Automatic Course Website Discovery from Search Engine Results

Rui Meng[1], Zexin Zhao[1], Yu Chi[1] and Daqing He[1]
[1]School of Information Sciences, University of Pittsburgh, USA

Abstract
With the rapid development of Internet Technology, the forms of education have been undergoing drastic changes. Instructors are used to posting teaching materials on course websites and setting them publicly accessible. Thus large amounts of course resources have been well organized and shared, which also provide possibilities for building knowledge graphs for a specific domain. However, so far no specific method has been developed for collecting online course resources. In this paper, we propose a method to identify course websites by filtering search results from a general search engine. Experiment results show that the proposed method could achieve good performances on both within-domain and cross-domain tasks, which lays a solid foundation for further work on mining and integrating the online educational resources.

Keywords: course website; website identification; search results filtering; web mining; resource discovery; online educational resource

Citation: Meng, R., Zhao, Z., Chi, Y. & He, D. (2017). Automatic Course Website Discovery from Search Engine Results. In iConference 2017 Proceedings, Vol. 2 (pp. 142-146). `https://doi.org/10.9776/17353`

1 Introduction

The dramatic increase of online educational resources has provided great opportunities to innovate new ways for learning and teaching. For example, students can enhance their understanding of course knowledge after class by reading the online resources from other similar courses. When designing a new course, an instructor can also come out useful ideas if he/she is referred to the course syllabus from other instructors. More importantly, the contributors of online courses, most of whom are instructors, are usually experts of one domain. When presenting a certain knowledge concept, they usually provide various related materials including textbooks, slides, readings, videos along with syllabus to augment students' comprehensive understanding of the concept (Meng, Han, Huang, He, & Brusilovsky, 2016). As a result, a proper linking of online educational resources is of great important, for both learning and teaching purposes (Huang, Yudelson, Han, He, & Brusilovsky, 2016; Wang et al., 2016; Land & Greene, 2000; Liu, Jiang, & Gao, 2015).

Among the given variety of educational resources, this paper focuses on the online course websites since they provide more comprehensive organizations of knowledge within a domain. To achieve this goal, our first step is to build a data collection for online courses. However, to the best of our knowledge, there is no existing work regarding this topic in academia. Though the general search engines like Google can serve as an effective channel for collecting such resource at a low cost, their results usually consist of too much noise. This motivates us to invent new automated approaches to re-rank or filter the noised return results. Specifically, we first collect a large amount of course web pages from search engine results; then, we identify whether a result page is a course page or not based on classification (Käki & Aula, 2005; Brin & Page, 2012). Our classification approach utilizes both content features and layout features. The content feature attempts to capture the way how the content of a web page is written. We assume a course web page is more likely to include course-dependent words such as syllabus, week, schedule, reading and etc. than other webpages. The layout feature, on the other hand, tries to analyze how a webpage is presented to users. We observe that the course webpages usually contain several tables and listed items.

The remaining of the paper is organized as follows: in Section 2, we present the definition of our task and the feature design. The experiment setup and its corresponding results are provided in Section 3. At last, we provide our conclusion, as well as limitations and future directions.

2 Methodology

2.1 Task Definition

We define the course website as a site that presents detailed content regarding a specific course. The content includes the course description, syllabus, schedule, textbook, reading and so on. We do not consider a pure course introduction page as a course website in this study since it only provides very limited course content. Specifically, our approach includes two steps for building online course data collection, as listed below.

– Step I: Retrieves a set of course website candidates $C = \{c_1, c_2, c_3, \ldots, c_m\}$ (m is a predefined number indicating the depth of search results we went through) from a search engine using a query q. We will explain the ways of choosing query q in Section 3.1.

– Step II: As for each website candidate c_i in C, a binary classifier is adopted to determine whether c_i is a course website or not. The classifier is trained on a manually-labeled dataset.

