Improvements of a Hybrid Syllabus Search Tool by Syllabus-related Heuristics

Takayuki Sekiya
Information Technology Center
The University of Tokyo
3-8-1 Komaba, Meguro,
Tokyo, Japan
sekiya@ecc.u-tokyo.ac.jp

Yoshitatsu Matsuda
Department of Computer and Information Science
Seikei University
3-3-1 Kichijojikitamachi, Musashino-shi,
Tokyo, Japan
matsuda@st.seikei.ac.jp

Kazunori Yamaguchi
Graduate School of Arts and Sciences
The University of Tokyo
3-8-1 Komaba, Meguro,
Tokyo, Japan
yamaguch@graco.c.u-tokyo.ac.jp

***Abstract*—This Research Full Paper proposes a new method to collect course syllabi. A syllabus is essential information about a course in a university. Students grasp the topics covered by a course through its syllabus, and faculties understand the curriculum offered by the university by a set of syllabi. Thus, a syllabus helps to analyze educational activities. Our previous work proposed a hybrid method that combines Google API as a general keyword search engine and linear support vector machine (SVM) as content-based classification models. We could find more computer science (CS) syllabus pages than using each method alone by employing the hybrid method. This paper extends the hybrid method for finding a directory page with many links to the syllabus pages. We use the hybrid method to collect candidate directory pages and select the true directory pages from the candidates by the three heuristics: (1) Hyperlink-Induced Topic Search (HITS) score. (2) URL pattern, and (3) Content word. (1) HITS score: The relation between directory pages and syllabus pages resembles the relation between the HITS algorithm's hubs and authorities. We use hub scores to select directory pages. (2) URL pattern: Pages with similar contents and roles share a part of their URLs. We exploit this observation to select syllabus pages. (3) Content word: We expect a syllabus page to include words of the Body of Knowledge (BOK) 'Computing Science Curricula CS2013,' released by the ACM and IEEE Computer Society. We use the words extracted from the CS2013 BOK to measure how each candidate syllabus page is related to CS2013. With these three heuristics, we achieved 32%, the percentage of CS syllabus pages included on candidate pages.**

***Index Terms*—Computer Science Curricula, Syllabus Analysis, Web Crawling, Linear SVM, HITS algorithm**

I. Introduction

A syllabus gives essential information about a course offered by a university. For students, a syllabus is one of the most important documents to take a course because the students grasp the topics covered by the course through the syllabus. For faculties, a set of syllabi is one of the ways to tell the curriculum offered by the university to students. In other words, a syllabus is the communication tool between faculties and students. Therefore syllabus analysis is one of the essential techniques in educational engineering.

Many kinds of research have been conducted to analyze and design curricula based on syllabi [1]–[8]. Kousha et al. collected academic course syllabi to evaluate how the textbooks are utilized for teaching [2]. They used Microsoft Bing Search API to search the web pages containing the authors and the titles of the textbooks and some syllabus-related terms such as 'syllabus,' 'course description,' 'module,' and so on. Unfortunately, systems that rely on existing search engine API services do not work due to service termination. Assis et al. described a focused crawler for syllabus web pages that exploit both genre and content of web pages using the cosine similarity function to determine the similarity among the fetched web pages [4]. Ota et al. developed an issue-oriented syllabus classification system that semi-automatically categorizes many syllabus data [3]. Rathod et al. developed classifiers to recognize computer science (CS) syllabi from other web pages, crawled 50 computer science departments in the United States, and collected 100,000 candidate pages [6]. One of the most significant projects to collect syllabi by crawling is the Open Syllabus Project (OSP) [9]. The OSP has used a large amount of computing and human resources to collect syllabi across all fields. One of their purposes is to find popular textbooks in each field. Although many related studies aim to acquire syllabus-like information, we aim to collect a sufficient volume of syllabi from each university to analyze a curriculum. Garshcha et al. used course descriptions from Australian universities to analyze whether the information technology (IT) curriculum meets market needs [8]. If it is possible to collect syllabi from many universities using the study, it will be possible to analyze a broader range of areas.

