Using Automatic Metadata Extraction to Build a Structured Syllabus Repository

Xiaoyan Yu[1], Manas Tungare[1], Weiguo Fan[1], Manuel Pérez-Quiñones[1], Edward A. Fox[1], William Cameron[2], GuoFang Teng[2], and Lillian Cassel[2]

[1] Virginia Tech, Blacksburg VA 24060, USA.
{xiaoyany, manas, wfan, perez, fox}@vt.edu,
WWW home page: http://doc.cs.vt.edu/
[2] Villanova University, Villanova PA 19085, USA.
{william.cameron, guofang.teng, lillian.cassel}@villanova.edu

Abstract. Syllabi are important documents created by instructors for students. Students use syllabi to find information and to prepare for class. Instructors often need to find similar syllabi from other instructors to prepare new courses or to improve their old courses. Thus, gathering syllabi that are freely available, and creating useful services on top of the collection, will yield a digital library of value for the educational community. However, gathering and building a repository of syllabi is complicated by the unstructured nature of syllabus representation and the lack of a unified vocabulary in syllabus construction. In this paper, we propose an intelligent approach to automatically annotate freely-available syllabi from the Web to benefit the educational community by supporting services such as semantic search. We discuss our detailed process for converting unstructured syllabi to structured representations through effective information recognition, segmentation, and classification. Our evaluation results proved the effectiveness of our extractor and also suggested a few aspects in need of improvement. We hope our reported work will also benefit people who are interested in building other genre specific repositories.

1 Introduction

A course syllabus is the skeleton of a course. One of the first steps taken by an educator in planning a course is to construct a syllabus. Later, a syllabus can be improved by borrowing information from other relevant syllabi. Students prepare for a course by reading a course syllabus to identify textbooks. Students may use the syllabus to identify course policies, assignment deadlines, etc., during a school semester. Typically, a syllabus sets forth the objectives of the course. It may assist students in selecting electives and help faculty identify courses with goals similar to their own. In addition, a life-long self-learner identifies the basic topics of a course and the popular textbooks by comparing syllabi from different universities. A syllabus is thus an essential component of the educational system.

Unfortunately, searching for a syllabus on the Internet using a generic search engine is an error-prone process and often yields too many non-relevant links. Free and fast access to a collection of syllabi could have a significant impact on education. Furthermore,

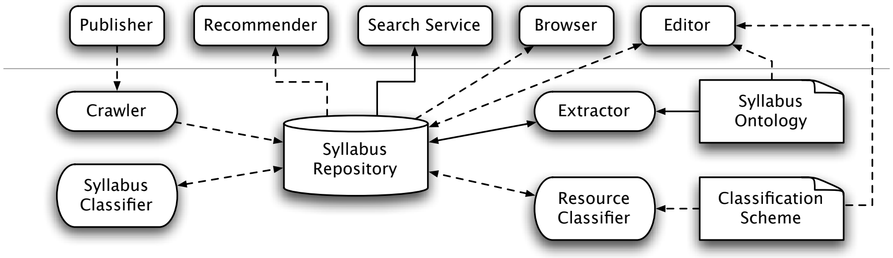


Fig. 1. System architecture.

structured search and retrieval of information from a syllabus is a desired goal if we are to support all the activities mentioned above. One positive example is the MIT OpenCourseWare[3] project which provides free access to MIT course materials to educators, students, and self-learners. It collects 1,400 MIT course syllabi and publishes them in a uniform format. The project is a good start towards making a wide and open digital library of syllabi.

However, there exists a chicken-and-egg situation regarding the adoption of such a repository on a larger scale: there is little incentive for instructors to take the additional effort to add their syllabi to this repository unless there are existing services that they can then use. On the other hand, useful services would need a large dataset of syllabi to work on. Hence, to break out of this deadlock, we decided to seed our repository with syllabi crawled from the Web in order to bootstrap the process. We are creating a digital library of computer science syllabi, detailed in [1]. We restrict our focus to computer science syllabi offered by universities in the USA as a starting point of our proof-of-concept project. The methodology and the system could be extended easily to other disciplines and locations.