2.2 Feature Design

In the classification step (i.e., Step II), we extract a set of n features $F = f_1, ..., f_n$. According to a previous study (Dong, Qi, & Gu, 2005), both structure layout features and content features play important roles for identifying domain-specific websites. Inspired by their work, both of these two features are employed in our study, as listed below.

– For the **content feature**, we adopted the Bag-of-Word (BOW) feature on the textual content. The BOW feature is a widely-adopted feature for variety of text classification tasks (Sriram, Fuhry, Demir, Ferhatosmanoglu, & Demirbas, 2010; Lodhi, Saunders, Shawe-Taylor, Cristianini, & Watkins, 2002). Note that for each feature (one word), we adopt the TF-IDF (Term Frequency - Inverse Document Frequency) to indicate the feature importance, expecting that it helps remove the domain-dependent features and makes our method be generalized on cross-domain scenarios.

– We expect the **structural feature** would provide additional information for classification. For example, in the context of course website, HTML tags like <table> and <li> appear frequently in syllabus pages.

3 Experiment

3.1 Data Collection

Four courses were selected in this paper: *Database, Java Programming Language, Information Retrieval* and *Human Computer Interaction.* These four courses cover important topics in Information Science domain, which widely-recognized as important research topics and taught in most iSchools. The first two courses are relatively basic and the latter two are advanced topics.

We collected 200 search results for each course from Google search (800 website pages in total). We examine three different ways of composing search queries: *CourseName*, *CourseName + course* and *CourseName + syllabus*, and finally choose *CourseName + syllabus* because of its relative good performance of retrieving relevant results from Google.

To train the classifier, we need a manually-labeled ground truth. This was obtained from manual labeling of two authors. The labeling on *Java Programming Language* is used for examining the agreement, in which we find a Kappa coefficient of 0.7465, indicating a fairly high agreement. The other three courses are then labeled separately.

Eventually, 800 course website candidates are labeled and the descriptive statistics is summarized in Table 1. Figure 1 shows the distribution of course websites in top 100 search results, from which we can see that the course websites only represent a small share of all results, and may occur in all different ranges of ranking positions. This further indicates the importance of putting extra efforts on result filtering.

The noise pages of each course are quite different. In *Human-Computer Interaction*, a large share of noises are about HCI degree introduction instead of course page. As for *Java Programming Language*, most noises are company advertisements and tutorials, but some companies also offer high-quality course websites.

Course Name	Total Number	Number of Course Websites
Database (DB)	200	44
Human Computer Interaction (HCI)	200	50
Information Retrieval (IR)	200	65
Java Programming Language (Java)	200	51

Table 1: The Statistics of labeled ground truth

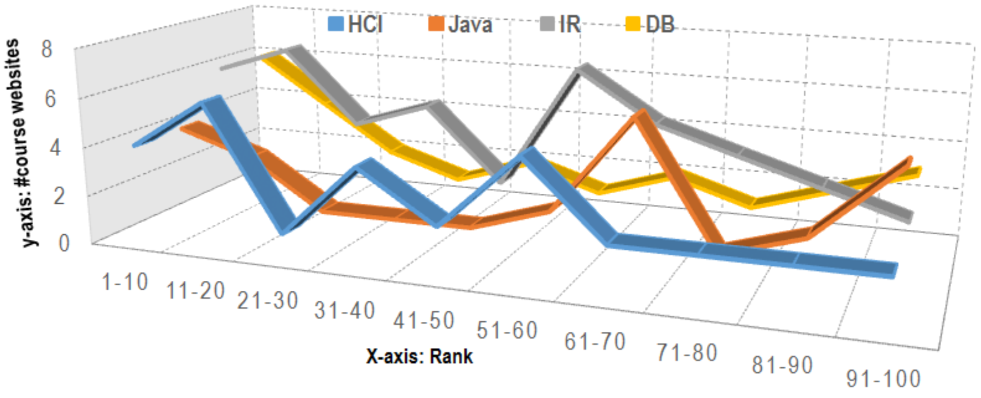

Figure 1: Course website distribution in top 100 Google search results. X-axis: the ranking positions of course websites on the Google search result page. Y-axis: number of course websites.