We have been collecting CS-related syllabi for the analysis of CS curricula. The 'syllabus' we are studying describes the academic field or topic covered by the lecture, often posted with the title 'course description' in syllabus information. For an example of such syllabi, we manually collected more than 3,000 actual syllabi of the course curriculum from the CS departments of the top 47 ranked universities. We discovered the following two key features from the syllabi: 'locality bias' over the world and 'combination of two simple factors' in [10]–[13]. From educational engineering perspective, by comparing the syllabi of many universities with the syllabi of our university, we will be able to identify areas in which our university lacks and complement it, or to discover the characteristics of our own university and differentiate it from others. Such knowledge can be obtained quantitatively and

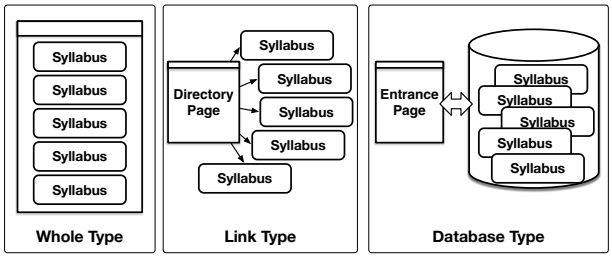


Fig. 1. Three Types of Syllabus Page Structures: Whole type, Link type, and Database type.

statistically from the analysis of syllabi. In addition, by relating the syllabus under analysis to a standard curriculum such as the 'Computing Science Curricula CS2013,' released by the ACM and the IEEE Computer Society, the scope of coverage of the university in question can be further clarified.

For the analysis, we needed several dozen syllabi per university. To obtain syllabi from the universities' websites, it was necessary to locate appropriate web pages. We had to extract texts from the web pages by page structure-specific tools and 'copy and paste' manual operations. This operation took several hours per university.

This operation is very labor-intensive, and we want to automate the process because 47 universities are too small to get a concrete result by statistical analysis. We have to repeatedly get syllabi from the same university to analyze the time evolution of syllabi.

For collecting syllabi, we collected a page that contains the syllabus of a single course (hereafter referred to as a 'syllabus page.') From our experience with 47 universities, we could collect syllabus pages efficiently if we could find a directory page containing links to many syllabus pages. We found a directory page in a dozen candidate pages for each university. Such candidate pages can be found by a semi-automated syllabus crawling method with a combination of Google Search API and linear support vector machine (SVM) reported in [13]. However, the rate of the true syllabus pages to pages obtained by following links in the candidate directory pages (called 'syllabus rate') is as low as 17.0% and insufficient for statistical analysis. This paper improves the syllabus rate to 32.0% by introducing three heuristics.

This paper is organized as follows. In Section II, we explain the findings of the types of pages offering syllabi in our previous study. In Section III, we explain the previous crawling system. In Section IV, we explain three heuristics. In Section V, we report the result of collecting course syllabi related to computer science. In Section VI, we discuss the performance of our heuristics. Section VII concludes this paper.

II. Structures of Syllabus Pages

In our previous work, we picked up the top-ranked universities in the CS field from Times Higher Education, World University Rankings[1] 2018 with computer science (THE2018CS) as a target for collecting syllabi. In order to investigate the structure of the website for semi-automated syllabus collection, we increased the number of target universities from the previous 47 to the top 100 universities in THE2018CS. We attempted to collect syllabus pages, and we could obtain the syllabi of the CS departments of 58 universities. Other 42 universities did not make their English syllabi public.

By examining the collected web pages of the 58 universities, we found that the structures of the syllabus pages at the universities' websites could be categorized into the following three types: Link type, Whole type, and Database type. Figure 1 illustrates these three types conceptually. A *Whole type* consists of a long page (called a *whole page*) including whole syllabi. A *Link type* consists of a page (called a *directory page*) having hyperlinks to syllabus pages. Each linked page contains one syllabus. A *Database type* consists of an entrance (called an *entrance page*) to a database storing all the syllabi of the whole university or a department. We could identify directory pages, whole ones, and entrance ones, which we call *key page*s generically, in the collected pages by linear SVM. For classification, we use a decision model for each type. Please refer to our paper [12] for the details.

III. Semi-automated Crawling System

We have been developing a system to support the syllabus collection process. Figure 2 shows the system architecture. It was implemented in Python using the Google Search API and the general-purpose crawler, Scrapy[2]. The system intends to retrieve a set of syllabus pages for each university. However, it currently outputs a subset of pages corresponding to three types: whole pages of Whole type, directory pages of Link type, and entrance pages of Database type. As the candidates, the system extracts the ten web pages obtained by Google Search API and the two web pages with high confidence scores of linear SVM. In some cases, both Google Search API and linear SVM offer the same pages; therefore, the number of candidates is 10 to 12. Please refer to our paper [13] for the details.

IV. Proposed Heuristics

In the previous work, we obtained three types of candidate key pages that can be used as clues to find the syllabus page. A key page (directory, whole, or entrance page) of the CS syllabus can be found in the top 10–12 candidates obtained by the crawling system in Section III in more than 95% of the universities. However, some of these pages include pages that are not directory pages, and simply following the links in the collected pages result in many non-syllabus pages.

