LCRec: Learning Content Recommendation (Wiki-based Skill Book)

Chalothon Chootong[1], Timothy K. Shih[1], Ankhtuya Ochirbat[2], Worapot Sommool[1], W. K. T. M. Gunarathne[1], Carl K. Chang[3]

[1] Department of Computer Science and Information Engineering, National Central University, Taiwan
[2] Department of Information and Computer Science, National University of Mongolia, Mongolia
[3] Department of Computer Science, Iowa State University, USA
chalothon.cs@gmail.com, timothykshih@gmail.com, title.wo@gmail.com, tharanga.gunarathne @gmail.com, ankhaa8@gmail.com, chang@iastate.edu

Abstract

Knowledge skills in the ICT-industry always evolve. With the vast variety of jobs available, it is unlikely to educate students with skills to fit every job-requirement. This issue inspired us to develop the Learning Content Recommender (LCRec) for students to find appropriate learning contents based on required job-skills. In order to bridge the required skill for industry and academia, we have to work on IT job-skills and the Computer Science Curriculum 2013 (CS2013). Skills from 48 publicly available job searching websites are used to investigate what the industry needs. We carried out experiments among professionals, academics, and students to test the usefulness of LCRec, and evaluated the feedbacks. LCRec successfully used Knowledge Units from CS2013, Wikipedia, and essential skills from job hunting websites, to benefit entry-level job seekers for finding necessary learning contents to study. It is also convenient for academics to look at the skills needed in industries, and to consider enhancing the curriculum with new skills. The study result demonstrated that it is possible to bridge the gap (what learning contents are lacking) between the academia and the industry.

Keywords: Learning content recommendation, Knowledge management, Auto generated contents, Wikipedia, Computer Science Curriculum 2013

1 Introduction

The software development industry is rapidly growing every year with new technologies and emerging innovations. As a result, employees who have been in the Information Communications Technology (ICT) industry regularly need to gain new skills and to be updated with new technologies. With the growing knowledge and a vast variety of different jobs available to graduates, it is not possible to educate students with all necessary skills that they will need for different potential jobs. However, one of the main duties for universities is to prepare students for their future careers, and to ensure that they are equipped with the knowledge and skills necessary to succeed after graduation.

According to recent research findings, there exists a huge gap between what is being taught in the university and what industry actually needs in terms of professional skills and talents [1]. Computer Science, information systems and IT students do not always have the necessary technical capabilities, personal skills and knowledge to succeed in their careers after graduation [2-3]. This deficiency indicates a gap between the abilities of graduates and industry expectations.

In order to bridge the gap between graduate competency and industry expectations, universities need to work more closely with industries and place required skills and knowledge in the curriculum. In a university system, revising a curriculum according to the industry requirements would not be an easy task. Therefore, it could be a useful service for instructors to indicate a specific job, and the system can discover a list of required skills that are related to that job. The university instructors can add additional knowledge or materials that related to the job-skills to their syllabus. In turn, students graduating from a computing curriculum can employ the same services to match the skill set with a career goal.

The main goal of our research is to construct the Learning Content Recommender (LCRec) system for both instructor and students to discover appropriate learning contents, based on the intended job criteria. First, LCRec helps the user to search for an intended job and provides the results as two type of functions: from job to skills and from skill to jobs. The first function provides the mapping between a job position and the related skills that were extracted from job

*Corresponding Author: Chalothon Chootong; E-mail: chalothon.cs@gmail.com
DOI: 10.3966/160792642019102006007

descriptions. The second inform users whether particular skills can be applied to a set of job positions. Finally, LCRec produces a collection of learning contents analyzing the correlations between job-skills and Knowledge Units (KUs) from Computer Science Curriculum 2013 (CS2013), a recent update of Computing Curricula 2001 (CC2001) [4]. Our on-line system and instruction are available at http://brs.minelab.tw. Please visit this website to try the system. The demonstration of functionality, how to use the system, and the analysis of test outcomes are available at http://brs.minelab.tw/about/demo.html.

The rest of the paper is organized as follows. Section 2 presents related. In Section 3, we introduce the proposed system, methodologies, and system functionalities. Section 4 provides experiments and evaluations. In Section 5, we compare our approach with Wikibooks. Discussion and conclusion are presented in Section 6 and 7, respectively.

2 Related Studies

2.1 Learning Content Recommendation

Educational recommender systems should deliver learning materials to students in a format that best suits an individual student's personal preference and learning experience. Recommending learning contents should be pedagogically guided by educational objectives, and not only by learners' preferences [5, 16]. Learners achieve different levels of capabilities, which have various levels in different learning domains. Therefore, it is important to identify the relevant learning goals and use a system to support learners.

Building a learning content recommendation system for job-skills in the fields of ICT is different compared to other domains. Fields in ICT rapidly change with the new technologies and emerging innovations. In order to catch up the new changes, our proposed system recommends learning contents based on the most recent required skills, which were extracted from the latest job posts on the Internet that we did not found this kind of work in the existed research.

Tang and McCalla [7] introduced a smart recommendation for an e-learning system by pointing out the differences of making recommendations in e-learning and other domains. They proposed two pedagogy features in the recommendation: learner interest and background knowledge. Mödritscher [8] introduced recommended strategies for supporting collaborative activities in the personal learning environment (PLE) settings. The author characterized collaborative activities in such situations and built a model of learning activities in order to capture learner interactions with the environment and to generate recommendations. Another e-Learning personalization system was developed based on hybrid recommendation strategy and learning style identification [10]. In this work, their personalized e-learning system can automatically adapt to the interests, habits and knowledge levels of learners. Kanetkar et al. [11] proposed a hybrid approach and implemented a prototype system in the course repository of the Virtual University of Tunis. Castillo-Carrero et al. [12] implemented a prototype of learning content recommendation system based on ClipIt, using social networks in education.

2.2 Book Recommendation

Numerous research studies were carried out in the area of recommending books to users based on their interest. De Clercq et al. [17] investigated on extracting and adding semantic features based on Linked Open Data to a content-based book recommender. Chen et al. [18] applied item-based collaborative filtering approach to analyzing the users' behavior on a large book loan data of a university library system to recommend books. According to the research studies of [19], and [20], authors introduced a book recommendation systems based on combined features of content-based filtering, collaborative filtering and association rule mining to produce efficient and effective recommendation based on the buyer's interest.

On top of the commonly used recommendation techniques, there have been numerous research studies and a number of recommender systems developed to recommend books using different classification techniques [21-24]. Semantic Analysis, Nippon Decimal Classification (NDC), Opinion mining techniques, Naive Bayesian classifier, and other recommendation approaches like demographically based systems and community-based systems are selected ones.

After analyzing previous book recommenders, we found that most authors focused on either curriculum as the baseline for course content [7-10, 15], or on personal learning recommendations [11-12, 14]. In addition, various studies were carried out to find the gap between academia and industries [1-2]. Muthyala et al. [34] proposed an algorithm for job searching based on user profiles or resume, to enhance searching results by using skills or company history. However, the authors have not focused on learning contents. Instead, our proposed system recommends learning contents based on job-skills that have significant relationship with the academic curriculum.

2.3 The Computer Science Curriculum (CS2013)

Since 1978, Computer Science educators have looked to the ACM and the IEEE for guidelines for building undergraduate Computer Science courses and degrees [25-26]. CS2013 identified the essential skills and knowledge that should be required of all graduates of Computer Science programs [27]. Scholars surveyed approximately 1,500 Computer Science department chairs and undergraduate studies directors in the US.

CS2013 is categorized into a list of 18 Knowledge Areas (KAs), corresponding to the essential areas of computing study. All KAs consist of the body of knowledge called Knowledge Units (KUs):163 knowledge units in total. Topics in KUs are identified as "Core" or "Elective." The Core topics are further subdivided into "Tier-1" and "Tier-2". The Core Tier-1 topics are compulsory courses available in any Computer Science curriculum where all students should complete during their introductory courses. The Core Tier-2 topics are generally essential in undergraduate Computer Science degree programs. Computer Science programs can allow students to focus on certain areas in which some Core Tier-2 topics are not required. Figure 1 shows a hierarchical representation of the "Algorithms and Complexity" from the list of Knowledge Areas, which includes seven KUs, Core-Tier1 topics, Core-Tier2 topics, and learning outcomes.