3.2 Experiment Setting

The classification performance reported in this paper was based on the average of 5-fold cross-validation. Training and testing data was randomly generated on the dataset by 50 times. The Support Vector Machine with L2 regularization was employed (Yu, Ho, Juan, & Lin, 2013). We mainly report the F-1 score of positive class (positive indicates that a given website is a course website). And the reported performance is based on the average of 50 testing runs on optimal classifier parameters.

As mentioned above, we extracted both BOW and HTML structure features. Since a pure HTML structure feature based classifier does not perform well on most of the data collection. Here, we report the performances based on two feature groups: BOW and BOW + HTML. In the following sections, we consider two tasks. First, we built one classifier for each course, i.e., we split the dataset for one course into training and testing. Therefore, we can learn four classifiers and predict each course separately.

3.3 Within-course Website Identification

The performance of within-course prediction can be illustrated in Figure 2. The classifier achieves a fairly good performance on all the courses, especially on the course *Java Programming Language*. This might because of the relatively clear difference between its noises and the real course websites. It is easier for the classifier to differentiate websites of interest from others. Affected by the degree introduction websites under the similar titles, the *Human-Computer Interaction* fails in making a good result. As for the feature comparison, by adding additional HTML tags into the features, results of *Information Retrieval* and *Java Programming Language* improve significantly than the results based on only textual content features. Surprisingly, the performance of *Database* drops drastically by adding HTML features, and details show that the accuracy goes from 0.7428 up to 0.7928 but the recall goes from 0.9030 down to 0.7394. The HTML features help rule out irrelevant results at the cost of decreasing recall.

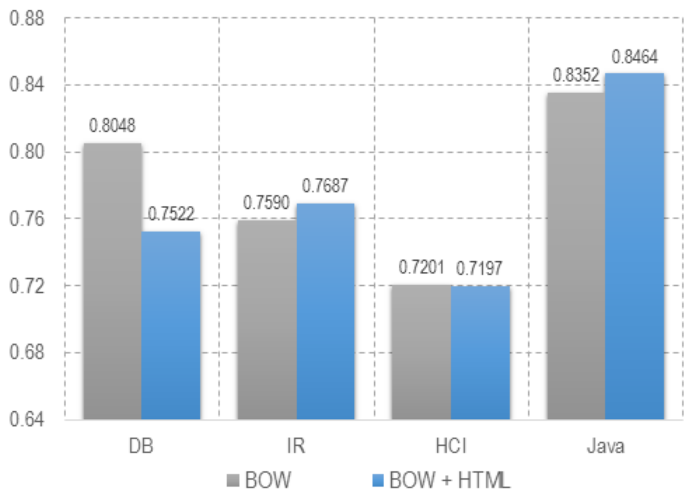

Figure 2: Performance in within-course website identification.

3.4 Cross-course Website Identification

Besides the regular within-course scenario, we are highly interested in the performance on the cross-course task. As we cannot label data for every course, whether our model can generalize well on other courses becomes a critical issue. In this set of experiments, we fix the same test sets which are used in within-course evaluations. This means we can compare the performance of the cross-course task to the within-course. But here we change the training sets to other courses to implement the transfer learning. Here we only report the results based on Whole HTML feature set. The experiment on another feature set shows similar results.

The experiment results are shown in Figure 3. In each course cluster, *By Self* is the performance same to 3.3, which is based on 5-fold cross-validation on one course. The **By Coursename** means the model is trained on the whole dataset of *corresponding course*, and **By All Three** means model trained on all three courses except the tested one. We can see that for most cases the model can be applied to another domain without losing to much performance, which indicates the generalization ability of our features. The high score of **By All Three** shows the benefits from abundant data. However the *Java* cluster presents a different pattern. Due to the characteristic of Java websites (unique content and many course websites maintained by businesses), the **By All Three** achieves a similar score to **By Java**.