This study aims to obtain the syllabus pages from the candidate pages semi-automatically and efficiently. First, the three types are examined again from semi-automatic extraction.

A Database type is dedicated to human interaction and is unsuitable for automatic crawling. A Whole type has a variety of ways to incorporate the entire syllabi in a single page, and it

[1]https://www.timeshighereducation.com/world-university-rankings

[2]https://scrapy.org/

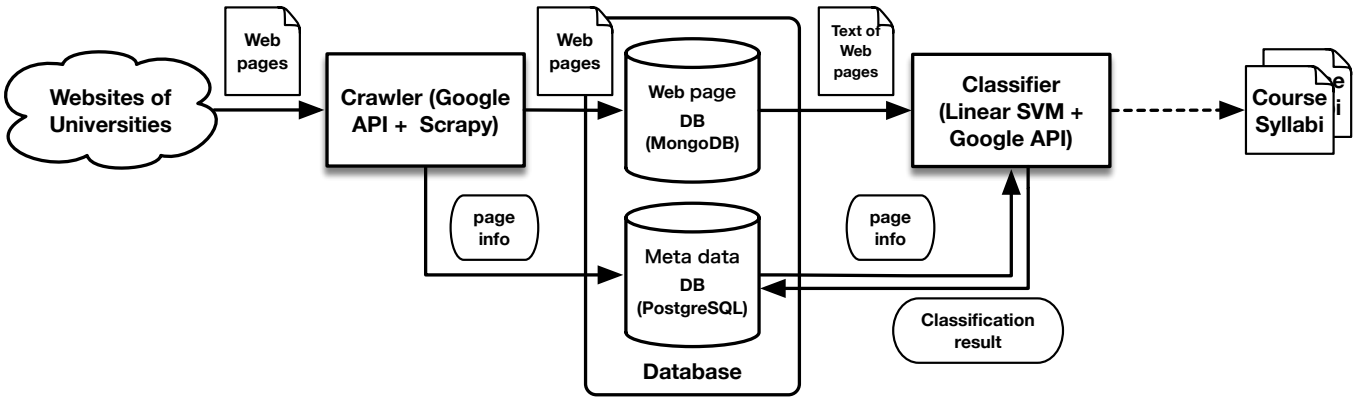


Fig. 2. The System Architecture of The Syllabus Collection Support System [13]

is difficult to decompose it into syllabi in a systematic method. On the other hand, a directory page of a Link type has links to syllabus pages from which syllabi can be extracted easily. Thus, we focus on the Link type. The syllabus rate is around 17.0%, too low to use even for the statistical analysis. To reduce such non-syllabus pages, we introduce the following three heuristics: HITS, URL patterns, and Content words. Figure 3 schematically illustrates how these heuristics work.

a) HITS score: Assuming that there are 'authoritative' pages and 'hub' pages on the Web, that good authoritative pages are linked from many hub pages, and that good hubs have links to many authoritative pages, Kleinberg developed the Hyperlink-Induced Topic Search (HITS) algorithm to discover authoritative pages on the Web. The HITS algorithm calculates two scores, a hub score and an authority score, from a graph that represents the link structure among web pages, with a page with a high hub score being a hub linked to many authoritative pages. In this study, we call the hub score 'HITS score.' Our proposed link type consists of a directory page and several syllabus pages linked from the directory page, which would result in a high HITS score. Therefore, we expect that extracting candidate directory pages with high HITS scores improves the syllabus rate.

b) URL pattern: Because many candidate directory pages contain links to other pages, it is impossible to determine whether a page is a directory page or not only by its number of links. Our previous study observed that the syllabus pages linked by a directory page are often categorized by semesters, mandatory/elective types, and programs. Each category contains a considerable number of syllabus pages. In other words, syllabus pages in different categories are placed in different URL domains to ease the management tasks. To exploit this, we group all the links included in the candidates of directory pages by the common part of the link URL and remove the links whose group has too few links. Specifically, we use the following procedure to obtain candidate syllabus pages.

1) Extract absolute URLs from links in a directory page.
2) Split each URL into the components consisting of the domain name, the path segments, and the query. For example, if the original URL is https://www.example.com/dir1/dir2?name1=value1 we get, `www.example.com`, `dir1`, `dir2`, `name1=value1`.
3) For each URL, find other URLs sharing the longest components from the left with the URL and the number of the URLs exceeds the specified minimum number of URLs. We suppose that there are 30 URLs having common components up to `dir1` and only 15 URLs up to `dir2`. If the minimum number is 20, we output the 30 URLs common up to `dir1` and if that is 10, we output the 15 URLs common up to `dir2`.