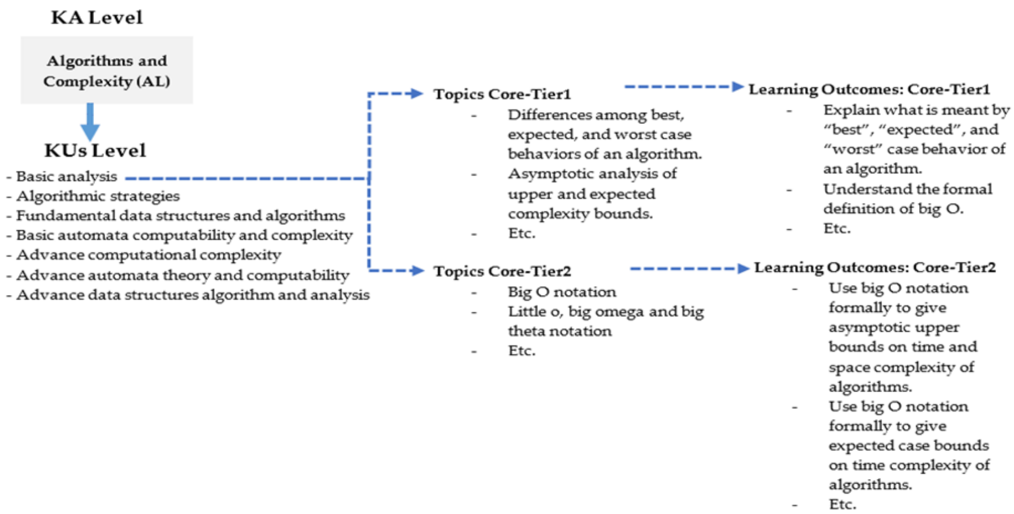


Figure 1. An example illustration of Computer Science Curriculum 2013 structure

There is a lot of research [3, 6-7] proposed the advised system for course selection. They utilize various technologies for construction of the application for university course advising. However, their research did not consider using job-skills information. In the initial stage of our study, we explored KAs and KUs from CS2013 before considering job-skill data from the industry.

2.4 The Gab between Academia and Information Communications Technology (ICT) Industry

Technologies are rapidly growing and emerging innovation. It is impossible for universities to educate students with the necessary skills that they will need to know for every potential job. According to the recent research studies, it has been recognized that most graduates are not fulfilling the industry requirements of required skills, abilities and knowledge [2, 30]. This indicates a gap between the abilities of graduates and industry expectations and it can prevent them from succeeding in their careers.

Radermacher and Walia [31] discussed why students are graduated with lack of required skills and abilities. They found out that there are still missing areas where students do not possess with necessary skills or knowledge as expected by the industry. Fuqha et al. [1] revealed that there is a huge gap between what is being taught in the university and what industry actually needs in terms of professional skills and talents. According to the aforementioned studies, it reveals that the curriculum should be up-to-date and cover the industrial requirements including graduates' working abilities, skills, and knowledge. However, it is not an easy task to improve the curricular regularly.

3 The Proposed System

In this section, we introduced our Learning Contents Recommender (LCRec), which was developed for producing learning contents based on required job-skills (JSs) and knowledge units (KUs) from CS2013. As of now, CS2013 is the latest curricular guidelines for undergraduate programs in Computer Science. KUs were used as the first dataset, which is the base of knowledge for computer science students. In order to identify the industry requirements for IT, we extracted the job-skills from the real IT job websites and used them in our system, as the second dataset.

Figure 2 illustrates the system architecture in two separate sections: the Interface Block (top) and the Functional Block (bottom). LCRec users directly interact with the functions presented in the Interface Block. First, the user provides keywords of the job-position, the job-skill, or both in the search boxes as the input data. Then, the system will show a list of relevant job-positions, job-skills, and a short description of each job-positions. The user can select the interesting job-position or job-skill and get the learning contents based on their interested skill. Learning contents are generated by the system for users in PDF file based on job-skills and the relevant knowledge. Moreover, we evaluate user satisfaction by constructing three different questionnaires for three groups' user including professional, academic, and student. Interested readers should visit our website for a demo at http://brs.minelab.tw/about/demo.html.

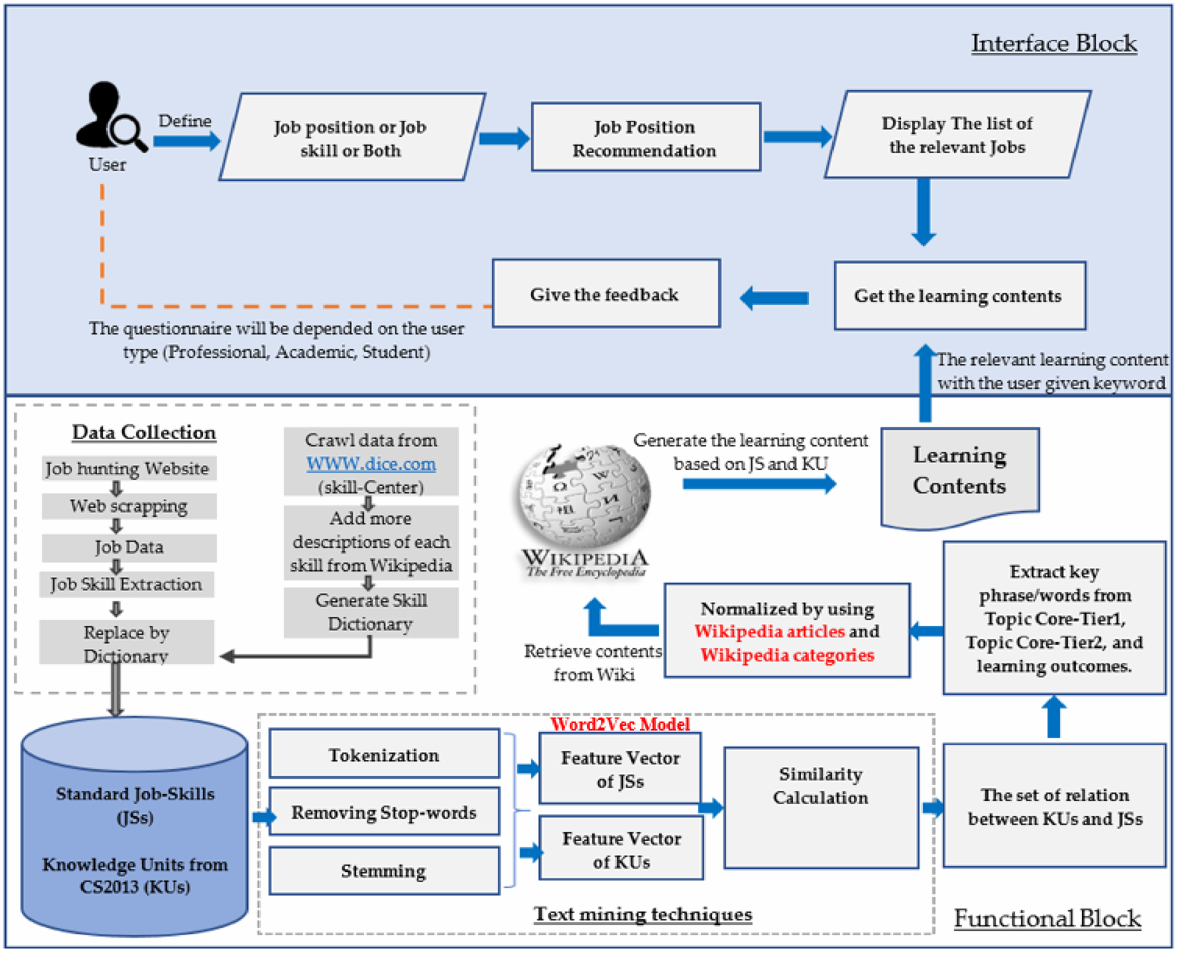


Figure 2. The system architecture of Learning Contents Recommendation System (LCRec)