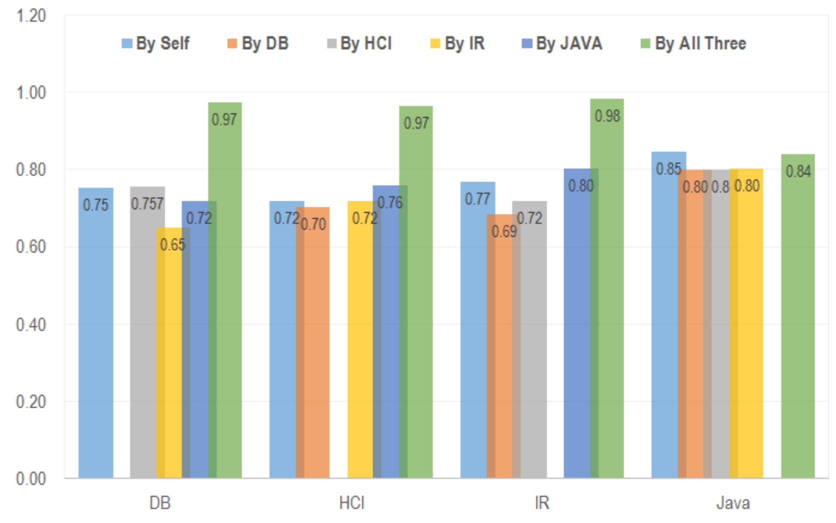

Figure 3: Performance in cross-course website identification.

4 Conclusion

The numerous of course websites provide massive amounts of valuable educational resources, but it remains challenging to differentiate the course websites from others. In order to solve the problem of collecting course websites, we propose a method to identify course website by filtering search results from a general search engine. Experiment results show that the proposed method could achieve good performances on both within-domain and cross-domain tasks. The robust performance and the strong generalization ability indicate the effectiveness of our proposed methods. There are several interesting directions to explore in the future, including exploring our approach on a larger scale and looking for effective methods for capturing domain-independent features. In addition, we are interested in working on potential applications such as online educational resources integration and recommendation.

References

Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, *56*(18), 3825–3833.

Dong, B., Qi, G., & Gu, X. (2005). Domain-specific website recognition using hybrid vector space model. In *International conference on web-age information management* (pp. 840–845).

Huang, Y., Yudelson, M., Han, S., He, D., & Brusilovsky, P. (2016). A framework for dynamic knowledge modeling in textbook-based learning. In *Proceedings of the 2016 conference on user modeling adaptation and personalization* (pp. 141–150).

Käki, M., & Aula, A. (2005). Findex: improving search result use through automatic filtering categories. *Interacting with Computers*, *17*(2), 187–206.

Land, S. M., & Greene, B. A. (2000). Project-based learning with the world wide web: A qualitative study of resource integration. *Educational Technology Research and Development*, *48*(1), 45–66.

Liu, X., Jiang, Z., & Gao, L. (2015). Scientific information understanding via open educational resources (oer). In *Proceedings of the 38th international acm sigir conference on research and development in information retrieval* (pp. 645–654).

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, *2*(Feb), 419–444.

Meng, R., Han, S., Huang, Y., He, D., & Brusilovsky, P. (2016). Knowledge-based content linking for online textbooks. In *2016 ieee/wic/acm international conference on web intelligence* (Vol. 1, pp. 13–16).

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international acm sigir conference on research and development in information retrieval* (pp. 841–842).

Wang, S., Ororbia, A., Wu, Z., Williams, K., Liang, C., Pursel, B., & Giles, C. L. (2016). Using prerequisites to extract concept maps fromtextbooks. In *Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 317–326).

Yu, H., Ho, C., Juan, Y., & Lin, C. (2013). Libshorttext: A library for short-text classification and analysis. *Rapport interne, Department of Computer Science, National Taiwan University. Software available at http://www. csie. ntu. edu. tw/˜ cjlin/libshorttext*.