Educational Contents among Informational Science and Engineering Departments by Using Syllabus (Intermediate Report)," tech. rep., Information Processing Society of Japan, 2010.
[9] K. Kawintiranon, P. Vateekul, A. Suchato, and P. Punyabukkana, "Understanding knowledge areas in curriculum through text mining from course materials," 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), pp.161–168, Dec. 2016.
[10] D. Lin, "An information-theoretic definition of similarity," Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, San Francisco, CA, USA, pp.296–304, Morgan Kaufmann Publishers Inc., 1998.
[11] M. Schuhmacher and S.P. Ponzetto, "Knowledge-based graph document modeling," Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14, New York, NY, USA, pp.543–552, ACM, 2014.
[12] C. Paul, A. Rettinger, A. Mogadala, C.A. Knoblock, and P. Szekely, "Efficient graph-based document similarity," The Semantic Web. Latest Advances and New Domains, Cham, vol.9678, pp.334–349, Springer International Publishing, May 2016.
[13] Y. Ni, Q.K. Xu, F. Cao, Y. Mass, D. Sheinwald, H.J. Zhu, and S.S. Cao, "Semantic documents relatedness using concept graph representation," Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM '16, New York, NY, USA, pp.635–644, ACM, 2016.
[14] G. Erkan and D.R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," J. Artif. Int. Res., vol.22, no.1, pp.457–479, Dec. 2004.
[15] X. Wan and J. Xiao, "Exploiting neighborhood knowledge for single document summarization and keyphrase extraction," ACM Trans. Inf. Syst., vol.28, no.2, pp.8:1–8:34, June 2010.
[16] D. Parveen and M. Strube, "Integrating importance, non-redundancy and coherence in graph-based extractive summarization," Proceedings of the 24th International Conference on Artificial Intelligence,

IJCAI'15, pp.1298–1304, AAAI Press, 2015.
[17] H. Rashidghalam, M. Taherkhani, and F. Mahmoudi, "Text summarization using concept graph and BabelNet knowledge base," 2016 Artificial Intelligence and Robotics (IRANOPEN), pp.115–119, April 2016.
[18] Z. Xie, "Centrality measures in text mining: Prediction of noun phrases that appear in abstracts," Proceedings of the ACL Student Research Workshop, ACLstudent '05, Stroudsburg, PA, USA, pp.103–108, Association for Computational Linguistics, 2005.
[19] K. Coursey and R. Mihalcea, "Topic identification using wikipedia graph centrality," Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, NAACL-Short '09, Stroudsburg, PA, USA, pp.117–120, Association for Computational Linguistics, 2009.
[20] F. Rousseau and M. Vazirgiannis, "Graph-of-word and tw-idf: New approach to ad hoc ir," Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM '13, New York, NY, USA, pp.59–68, ACM, 2013.
[21] ACM and IEEE, Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, ACM, New York, NY, USA, 2013. 999133.
[22] L.C. Freeman, "Centrality in social networks conceptual clarification," Social Networks, vol.1, no.3, pp.215–239, 1978.
[23] G. Csardi and T. Nepusz, "The igraph software package for complex network research," InterJournal, Complex Systems, vol.1695, no.5, pp.1–9, 2006.
[24] M. Newman, "The structure and function of complex networks," SIAM Review, vol.45, no.2, pp.167–256, 2003.
[25] C.T. Butts, "Social network analysis with sna," Journal of Statistical Software, vol.24, no.6, pp.1–51, 2008.
[26] S. Wasserman and K. Faust, "Social Network Analysis: Methods and Applications," Structural Analysis in the Social Sciences, Cambridge University Press, 1994.
[27] R. Ramakrishnan and J. Gehrke, Database management systems, McGraw Hill, New York, 2000.
[28] P. Ferragina and U. Scaiella, "Tagme: On-the-fly annotation of short text fragments (by wikipedia entities)," Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, New York, NY, USA, pp.1625–1628, ACM, 2010.
[29] J. Daiber, M. Jakob, C. Hokamp, and P.N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," Proceedings of the 9th International Conference on Semantic Systems, I-SEMANTICS '13, New York, NY, USA, pp.121–124, ACM, 2013.
[30] J. Brank, G. Leban, and M. Grobelnik, "Annotating documents with relevant Wikipedia concepts," Proceedings of the Slovenian Conference on Data Mining and Data Warehouses (SikDD 2017), Ljubljana, Slovenia, 2017.
[31] C. Date, An Introduction to Database Systems, 8 ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
[32] H. Garcia-Molina, J.D. Ullman, and J. Widom, Database systems: the complete book, 2nd ed., Pearson Education, Inc., New Jersey, 2008.
[33] K. Pearson and O.M.F.E. Henrici, "Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia," Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, vol.187, pp.253–318, 1896.

Yiling Dai received the BS degree from the School of Management, Zhejiang University, Zhejiang, China, in 2012, and the MS degree from the Graduate School of Business, Rikkyo University, Tokyo, Japan, in 2015. Since April 2015, she has been a PhD candidate in the Graduate School of Informatics, Kyoto University. Her research interests include information retrieval and knowledge engineering, especially applied to educational scenarios.

Masatoshi Yoshikawa received his BE, ME, and Dr. Eng degrees from the Department of Information Science, Kyoto University, in 1980, 1982, and 1985, respectively. From 1985 to 1993, he was with Kyoto Sangyo University. In 1993, he joined the Nara Institute of Science and Technology as an associate professor in the Graduate School of Information Science. From June 2002 to March 2006, he served as a professor at Nagoya University. Since April 2006, he has been a professor at Kyoto University. His general research interests are in the area of databases. His current research interest includes privacy protection technologies, personal data market, and multi-user routing algorithms and services. He is a member of the ACM and IPSJ.

Yasuhito Asano received the BS, MS, and DS degrees in information science from the University of Tokyo in 1998, 2000, and 2003 respectively. In 2003-2005, he was a research associate in the Graduate School of Information Sciences, Tohoku University. In 2006-2007, he was an assistant professor in the Department of Information Sciences, Tokyo Denki University. He joined Kyoto University in 2008 as an assistant professor. In 2009-2018, he was an associate professor in the Graduate School of Informatics, Kyoto University. Since 2019, he has been a professor at Faculty of Information Networking for Innovation and Design, Toyo University. His research interests include web mining and network algorithms. He is a member of the IEICE, IPSJ, DBSJ and OR Soc. Japan.