The minimum number should be several tens if the directory page contains links to syllabus pages for all courses offered during a single semester or academic year. If the minimum number is too large, pages linked from true directory pages are omitted. Thus, the minimum number is set so large that not all the pages are omitted for any university. Setting an appropriate minimum number of links makes it possible to extract groups of syllabus pages containing a common URL, which is expected to increase the syllabus rate.

c) Content word: We expect that a syllabus page contains more syllabus-related words than the other pages. As syllabus-related words, we use the CS2013 BOK to analyze the CS syllabus [10]. Therefore, the syllabus rate is expected to increase by extracting web pages that contain more BOK words.

V. Experiments

To test the validity of the heuristics in Section IV, we used the 21 university websites from which we found Link Type syllabus pages in our previous study [12]. Table I shows these 21 universities and their URLs. We obtained 12 candidate directory pages for each of these universities using the method in [13]. We could find directory pages for 17 of 21 universities by manually inspecting the pages. We also visited over 10,000 web pages linked from the candidate directory pages, visually determined whether the course descriptions about computer science (CS) were listed on those pages, and considered the included pages "true syllabus pages." Using the judgment results, the syllabus rate obtained by applying the heuristics

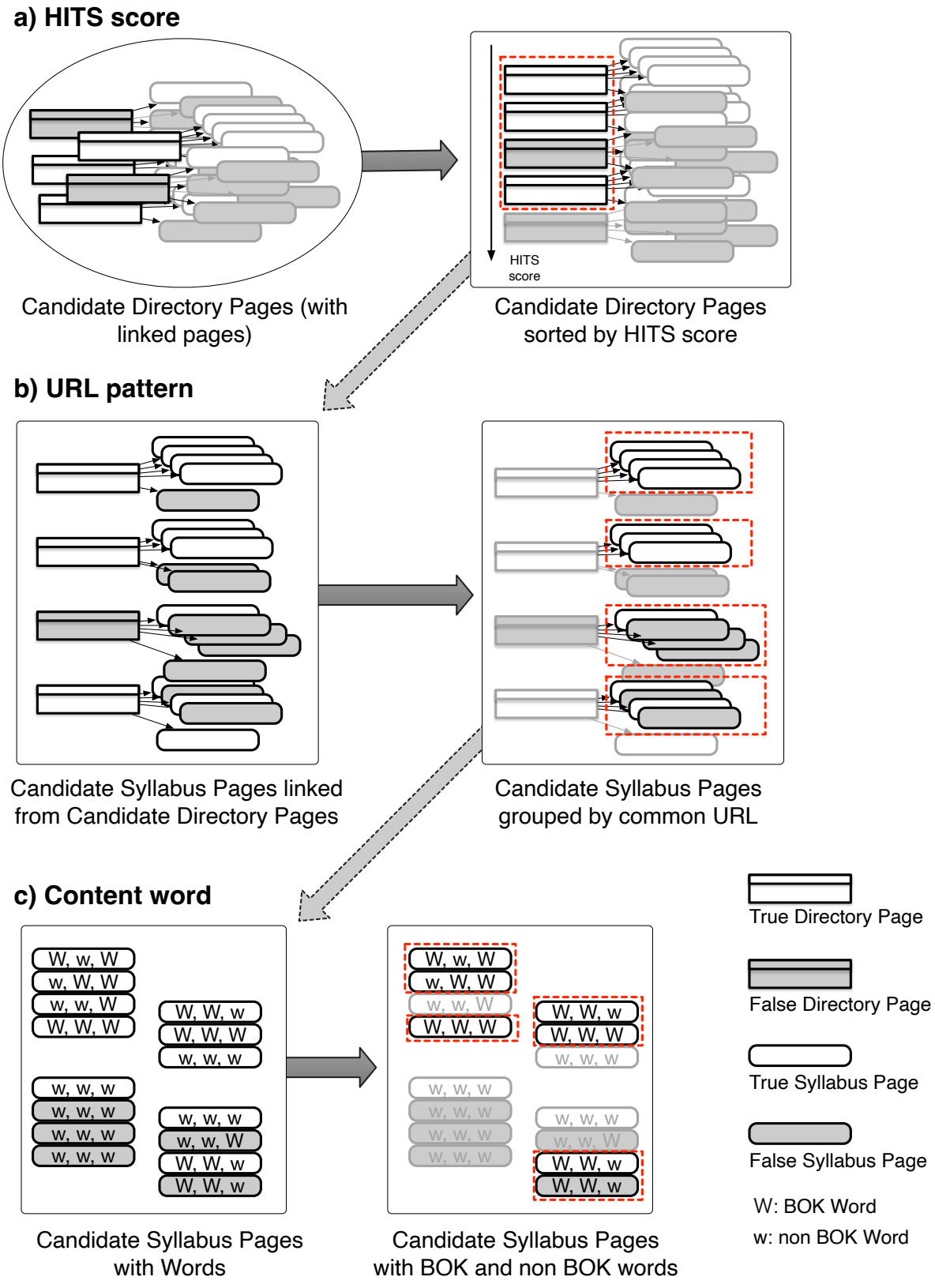


Fig. 3. Three heuristics, a) HITS score, b) URL pattern, and c) Content word

TABLE I
REFERENCE UNIVERSITIES FOR EVALUATION.

University	URL
Australian National University	https://www.anu.edu.au
Queensland University of Technology	https://www.qut.edu.au
Aalto University	https://www.aalto.fi
City University of Hong Kong	https://www.cityu.edu.hk
Seoul National University	https://www.snu.ac.kr
King's College London	https://www.kcl.ac.uk
University of Cambridge	https://www.cam.ac.uk
The University of Edinburgh	https://www.ed.ac.uk
Newcastle University	https://www.ncl.ac.uk
University of Oxford	https://www.ox.ac.uk
University of California & Santa Barbara	https://www.ucsb.edu
University of Southern California	https://www.usc.edu
University of Chicago	https://www.uchicago.edu
University of Illinois at Urbana-Champaign	https://illinois.edu