In the Functional Block diagram, first, we crawled job information from 48 websites. Then, we extracted the required skills from the job description. All job-skills were standardized by the skill-dictionary, which was constructed by referring skill-center from www.dice.com and Wikipedia. How to standardize Job-skills was explained in section 3.3.1 and Algorithm 1; the extracted Job-skills are in various style, such as "C", "C language", "C programming", etc. To avoid ambiguity, we created the Skill-Dictionary to standardize all job-skills. Hence, above skills will be transformed to "C (programming language)". Next, text pre-processing techniques were applied: word-tokenization, removing stop-words, and word stemming, to clean our data sources. KUs and JSs were represented as a vector of a weighted point cloud of embedded words. We leverage word2vec model which celebrated by Mikolov [13]. This model learns a vector representation for each word using a neural network language model. We then apply cosine similarity method to define the relation between KUs and JSs.

The processed result is a set of correspondence between the base of knowledge and job-skills. We extracted keywords from KUs using Rapid Automatic Keyword Extraction (RAKE) algorithm [29] then used Wikipedia articles and Wikipedia categories to standardize these keywords. Finally, LCRec generated a PDF file, which includes a collection of learning contents extracted from Wikipedia.

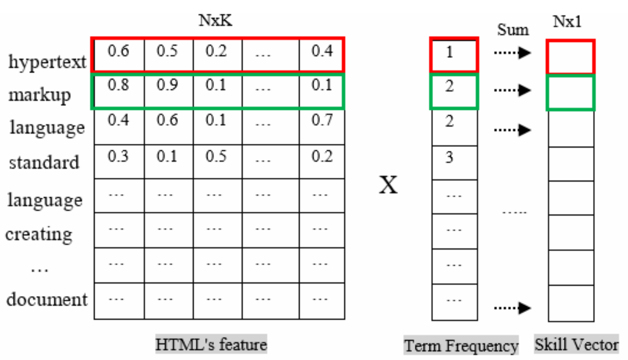
3.1 Data Collection

3.1.1 Job Data

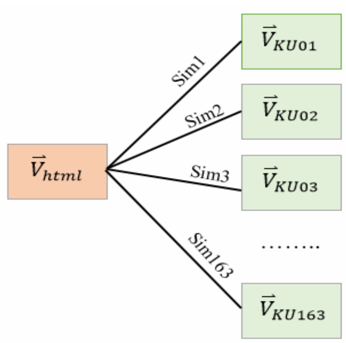
The main job data set involves 32,198 entries for job representation. Job attributes include the following fields: a job id, a job title, job description, the location of a company, the name of an employer, job category such as IT, Telecommunication, Monitoring, annual salary, and the reference to the job website.

We used real and live data available on different job-hunting sites. There are in total 48 job websites (http://brs.minelab.tw/about/paper_ref.html?#jobwebsite.), 951 locations, 3,334 companies, and 21,531 titles in our experiments. After examining the contents, it shows that 29,961 jobs have related skills on Computer Science while 2,237 jobs do not related. Figure 3 shows an example of job-description that we used “web developer” as a keyword for retrieving job position from the cv-library.co.uk website. We can observe the required skills embedded in the description. A web crawler can be used to retrieve the job description and to extract the required skills.

Figure 3. Job description example from cv-library.co.uk used “web developer” as the keyword



(a) The vector representation of HTML skill



(b) The matching between HTML skill and all Knowledge Units

Figure 4.

3.1.2 Skill Data

Our skill dataset consists of 2,799 entries, which are related to Computer Sciences. Those skills were normalized by the skill-dictionary. To creating the skill-dictionary are explained in Section 3.3.1. Additional descriptions of each skill were extracted from Wikipedia by using MediaWiki API, which is a web service that provides convenient access to wiki features, data, and metadata over HTTP. We employed query and parse “actions” as parameters to obtain information.

3.2 System Functionalities

3.2.1 Job Position Recommendation

This method uses a content-based similarity technique to find relevant job positions with given keywords from the user. In this process, standard procedures for text preprocessing were applied, such as Tokenization, Stop-words removal, and Word stemming. The given keyword is searched in the job collection to find the relevant job position. The job collection is represented in vector space model, which is an algebraic model for representing text document as vectors of identifiers. Cosine-Similarity and TF-IDF weight were used to find the relevant job positions with user query terms. The results are ranked based on the similarity score then display *n* job position to the user. Figure 5 shown the example of required skills based on job position and the list of job positions based on job skill. Users could select an interest job or skill.

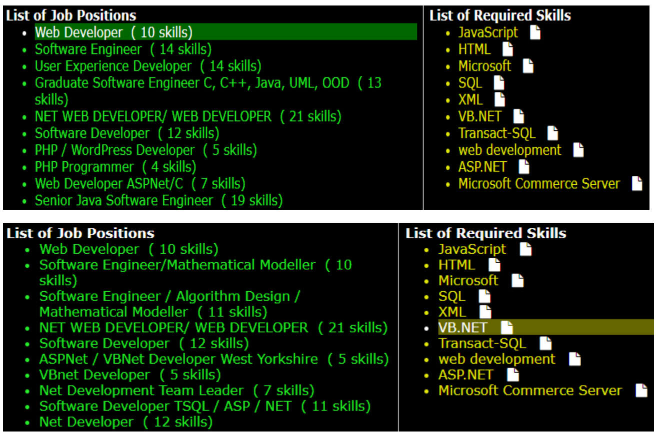


Figure 5. The list of required skills based on job position

3.2.2 Job-Skill and Knowledge Units Matching

One of the main purposes of LCRec is to bridge the gap between academic and industries. To overcome this issue, we created the matching function to identify the relation between KUs and JSs. Each job-skill and knowledge unit was represented as a vector of word embedded weight from the trained model. The ability to train on very large datasets allows the model to learn complex word relationships. We trained the word2vec on Wikipedia corpus with 100 embedded dimensions based on the skip-gram model which are shown Figure 6. Each word of skill description are represented as a weight vector. The Term Frequency (TF) of each word in our dataset (KUs and JSs) are found and are multiplied with its corresponding word vector. We sum the word vectors together element wise to get a single vector. Finally, each our data is represented by *Nx1* dimensional vector, where *N* is the number of skill's feature. Figure 4(a). demonstrates the vector representation of HTML skill with *NxK* dimensions, which *N* is the number of HTML's feature and *K* is the embedded dimension. After, each word vector is multiplied with its TF, we then sum them together. Finally, the vector of HTML will be normalized to *Nx1* dimension. We repeated this process for all job-skills and knowledge units, therefore our data set is represented in the vector space model.