Dartmouth College	https://dartmouth.edu
Rutgers University	https://www.rutgers.edu
Columbia University	https://www.columbia.edu
Stony Brook University	https://www.stonybrook.edu
University of Pittsburgh	https://www.pitt.edu
Brown University	https://www.brown.edu
The University of Texas at Austin	https://www.utexas.edu

TABLE II
DIRECTORY PAGE RANK(S) BY HITS SCORE.
'—' INDICATES THAT NO DIRECTORY PAGE COULD BE FOUND FOR THE UNIVERSITY

University	Rank(s)
Australian National University	2,3,8
Queensland University of Technology	1,2,3
Aalto University	5,6,7
City University of Hong Kong	7,8,9,10
Seoul National University	1,2,3,4
King's College London	—
University of Cambridge	6,7,8,9,10,12
The University of Edinburgh	—
Newcastle University	1,3,4,11
University of Oxford	5
University of California & Santa Barbara	—
University of Southern California	—
University of Chicago	1,2,3
University of Illinois at Urbana-Champaign	11
Dartmouth College	6
Rutgers University	1,2
Columbia University	1,2
Stony Brook University	1,3
University of Pittsburgh	7,8,11,12
Brown University	1
The University of Texas at Austin	4,5,7

TABLE III
SYLLABUS RATE FOR MINIMUM URLS

minimum URLs in the same domain	total	syllabus	syllabus rate (percentage)	zero university
0	11,180	1,965	17.6%	0
10	6,458	1,776	27.5%	0
20	6,379	1,759	27.6%	0
30	5,597	1,731	30.9%	0
40	5,144	1,634	31.8%	0
50	4,389	1,527	34.8%	3
60	3,956	1,433	36.2%	4

is calculated to determine the effect of the heuristics and find appropriate parameter values.

A. HITS score

Using the method in Section III, we obtained 12 candidate directory pages for the 21 universities in Table I. Then, a graph (called a *local graph*) is generated using the candidate directory pages and the pages linked by those pages. We applied the HITS algorithm to this graph and got the hub score for each page. Table II shows the ranks of the true directory pages by hub score for each university. As this result shows, the hub score of the directory pages was high, and all directory pages were among the top 12 pages according to the hub score. Note that the hub score of the true directory pages for the local graph is higher than that for the graph of the whole university website.

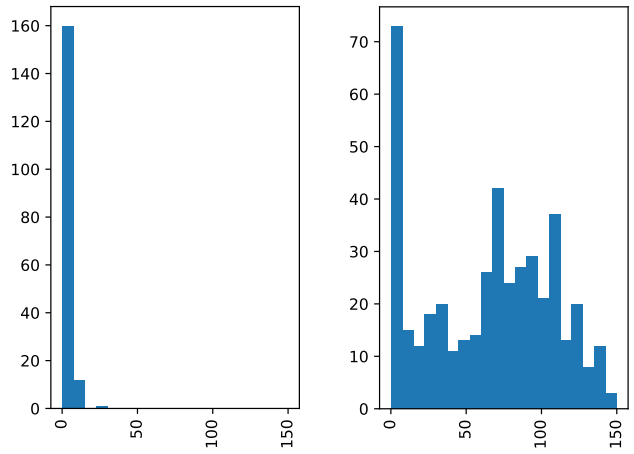
Of the 47 directory pages, half were ranked in the top five, and three-quarters were ranked in the top eight. Therefore, we used the pages ranked in the top eight.

B. URL pattern

Table III shows the syllabus rates for the minimum number of URLs in the same domain every ten from 0 to 60. In this table, 'minimum URLs in the same domain' is the minimum number, 'total' is the number of candidate pages for the minimum number, 'syllabus' is the number of true syllabus pages, 'syllabus rate' is syllabus/total, and 'zero university' is the number of universities that had no candidate page for the minimum number. In the range 0 to 60, the rate of syllabus pages increases as the minimum number increases, but 'zero university' increases also. From this trade-off, we set the minimum number to 40.

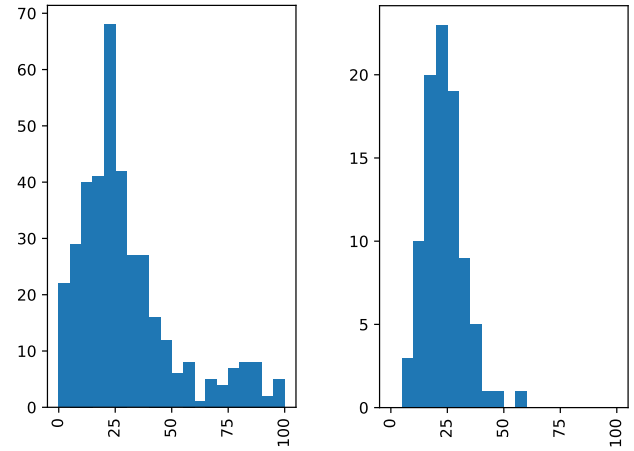
C. Content word

The number of word types in CS2013 was investigated for each syllabus and non-syllabus page. Figures 4 and 5 show histograms based on the number of CS2013 word types in the candidate syllabus pages. The horizontal axis is the number of word types, and the vertical axis is the frequency of the number in the non-syllabus pages (left) and the syllabus pages (right). As these figures show, the syllabus pages include more CS2013-related word types than the non-syllabus pages. We set the minimum number of word types to output pages to utilize this difference for syllabus page identification. We calculated the syllabus rate by varying the minimum number to determine an appropriate minimum number. Table IV shows the result. 'minimum word type' is the minimum number of the word types, 'syllabus' is the number of true syllabus pages, 'syllabus rate' is syllabus/total, and 'zero university' is the number of universities that had no candidate page for the minimum number. We slightly improved the rate to 32.0% from 31.8% by the minimum word types of 20.



(a) non-syllabus page (b) syllabus page

Fig. 4. CS2013 Word Type Distribution for University of Cambridge



(a) non-syllabus page (b) syllabus page

Fig. 5. CS2013 Word Type Distribution for The University of Texas at Austin

TABLE IV
SYLLABUS RATE FOR MINIMUM WORD TYPES

minimum word types	total	syllabus	syllabus rate (percentage)	zero university
0	5,135	1,634	31.8%	0
5	4,711	1,486	31.5%	0
10	4,448	1,389	31.2%	0
20	3,901	1,249	32.0%	0
30	3,263	1,017	31.2%	0
40	2,790	826	29.6%	0
50	2,194	453	20.6%	1
100	1,300	189	14.5%	2

VI. DISCUSSION

The simple average of syllabus rates for 21 universities was 17.0%. Our heuristics improves it to 32.0%. The HITS score improves 17.0% to 17.6%, the URL pattern does 17.6% to 31.8%, and the Content word does 31.8% to 32.0%. These are shown in the corresponding part of Figure 6. We discuss the effect of these heuristics more closely.

Table V shows the result of the URL pattern heuristics

TABLE V
SYLLABUS RATE FOR MINIMUM URLS WITHOUT HITS SCORE

minimum URLs in the same domain	total	syllabus	syllabus rate (percentage)	zero university
0	13,855	2,369	17.1%	0
10	8,036	2,164	26.9%	0
20	7,788	2,133	27.4%	0
30	6,997	2,107	30.1%	0
40	6,514	2,000	30.7%	0
50	5,578	1,866	33.5%	2
60	5,054	1,769	35.0%	4

TABLE VI
SYLLABUS RATE FOR MINIMUM WORD TYPES WITHOUT HITS SCORE

minimum word type	total	syllabus	syllabus rate (percentage)	zero university
0	6,501	2,000	30.8%	0
5	5,910	1,852	31.3%	0
10	5,558	1,726	31.1%	0
20	4,866	1,576	32.4%	0
30	4,093	1,324	32.3%	0
40	3,439	1,065	31.0%	0
50	2,718	625	23.0%	1
100	1,495	247	16.5%	2

without applying the HITS score heuristics, and the best achievable rate without a university having no candidates is 30.7%. Compared with the result in Table III, the best achievable rate is 1.1% lower. Even though the improvement of the syllabus rate by the HITS score heuristics is only 0.6%, the HITS score heuristics make candidate directory pages favorable to the URL pattern heuristics.