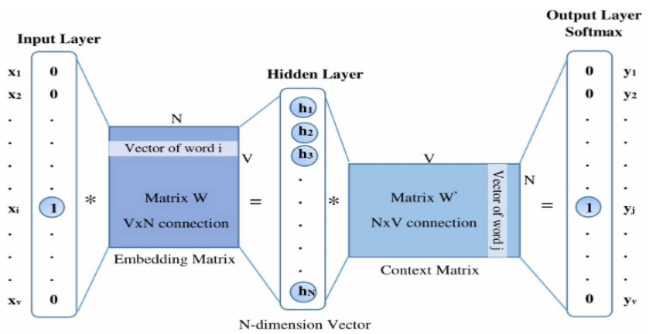


Figure 6. Word2Vec training process

We can obtain how each job-skill match into each knowledge unit by applying Cosine Similarity to measure similarity score as shown in Figure 4(b). We finally choose the top three highest score of KU for each job-skill. Given two vectors of attributes, Job-skill vector(JS) and Knowledge Unit (KU), the cosine similarity, cos(θ), is represented using a dot product and magnitude as:

$$sim(JS, KU) = \frac{\sum_{i=1}^{n} JS_i KU_i}{\sqrt{\sum_{i=1}^{n} JS_i^2}\sqrt{\sum_{i=1}^{n} KU_i^2}} \quad \textbf{(1)}$$

3.2.3 Creating a List of Content Topics and Skill Book Generation

We combined Rapid Automatic Keyword Extraction (RAKE) [33] and MediaWiki APIs to construct the learning contents, based on job-skills and CS2013. RAKE method begins at keyword extraction on an individual document by parsing its text into a set of candidate keywords. In our work, we used Topics-Core-Tier1 and Topics Core-Tier2 from CS2013 as the Knowledge Unit description (KU-desc) to extract the set of keywords. Figure 7 shows an example of KU-desc of "Algorithm Strategies."

Algorithmic Strategies
Brute-force search algorithms. Greedy algorithms. Divide-and-conquer (cross-reference SDF Algorithms and Design Problem-solving strategies). Recursive backtracking. Dynamic Programming, Branch-and-bound. Heuristics. Reduction. transform-and-conquer

Figure 7. The combination between Topics-Core-Tier1 and Topics Core-Tier2 of KU02 Algorithmic Strategies

To generate a set of KU-desc keywords, first, KU-desc was split into an array of words by specified word delimiters. This array was then split into sequences of contiguous words at phrase delimiters and stop-word positions. Input parameters for keyword extraction comprise a list of stop-words, a set of phrase delimiters, and a set of word delimiters. Second, words within a sequence were assigned the same position and were considered as a candidate keyword, which is denoted by *P* in Algorithm 1. After every candidate keywords were identified, a keyword score is calculated for each candidate keyword(*score_x_l*). In order to calculate the score, we used a matrix to represent the word co-occurrence ($M[i][j]$), where words were arranged in rows and columns. The frequency of word co-occurrences was denoted in the element of M . We use the word frequency (*freq(w_i)*), word degree(*deg(w_i)*), and ratios of degree to frequency (*freq(w_i)/deg (w_i)*) to calculate word scores. The keyword score for each candidate keyword was computed as the sum of its member word scores. As an example, the list of candidate keywords of "*Algorithmic Strategies*" knowledge unit and their scores are shown in Figure 6. In order to enhance the quality of the keyword extraction, Wikipedia-Categories and Wikipedia-Article play an important role in normalizing. Since Knowledge Area was defined with respect to Wikipedia categories based on Areas of Computer Scienc (http://brs.minelab.tw/about/paper_ref.html?#KAandWikiCategories), we can differentiate KU categories through their KA. Then, we can it to filter out the useless candidate keywords that not related with Computer Science. The implementation detail of the KUs keyword generation is summarized in Algorithm 1.

Let us formally define and give some examples of variable in Algorithm 1 as follows:

Algorithm 1. Knowledge Unit (KU) Keyword Extraction

Input: $\Omega = \{KU\_desc\}$, $C = \{Wikipedia\_catogory\}$

Output: $\beta = \{KU\_keyword\}$

$K = \{KU\_candiKey \mid K_i = [candiKeyOfKU_i]\} \quad \forall \omega(\omega \in \Omega)$

$K_i \leftarrow RAKE(\omega_i)$

$\forall k \in K_i$

$c \leftarrow CategoryOfk$

IF $c \subseteq C$

$\beta_{ij} \leftarrow a \mid a$ *is a wiki-article of k*

Return β_i

Procedure of $RAKE(\omega_i)$

$\omega_i = [w_1, w_2, \ldots.., w_d] \mid d = \mid \omega_i \mid$

$P = [x_1, x_2, \ldots.., x_q] \mid x$ *is co-occurrence word in* ω_i

$M[i][j] \leftarrow 0$

For *(i=0 to d)*

For *(j=0 to d)*

If $i == j$

$M[i][j] = freq(w_i)$

Otherwise

$M[i][j] = \deg(w_i)$

$RatioW_i \leftarrow freq(w_i) / \deg(w_i)$

For $l = 0$ **to** $q \mid q = \mid P \mid$

$score\_x_l \leftarrow \sum_{i=0}^{m} RatioW_i \mid w_i \in x_l$

Order P by score

Return P

- Let Ω be a set of KUs, where ω is a vector of KU description. Thus, $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ and n is the number of KU that equals 163, while w_i represents a word in ω_i. The example of KU description is shown in Figure 5.
- Let K be a set of all KUs candidate keywords that are given from the RAKE procedure, where $K = [K_1, K_2, \ldots, K_n]$ and each $K_1 = [k_1, k_2, \ldots, k_m]$, where m is the number of candidate key of K_i. Figure 8 shows the set of candidate keys of KU02. We can refer its element by k_{ij}, here i = 2. So, $k_{2,0}$ is "design problem solving strategies," $k_{2,1}$ is "brute-force search algorithms," ect. Each k_{ij}, is defined as a category (c) by using Wikipedia APIs. Table1 shows the examples of $k_{2,1}$ and its categories.

[design problem-solving strategies] => 9
[brute-force search algorithms] => 8.66
[cross-reference sdf algorithms] => 8.66
[greedy algorithms] => 4.66
[transform_conquer] => 4
[branch_bound] => 4
[recursive backtracking] => 4
[dynamic programming] => 4
[divide_conquer] => 4
[heuristics] => 1
[reduction] => 1

Figure 8. A list of candidate keywords of KU2 and their keywords scores

- Let C be a set of Wikipedia Categories, which relates to each KU through KAs. Hence, categories of KU02 (Algorithmic Strategies) are "Algorithms and data structures" and "Theory of computation." All KAs categories are shown in Table 1.

Table 1. Algorithmic strategies keywords and their categories

Original Word from RAKE	Wikipedia-Category	Wikipedia-Article
Brute-force search algorithms	Search algorithms	*Brute-force_search*
Greedy algorithms	Optimization algorithms and methods	*Greedy_algorithm*
Recursive backtracking	Search algorithms	*Backtracking*
Dynamic Programming	Dynamic programming, Optimization algorithms and methods	*Dynamic_programming*
Heuristics	Problem solving methods	*Heuristic*

- Let v represents a Wikipedia-Article of $k_{i,j}$, if $k_{i,j}$, is "brute-force search algorithms," v is "Brute-force_search." Then we store v in β_i, where β_i is a set of keywords of KU_i, for an instance, β_2 ={"Brute-force_search," "Greedy_algorithm," "Backtracking," "Dynamic_programming," "Heuristic"}
- P is a set of co-occurrence words of KU description.
- $freq(w_i)$ is the occurrence frequency of w_i in the document.
- $\deg(w_i)$ is the number of words, which occurs with w_i together in the document.

After completing KU keyword extraction (Algorithm 1) for all KUs, the set of book chapter is generated. Table 1 shown the example of keywords of knowledge unit named "Algorithmic Strategies". First column shows the co-occurrence words that are extracted from the RAKE, while the second column shows word categories. The last column shows the Wikipedia article that we used as a set of keywords to retrieve data from Wikipedia and they are the chapter of the generated learning content as well. Wikipedia provides APIs to extract data, which we can turn data into a rich knowledge base. Finally, learning contents were constructed based on skill names and KU keywords. We extract data from Wikipedia and represent all content in the PDF format that users can use it as a material to improve their skills both online or offline.