Table VI shows the rate for URL pattern heuristics and Content word heuristics. Comparing the syllabus rates in Table VI and Table IV, Table IV is higher when the minimum number of word types is small (0, 5, 10), but Table VI is higher when the threshold is even larger. It was unexpected that the syllabus rate increased when only URL pattern and Content word were used without applying HITS score heuristics. We checked the syllabus rate of each target university and found this phenomenon is primarily influenced by City University of Hong Kong and University of Cambridge, where multiple directory pages are excluded by HITS score heuristics. A further detailed examination of the number of candidate syllabus pages for both universities revealed that more true syllabus pages were included without HITS heuristics and that the number of non-syllabus pages of both universities decreased when Content word heuristics were used. In particular, the University of Cambridge has a larger number of both directory pages and syllabus pages than any other university, and the fact that Content word heuristics can almost completely exclude non-syllabus pages from the total contributes significantly to the total. The effects of both the HITS score and the Content word are smaller than that of the URL pattern, and they seem to cancel each other out depending on the combination and the target university data. Further research is needed to detect which combination of heuristics is effective for each university website to improve semi-automatic syllabus crawling.

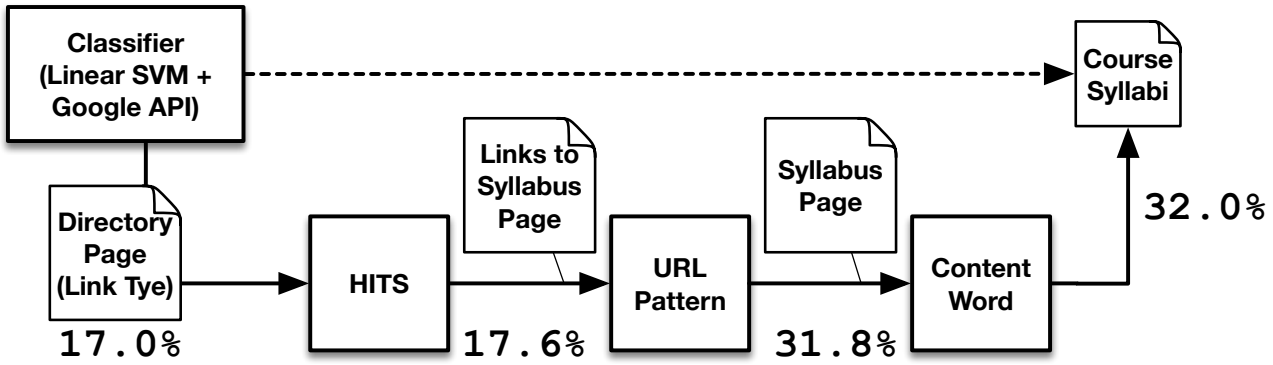


Fig. 6. Syllabus page extraction process (this work). This process is the elaboration of the classifier to the course syllabus (dashed arrow) of Figure 2. The rates (in percentage) of true syllabus pages are the averages for all the universities. For a directory page, the rate is calculated for the existing syllabus pages linked from the directory page.

TABLE VII
SYLLABUS RATE FOR MINIMUM URLS (WITH TOP-FOUR RANKED PAGES IN HITS SCORE)

minimum URLs in the same domain	total	syllabus	syllabus rate (percentage)	zero university
0	6,743	1,092	16.2%	0
10	4,345	1,044	24.0%	0
20	4,340	1,050	24.2%	0
30	3,737	1,022	27.3%	1
40	3,516	967	27.5%	4
50	3,080	955	31.0%	4
60	2,747	915	33.3%	5

TABLE VIII
SYLLABUS RATE FOR MINIMUM WORD TYPES (WITH TOP-FOUR RANKED PAGES IN HITS SCORE)

minimum word type	total	syllabus	syllabus rate (percentage)	zero university
0	4,336	1,050	24.2%	0
5	4,120	1,026	24.9%	0
10	3,875	966	24.9%	0
20	3,364	848	25.2%	0
30	2,794	675	24.2%	0
40	2,394	545	22.8%	0
50	1,847	211	11.4%	1
100	1,036	91	8.8%	2

As we described in Section V-A, we use the pages ranked in the top eight based on the results in Table II. To check whether HITS can extract directory pages that have links to many syllabus pages, we use the pages ranked in the top four to see if the syllabus rate improves. Table VII shows the change in the syllabus rate when the top four candidate directory pages in the HITS score are selected. Table VIII shows the rate for URL pattern heuristics and Content word heuristics. The syllabus rate decreases when URL pattern and Content word heuristics are applied to the top four HITS score pages as candidates for the directory page.

VII. CONCLUSION

We have been developing a system to support the syllabus collection process. In this paper we introduced three heuristics, HITS score, URL pattern, and Content word, to improve the collection rate of syllabus pages. With these three heuristics, we achieved the syllabus rate of 32%.

For future works, we will examine how the heuristics are effectively combined because the heuristics applied in this study have different contributions to improving the syllabus rate. Our long-term research objective was to quantitatively and statistically analyze the CS curriculum using syllabi, but we found that manually collecting syllabi was time-consuming and labor-intensive, so we began working on semi-automatic syllabus crawling. Although we improved the syllabus rate to some extent, including this result, we also found that it is not easy to extract only syllabi from university websites automatically. To return to our original objective, we need to verify whether the candidate syllabus pages obtained by our method can discover similarly valuable features when manually collected syllabi are used.

REFERENCES

[1] X. Yu, M. Tungare, W. Fan, Y. Yuan, M. Perez-Quinones, E. A. Fox, W. Cameron, and L. Cassel, *Automatic Syllabus Classification using Support Vector Machines*. IGI Global, 2009, pp. 61–74.

[2] K. Kousha and M. Thelwall, "An automatic method for assessing the teaching impact of books from online academic syllabi," *Journal of the Association for Information Science and Technology*, vol. 67, pp. 2993–3007, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/asi.23542

[3] S. Ota and H. Mima, "Machine Learning-based Syllabus Classification toward Automatic Organization of Issue-oriented Interdisciplinary Curricula," *Procedia - Social and Behavioral Sciences*, vol. 27, pp. 241–247, 2011.

[4] G. T. d. Assis, A. H. Laender, M. A. Gonçalves, and A. S. d. Silva, "Exploiting Genre in Focused Crawling," in *International Symposium on String Processing and Information Retrieval*. Springer, 2007, pp. 62–73.

[5] R. Gluga, J. Kay, and R. Lister, "PROGOSS: Mastering the curriculum," in *Proceedings of The Australian Conference on Science and Mathematics Education (formerly UniServe Science Conference)*, 2012, pp. 92–98.

[6] N. Rathod and L. Cassel, "Building a Search Engine for Computer Science Course Syllabi," in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, ser. JCDL '13. New York, NY, USA: ACM, 2013, pp. 77–86. [Online]. Available: http://doi.acm.org/10.1145/2467696.2467723

[7] C. Szabo and K. Falkner, "Neo-piagetian Theory As a Guide to Curriculum Analysis," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. ACM, 2014, pp. 115–120.

[8] P. Garscha and A. Wöhrer, "IT Curricula Versus Labour Market Requirements in the Area of Cloud Computing in Austria," in *IFIP Advances in Information and Communication Technology*, vol. 595 IFIP, 2020.

[9] The Open Syllabus Project, "The Open Syllabus Project – Opening the curricular black box," 2004. [Online]. Available: http://opensyllabusproject.org/

[10] T. Sekiya, Y. Matsuda, and K. Yamaguchi, "Curriculum Analysis of CS Departments based on CS2013 by Simplified, Supervised LDA," in *ACM International Conference Proceeding Series*, vol. 16-20-Marc, 2015.

[11] Y. Matsuda, T. Sekiya, and K. Yamaguchi, "Curriculum Analysis of Computer Science Departments by Simplified, Supervised LDA," *Journal of Information Processing*, vol. 26, pp. 497–508, 6 2018.

[12] T. Sekiya, Y. Matsuda, and K. Yamaguchi, "Investigation on University Websites for Semi-automated Syllabus Crawling," *2019 IEEE Frontiers in Education Conference (FIE)*, vol. 1, pp. 1–7, 10 2019.

[13] T. Sekiya, T. Tatejima, Y. Matsuda, and K. Yamaguchi, "A Proposal for a Hybrid Syllabus Search Tool that Combines Keyword Search and Content Based Classification," in *IEEE Global Engineering Education Conference (EDUCON)*, vol. 2021-April, 2021.