3.3 Wikipedia

Wikipedia is a global, free, and multilingual internet based encyclopedia. Wikipedia allows users to add, edit, or update the contents of their own or others' after registered. To access Wikipedia advanced functions, users can use MediaWiki, which is a unique Wiki engine developed by Wikipedia. In our proposed system, Wikipedia play the important role, which had been used for three main purposes as follows:

3.3.1 Creating the Skill Dictionary

Le et al. [28] proposed Skill2Vec that inspire from Word2Vec, which is a methodology to convert context of the document to vector by using the word-embedding technique. Their work also extracted job-skills from job description but without skill descriptions. This is one of the reasons to create the skill-dictionary to fulfill the skill description for our research. Wikipedia is helpful for skill-dictionary creation. The extracted Job-skills are in various style, such as "C", "C language", "C programming", etc. To avoid ambiguity, the skill-dictionary help to standardize all job-skills. Job-Skills are transformed to Wikipedia-article that we can use wiki content as skill description. For the example "C" or "C language" will be converted to "C (programming language)", "AJAX" will be converted to "Ajax (programming)", etc.

3.3.2 Content Table Creation

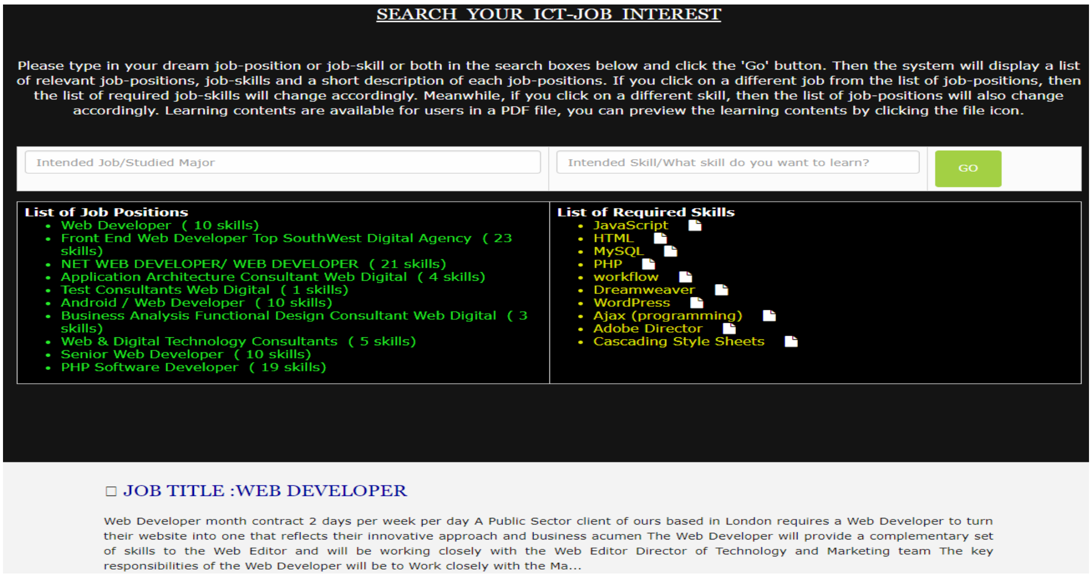
After the candidate keywords were extracted by RAKE algorithm, they were converted to Wikipedia article to create the content table of LOs. However, some Wikipedia articles are not relevant to ICT domain. Wikipedia categories were utilized as a filter to filter-out irrelevant words. The generated filter consists of 17 main categories in Computer Science area and its 5,000 subcategories. KUs were defined their categories based on relative values (http://brs.minelab.tw/about/paper_ref.html?#KAandWiki). During the converting process, we retrieved the categories of keyword (Wikipedia article), and then checked with the sub-categories based on the relevant KUs. If the keyword's categories are in the list, we kept it as one chapter in the content table. Otherwise, if it is not in the list, we removed it from the content table. For example, if we were considering a keyword from KU under "Algorithms and Complexity" area, its sub-categories should be relate to "Algorithms and data structures or Theory of computation". For instance, "Brute-force_search" had been extracted, which is an actual category in "Search algorithms." In this case, "Search algorithms" is a subcategory of "Algorithms and data structures." Therefore, "Brute-force_search" will become one chapter of the learning content. On the other hand, if "Mobile_application_development" had been extracted from this KU, we removed this item from the content table because its category is not related to "Algorithms and data structure" nor "Theory of computation," even though it is in the ICT domain. This is further explained in section 3.2.

3.3.3 Learning Contents Generation

After the content table was constructed, MediaWiki APIs was applied to retrieve the contents from Wikipedia. These contents are composed of Info-box, Text, Hyperlink, Image, and Table. Wiki parser and the particular function that we created parsed each Wikipedia page to our learning content format. Finally, Wiki-based Skill Book was produced in PDF for the users to print or download. Each learning topic of LCRec Skill Book has been listed as the book chapter that can directly link to the main content for that topic. Students easily to connect each knowledge unit that can improve visual fatigue that is one factor in e-book learning activity [9].

4 Experiment and Evaluation

LCRec (http://brs.minelab.tw/) was designed to help graduates and professionals to find appropriate learning contents. First, the system checks user's status and directs the user to a job search page. When the user searches for an intended job or skill (or both) by using a keyword (e.q., "web developer" or "PHP"), the system produces a list of relevant job-positions in the left block and the list of required skills in the right block, as shown in Figure.7a. If the user clicks on a different job from the list of job-positions, the list of required job-skills will be changed accordingly. Similarly, if the user clicks on a different skill, the list of required job-skills will be changed accordingly. The short descriptions of each job-position are shown below. In the event that the user clicks on the job title, the user will be directed to the full job-description page. In this page, the required skills are displayed on the top, which link the user to the Wiki skill-book (Figure 9b). Initially, the required skills are displayed according to the first job-position. Figure 9 demonstrates the essential skills of "Web developer" which comprise JavaScript, HTML, MySQL, PHP, etc. For this instance, "Web developer" was used as a searching keyword. A file icon can link users to the Wiki skill book.



(a) The searching page of Learning Contents Recommendation System (LCRec)

(b) The required job-skills and full job-description by clicking on "Web Developer" title. Link: (http://brs.minelab.tw/index.php)

Figure 9.

The Wiki skill-books are generated based on the job-skill and KUs. For the example, the content of "PHP" skill-book will involve "Web Platforms (ku-100)," "Software Construction (ku-139)," and "Web Security (ku-50)", while "CSS" skill-book will involve "Software Design (ku-138)," "Data Information and Knowledge (ku-20)," and "Designing Interaction (ku-35)," etc. The relation of JSs and KUs were described in Section 3 (3.2.2). This relation benefit for users in referring to the academic course via the body of knowledge and also provides a bridge between the industry and the academia.

Table 2 shows the comparison of 10 chapters of the learning contents from LCRec and Wikibook from https://en.wikibooks.org; Wikibooks is a collection of open-content textbooks, based on "PHP" skill. Since the LCRec learning contents are generated from job-skill and the relevant KUs, its contents include various topics. If students would like to be a PHP web developer, they should know about web standards, secure coding, software quality, etc., while Wikibook provides the only PHP content (see Section 5 for diversity discussion).

Table 2. A comparison of 10 chapters of LCRec learning content and Wikibook based on "PHP" Skill

	LCRec Learning Content	Wikibook
Chapter 1	PHP	Introduction
Chapter 2	List Of Programming Languages	Setup And Installation
Chapter 3	Computing Platform	How Do I Know My Setup Is Working?
Chapter 4	Web Standards	Uses Of PHP
Chapter 5	Defensive Programming	Start A PHP Beginner Tutorial
Chapter 6	Secure Coding	Beginning With "Hello World!"
Chapter 7	Software Quality	Commenting And Style
Chapter 8	Web Application Security	Comparison
Chapter 9	Application Firewall	Arrays
Chapter 10	Browser Security	The If Structure

LCRec was evaluated by IT professionals, academics and students. The experiments were carried out in Mongolia (MN), Sri Lanka (LK), Taiwan (TW), and Thailand (TH). In order to evaluate the psychometric characteristics of users, we created a questionnaire based on the IBM Computer Usability Satisfaction Questionnaires [35]. We evaluated and gathered both subjective and objective facts of the LCRec system for realistic scenarios-of-use. The questionnaire comprises 12 items for each user category (http://brs.minelab.tw/about/paper_ref.html?#questionnaires), which were designed for inquiring the basic personal information and measuring the users' satisfaction on usage, the content suitability, and contents consistency of the system. The questionnaire provides function to users and let them to give the satisfied score ranged from 1 to 5: 1 = very satisfied, 2 = satisfied, 3 = unsure, 4 = dissatisfied, and 5 = very dissatisfied. Based on our experiment, there are 461 users (248 students, 111 professionals, and 102 academics) have given feedback by completing the questionnaire from 477 users in total.

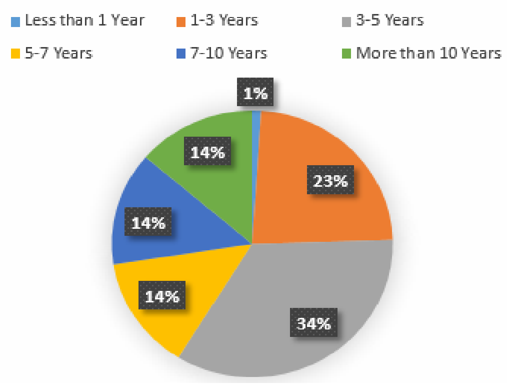
The feedback scores were analyzed by using the IBM SPSS statistic software package. We calculated the mean, standard deviation, standard error of mean and median for all user categories as shown in Table 3. Questions 1~3 are focused on personal information to identify the users' background of all three categories. The ratios of the academic and professional users based experiences are illustrated in Figure 10(a) and Figure 10(b), respectively. After analyzing the user experience for professionals and academics, the higher portion of the ratio is identified as 3-5 years. Questions 4~6 are designed to measure the content consistency of the system. Questions 7~9 measure the content suitability of the system and questions 10 and 11 are designed to check the user's satisfaction with the system. Finally, the user can provide additional comments or suggestions in Question 12. As illustrated in Table 3, the system was evaluated with a high mean value and low standard deviation value. A low standard deviation value indicates the average score for each question of user tends to be close to the mean. Based on these evaluations, we can claim that a large number of users are satisfied with our proposed system.

5 LCRec Content Diversity

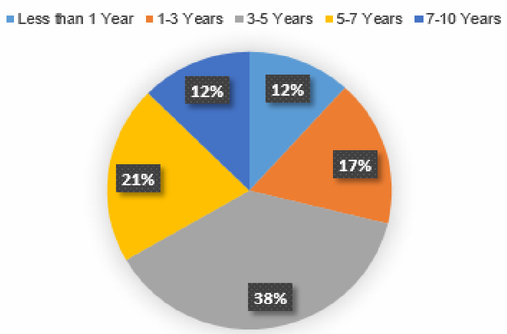
In a practical working environment, each IT job position requires the essential skills and each skill should be comprised of many knowledge areas. Therefore, the learning contents should provide a diverse knowledge to a student. Even though the main objective of our proposed system (LCRec) is to produce learning contents based on job-skills and knowledge units to cater a diverse knowledge for students, we decided to carry out a learning content diversity comparison between LCRec and completed Wikibooks from https://en.wikibooks.org. Wikibook is a collection of open-content textbooks that have no size limit.

Table 3. Feedback analysis

	Parameter	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Students	Mean	4.04	3.81	4.00	4.03	3.81	4.00	3.87	4.01	4.12
(N = 248)	Std. Deviation	.864	.903	.875	.837	.951	.823	.853	1.008	.907
Country: (1 CN, 41 MN, 3	Std. Error of Mean	.056	.059	.057	.055	.062	.054	.056	.066	.059
LK, 80 TW, 122 TH, 1 VN)	Median	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
Professional	Mean	-	4.49	4.06	3.67	3.29	3.87	4.29	4.22	4.40
(N = 111)	Std. Deviation	-	.939	.926	1.010	1.029	.883	.785	.916	.759
Country: (5 MN, 1 SG, 58	Std. Error of Mean	-	.090	.089	.097	.099	.085	.075	.088	.073
LK, 47 TH)	Median	-	5.00	4.00	4.00	4.00	4.00	4.00	4.00	5.00
Academic	Mean	-	3.95	4.17	4.13	3.84	4.14	4.27	4.22	4.20
(N=102)	Std. Deviation	-	.887	.767	.868	1.214	.939	.859	1.047	.949
Country: (31 MN,26 LK, 44	Std. Error of Mean	-	.088	.076	.086	.121	.093	.085	.105	.094
TH, 1 UK)	Median	-	4.00	4.00	4.00	4.00	4.00	5.00	5.00	4.00



(a) The professional user



(b) Based on the experience

Figure 10. The ratio of the academic user

We measure the diversity of document based on the content, and it can be interpreted as the expected value of the categorical distance, where the expectation is with respect to the distribution pairs of elements. As a starting point, we examine the diversity in the learning context by using Rao's theory measure [32]. The formulation was originally proposed as shown in (2),

$$div = \sum_{i=1}^{T}\sum_{j=1}^{T} p_i p_j \delta(i,j) = \pi' \Delta \pi \qquad \textbf{(2)}$$

where, p_i is the probability of the population t belongs to category i. p_j is the probability of the population t belongs to category j. δ(i, j) is the distance between categories i and j, Δ is a TxT matrix where elements of matrix are distance between categories i and j. π' is the transpose of the Tx1 vector of proportions π, and T is the total number of categories.

Bache et al. [33] also applied Rao's theory to construct a text based framework for quantifying how to diverse a document in terms of its content. They have used words as elements, topics as word categories, and documents as collections of words. They have created a *DxT* document topic count matrix with entries n_{dj} corresponding to the number of word tokens in document *d* that are assigned to topic *j*. Based on this; they can define Rao's diversity measure for each document *d* as in (3).

$$div^{(d)} = \sum_{i=1}^{T}\sum_{j=1}^{T} P(i\,|\,d)P(j\,|\,d)\delta(i,j) \qquad \textbf{(3)}$$

where *P(i|d)* is the proportion of word tokens in the document *d* that are assigned to topic *i*, estimated as $n_{di}/\ n_d$, n_d is the number of word tokens in *d*, and $\delta(i,j)$ is a measure of the distance between topic *i* and topic *j*.

According to the experimental design, first we created a count-matrix (http://brs.minelab.tw/about/paper_ref.html?#countmatrix) to represent the category distribution of learning content elements, where each row represents a document and column represents a category. Ten learning objects of LCRec and Original Wikibook were compared in diversity term based on seventeen categories. After constructing the count-matrix, we applied (3) to calculate the diversity value of each document based on job skill. In order to measure topic distance (i.e., *s(i,j)*), the cosine similarity was applied to calculate the topic similarity on co-occurrence within documents, which is defined as:

$$s(i,j) = \frac{\sum_d n_{di} n_{dj}}{\sqrt{\sum_d n_{di}^2}\sqrt{\sum_d n_{dj}^2}} \qquad \textbf{(4)}$$

where *i* and *j* represent two column indices (two categories) and $\sum_d$ represents a sum over all documents was indexed by *d*.

In order to convert each similarity measure into a distance measure, we used two functions: $\delta(i,j) = 1/s(i,j)$ and$\delta(i,j) = 1-s(i,j)$ for (2), *s(i,j)* is computed by (4) and the count-matrix. The diversity score of ten learning objects are compared and shown in Table 4, while the deiversity scores based on the converting function $\delta(i,j) = 1-s(i,j)$ are plotted in Figure 11 to easily compare. From the experiment results demonstrated that the diversity score of LCRec's learning content are higher than Wikibooks. The performance gain difference between the learning contents of LCRec and Wikibooks is 40%, which confirms that our learning contents are diverse and covered with many knowledge areas under the Area of Computer Science.

Table 4. The Diversity Measure Scores based on Two Transformation Method for Ten Job-Skills

	δ(i,j) = 1/s(i,j)		δ(i,j) = 1-s(i,j)	
Skill	LCRec	Wikibooks	LCRec	Wikibooks
Agile	0.507	0.413	0.046	0.016
CSS	0.525	0.369	0.094	0.038
HTML	0.473	0.450	0.069	0.034
Java	0.701	0.466	0.067	0.015
Python	0.501	0.444	0.049	0.051
C Programming	0.577	0.266	0.035	0.029
Computer Network	0.519	0.385	0.106	0.02
Data Compression	0.509	0.415	0.124	0.029
.Net Framework	0.713	0.443	0.048	0.031
Software Testing	0.514	0.287	0.085	0.016

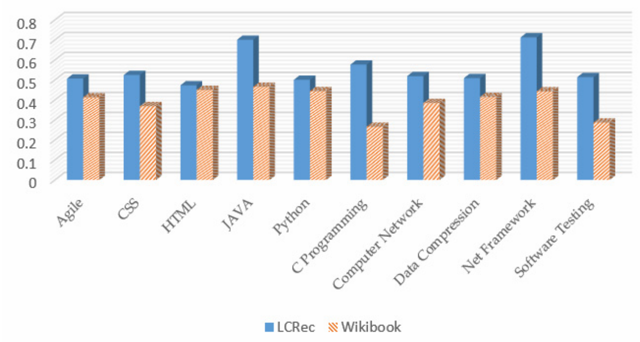


Figure 11. The diversity measure between two type of learning content of job-skills by using $(i,) = 1\text{-s}(i, j)$

6 Discussion

According to CS2013, 17 curriculum exemplars are provided from a variety of educational institutions. Each example shows how an institution's existing curriculum covers KU Core-Tier1 and KU Core-Tier2 topics. We summarized knowledge skills are taught in four institutes based on the curriculum overview and course descriptions from CS2013 as shown in Table 5. The first column shows the name of educational institute and its department, while the second column shows a list of skills that were extracted from their course descriptions. After comparing the skills in Table 5 and the top required skills from the selected job positions in Figure 12, we found that few skills are omitted in the University curricula, for instance "CSS," "Agile," "Oracle," and "Jquery", etc. These skills are practical skills, which could be taught in technical universities. However, similar skill concepts could be included in their related courses. Universities could consider including additional laboratory sessions or workshops based on the essential skills and could encourage students to practice themselves.

Table 5. List of skills found from four institutes

Educational Institutions	List of Skills
Stanford University (Department of Computer Science)	C++,C, SQL, XML, DTD, XPath, XQuery, XSLT, UML, OLAP, JSON, NoSQL, Java
Bluegrass Community and Technical College (Department of Computer and Information Technologies)	C++, Unix, Linux, Shell Script, GNU, HTML, OOP, Java Script, PHP, Ruby, Rail, XML, AJAX, ASP.NET, Graphic skill
Grinnell College (Department of Computer Science)	GNU, Linux, C, C++, Java, HTML, Python
Williams College (Department of Computer Science)	Java, TCP/IP, ALU Design, CMOS circuits, C++, OpenGL, HTTP/UDP/TCP, Lisp, Scala

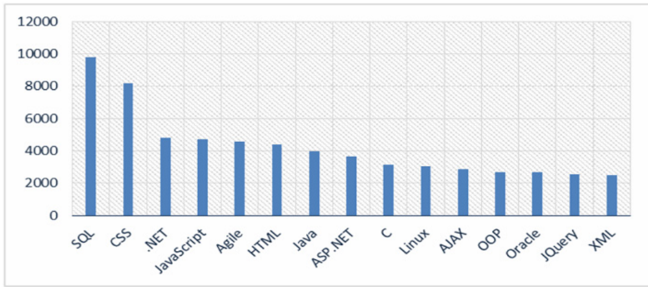


Figure 12. Top 15 skills in jobs

We noticed "Software Engineering" and "Information Assurance and Security" as the most relevant knowledge and SQL and CSS as the essential skills required in the IT industry. On the other hand, we found that 33 KUs are not related to job-skills, such as "Algorithmic Strategies," "Machine Level Representation of Data," "Multiprocessing and Alternative Architectures," "Performance Enhancement," and "Parallel Decomposition." We believe these KUs are theories, which are suitable for research studies but not for the practical knowledge required by the industry.

7 Conclusion

In this research, one of the key issues was to identify why universities may not educate students with necessary skills in order to meet industry expectations. Based on recent research studies, it was found that there is a gap between what is taught in the university and what industry actually needs in terms of professional skills and talents. In order to bridge the gap between graduate competency and industry expectations, universities should work more closely with the industry and line up the essential skills in the curriculum. LCRec can effectively support academics in updating curriculum that better matches required job-skills in the industry. Moreover, since it is not an easy to make regular changes in a university curricular, LCRec also supports students to find appropriate learning contents based on the required job-skills in the industry for their intended job criteria.

LCRec has been evaluated with 111 IT professionals, 102 academics, and 248 students in the field of Computer Science, Information Technology, and Software Engineering. The statistical result demonstrates that users are satisfied with the proposed system and our proposed system can help students to improve their essential skills.

In addition, we received some valuable suggestions from LCRec users for further improvements (http://brs.minelab.tw/about/paper_ref.html?#usercomment). Some suggestions are: "The subject of the online material should link to the university course", "The system should provide the online course", "The system should be developed into more specific areas", and "It is better to improve the learning contents related to Information Security and Information Technology jobs and its skills". These suggestions benefit for our future work.

Furthermore, given a list of skills taught in a university, we can identify the missing skills. Academics can use this information as a basic idea to enhance curricula. Many users test the prototype system and it is available for public usage at http://brs.minelab.tw. The feedback from users indicates that the system is ready for public to use. Technical transfer to a local company is under negotiation. In the future, we plan to work on semantic analyzation to produce the learning content sequencing and it would be more beneficial for LCRec users, if our research solution could be expanded to cater STEM disciplines.

Acknowledgments

The authors would like to thank Ministry of Science and Technology of Taiwan and Ministry of Education, Culture, Science and Sports of Mongolia for the support of research under contract No. MOST105-2923-S-008-001-MY3. This work has been done within the framework of the project (P2018-3587) supported by the National University of Mongolia.

References

[1] C. O. Kane, Bridging the Gap between Academics and Industry, *Proceedings of INTED2010 Conference*, Valencia, Spain, 2010, pp. 4263-4269.

[2] C. S. Nair, A. Patil, P. Mertova, Re-engineering Graduate Skills: A Case Study, *European Journal of Engineering Education*, Vol. 34, No. 2, pp. 131-139, April, 2009.

[3] P. K. Tulsi, M. P. Poonia, Expectations of Industry from Technical Graduates: Implications for Curriculum and Instructional Processes, *Journal of Engineering Education Transformations*, Vol. 28, No. 4, pp. 19-24, January, 2015.

[4] C. K. Chang, G. Engel, W. King, E. Roberts, R. Shackelford, R. H. Sloan, P. K. Srimani, Curricula 2001: Bringing the Future to the Classroom, *IEEE Computer*, Vol. 32, No. 9, pp. 85-88, September, 1999.

[5] O. C. Santos, J. G. Boticario, Modeling Recommendations for the Educational Domain, *Procedia Computer Science*, Vol. 1, No 2, pp. 2793-2800, August, 2010.

[6] Q. Shambour, S. Fraihat, M. Hourani, the Implementation of Mobile Technologies in Higher Education: A Mobile Application for University Course Advising, *Journal of Internet Technology*, Vol. 19, No. 5, pp. 1327-1337, September, 2018.

[7] T. Tang, G. McCalla, Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment, *International Journal on ELearning*, Vol. 4, No. 1, pp. 105-129, November, 2005.

[8] F. Mödritscher, Towards a Recommender Strategy for Personal Learning Environments, *Procedia Computer Science*, Vol. 1, No. 2, pp. 2775-2782, August, 2010.

[9] P. Y. Cheng, Y. N. Su, Y. C. Chien, T. T. Wu, Y. M. Huang, An Investigation of Visual Fatigue in Elementary School Students Resulting From Reading E-book, *Journal of Internet Technology*, Vol.19, No. 5, pp. 1285-1292, September, 2018.

[10] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification, *Computers & Education*, Vol. 56, No. 3, pp. 885-899, April, 2011.

[11] S. Kanetkar, A. Nayak, S. Swamy, G. Bhatia, Web-based Personalized Hybrid Book Recommendation System, *International Conference on Advances in Engineering and Technology Research* (ICAETR), Unnao, India, 2014, pp. 1-5.

[12] V. D. Castillo-Carrero, I. Hernán-Losada, E. Martín, Prototype of Content-based Recommender System in an Educational Social Network, *1st Workshop on Video based Learning*, Graz, Austria, 2011.

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality, *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol. 2, Lake Tahoe, Nevada, 2013, pp. 3111-3119.

[14] H. Imran, M. Belghis-Zadeh, T. W. Chang, S. Graf, PLORS: A Personalized Learning Object Recommender System, *Vietnam Journal of Computer Science*, Vol. 3, No. 1, pp. 3-13, February, 2015.

[15] L. P. Shen, R. M. Shen, Learning Content Recommendation Service Based on Simple Sequencing Specification, *Advances in Web-Based Learning – ICWL 2004*, LNCS3143, Beijing, China, 2004, pp. 363-370.

[16] K. H. Tsai, T. K. Chiu, M. C. Lee, T. I. Wang, A Learning Objects Recommendation Model Based on the Preference and Ontological Approaches, *IEEE 6th Int. Conf. Advanced Learning Technologies*, Kerkrade, Netherlands, 2016, pp. 36-40.

[17] O. De Clercq, M. Schuhmacher, S. P. Ponzetto, V. Hoste, Exploiting Frame Net for Content-based Book Recommendation, *CBRecSys at ACM RecSys 2014*, Silicon Valley, CA, USA, 2014, pp. 14-21.

[18] C. Chen, L. Zhang, H. Qiao, S. Wang, Y. Liu, X. Qiu, Book Recommendation Based on Book-Loan Logs, in: H.-H. Chen, G. Chowdhury (Eds.), *The Outreach of Digital Libraries: A Globalized Resource Network (ICADL 2012)*, LNCS7634, Springer, Heidelberg, pp. 269-278, 2012.

[19] A. S. Tewari, A. Kumar, A. G. Barman, Book Recommendation System Based on Combine Features of Content Based Filtering, Collaborative Filtering and Association Rule Mining, *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, India, 2014, pp. 500-503.

[20] P. Mathew, B. Kuriakose, V. Hegde, Book Recommendation System through Content Based and Collaborative Filtering Method, *IEEE Int. Conf. Data Mining and Advanced Computing (SAPIENCE)*, Ernakulam, India, 2016, pp. 47-52.

[21] P. Jomsri, Book Recommendation System for Digital Library

Based on User Profiles by Using Association Rule, *IEEE 4th Int. Conf. Innovative Computing Technology (INTECH)*, Luton, UK, 2014, pp. 130-134.

[22] A. S. Tewari, T. S. Ansari, A. G. Barman, Opinion Based Book Recommendation Using Naive Bayes Classifier, *IEEE 2014 Int. Conf. Contemporary Computing and Informatics (IC3I)*, Mysore, India, 2014, pp. 139-144.

[23] L. Xin, E. Haihong, J. Song, M. Song, J. Tong, Book Recommendation Based on Community Detection, in: Q. Zu, M. Vargas-Vera, B. Hu (Eds.), *Pervasive Computing and the Networked World (ICPCA/SWS2013)*, NLCS8351, Springer, Switzerland, 2013, pp. 364-373.

[24] K. Priyanka, A. S. Tewari, A. G. Barman, Personalised Book Recommendation System Based on Opinion Mining Technique, *IEEE 2015 Global Conf. Comm. Technol. (GCCT)*, Thuckalay, India, 2015, pp. 285-289.

[25] M. Sahami, S. Roach, E. Cuadros-Vargas, D. Reed, Computer Science Curriculum 2013: Reviewing the Strawman Report from the ACM/IEEE-CS Task Force, *43rd ACM technical Symp. on Comp. Sci. Edu*., North Carolina, USA, 2012, pp. 3-4.

[26] C. Armstrong, K. Asanovic, R. F. Babiceanu, *Computer Science Curricula 2013*, ACM&IEEE Computer Society, 2013.

[27] S. Roach, M. Sahami, R. LeBlanc, R. Seker, Special Session: The CS2013 Computer Science Curriculum Guidelines Project, *2013 IEEE Frontiers in Edu. Conf. (FIE)*, Oklahoma City, OK, USA, 2013, pp. 1311-1313.

[28] V. D. Le, V. M. Quan, D. Q. An, *Skill2vec: Machine Learning Approaches for Determining the Relevant Skill from Job Description*. <https://arxiv.org/abs/1707.09751>, 2017.

[29] S. J. Rose, W. E. Cowley, V. L. Crow, N. O. Cramer, *Rapid Automatic Keyword Extraction for Information Retrieval and Analysis*, US8131735B2, March, 2012.

[30] R. Martin, B. Maytham, J. Case, D. Fraser, Engineering Graduates' Perceptions of How Well They Were Prepared for Work in Industry, *European Journal of Engineering Education*, Vol. 30, No.2, pp. 167-180, January, 2005.

[31] A. Radermacher, G. Walia, Gaps between Industry Expectations and The Abilities of Graduates, *Proceeding of the 44th ACM technical Symposium on Computer Science Education*, Colorado, USA, pp. 525-530, March, 2013.

[32] C. Rao, Diversity and Dissimilarity Coefficients: A Unified Approach, *Theoretical Population Biology*, Vol. 21, No. 1, pp. 24-43, February, 1982.

[33] K. Bache, D. Newman, P. Smyth, Text-Based Measures of Document Diversity, *Proceedings of the 19th ACM SIGKDD Conference,* Chicago, IL, USA, August, 2013.

[34] R. Muthyala, S. Wood, Y. Jin, Y. Qin, H. Gao, A. Rai, Data-driven Job Search Engine Using Skills and Company Attribute Filters, *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, New Orleans USA, November, 2017, pp.199-206.

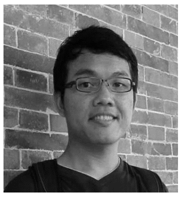
[35] J. R. Lewis, IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use, *International Journal of Human-Computer Interaction*, Vol. 7 No. 1, pp. 57-78, September, 2009.

Biographies

Chalothon Chootong is currently a Ph.D. candidate at the Department of Computer Science and Information Engineering, National Central University, Taiwan. Her research interests include social learning, social media mining, recommender system, and data mining.

Timothy K. Shih is a Distinguished Professor and the director of Innovative AI Research Center at the National Central University, Taiwan. He has received many research awards, including IIAS research award from Germany, HSSS award from Greece, Brandon Hall award from USA, the 2015 Google MOOC Focused Research Award.

Ankhtuya Ochirbat is an Associate Professor at the National University of Mongolia. She received her master in Computer Science from National University of Mongolia, and her Ph.D. in Computer Science from National Central University, Taiwan. Her research interests are social computing, machine learning, data mining, information retrieval, and recommender systems.



Worapot Sommool is currently working toward the Ph.D. degree in computer science and information engineering at National Central University, Taoyuan, Taiwan. His current research interests include big scholarly data, social network analysis, knowledge management, and recommender systems.

W. K. T. M. Gunarathne is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Central University, Tawan. His research interests include e-learning, information retrieval and extraction, and recommendation techniques.

Carl K. Chang is Professor of Computer Science, Professor of Human-Computer Interaction and Director of the Software Engineering Laboratory at Iowa State University. Chang is a Life Fellow of IEEE, a Fellow of AAAS, and a Life Member of the European Academy of Sciences.