To allow semantically rich queries over our syllabus collection, for example, which textbooks are usually used for data structure courses, a key tool is the use of metadata to explicitly annotate syllabi.requires explicit annotation with appropriate metadata. This is the general goal of the Semantic Web. However, two obstacles hinder this goal with respect to the syllabus genre. First, no metadata standard is specific to the syllabus genre although markup schemes, such as IEEE LOM [2], exist for education resources. We are able to annotate a document as a syllabus by the LOM's resource type property but still unable to annotate a piece of information inside a syllabus as a textbook by any of the available metadata standards. Second, it will require too much effort to annotate information inside manually and no approach is available to automate the process of information extraction from the syllabus genre. Motivated by the above observations, we propose a taxonomy and an extraction approach specific to the syllabus genre, and extend our syllabus digital library by extracting metadata from each syllabus and supporting semantic search upon them.

[3] http://ocw.mit.edu/

The overall infrastructure of building our structured syllabus digital library, shown in Figure 1, has several major components. A crawler [3] creates a potential syllabus collection as a result of searching the World Wide Web. It also allows individuals to submit URLs to be included in the next crawl. A syllabus classifier [1] filters noise from the potential syllabus repository. With the guidance of a syllabus taxonomy, an extractor generates structured information (e.g., course name, textbook, course description, etc.) from the unstructured syllabi. People can input and correct their syllabi through our editor service. The syllabi are also categorized to facilitate our browsing service according to a variety of classification schemes such as the ACM computing classification system[4]and the computing ontology [4]. The system provides an interface for users to search for syllabi, as well as several information syndication options.

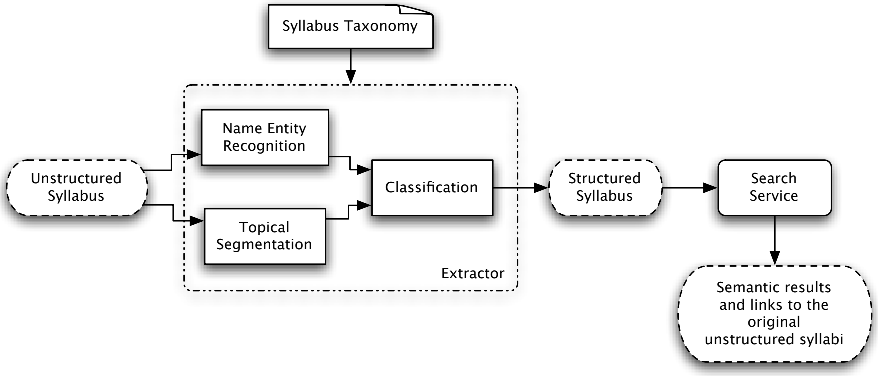


Fig. 2. Workflow from a unstructured syllabus to a structured syllabus.

The focus of this paper is the core of building a genre-specific structured repository shown as the solid flow in Figure 1. The flow is the transformation from an unstructured syllabus to a structured syllabus and then the retrieval of structured syllabi. Figure 2 shows the major modules involved in the process. Following the syllabus taxonomy, semantic information can be extracted from a syllabus, which becomes part of the Semantic Web[5]. The name entity recognition module identifies entities such as people and dates. The topical segmentation module identifies the boundary of a syllabus component such as a course description or a grading policy. The classification module finally associates a list of syllabus properties with the segmented values and stores them in the structured syllabus repository. These three modules work together as the extractor of our system. The search service indexes structured syllabi and provides semantic search results in RDF [6] and links to the raw syllabi.

[4] http://www.acm.org/class/

[5] http://www.w3.org/2001/sw/

[6] http://www.w3.org/RDF/

There are many other genres of unstructured data on the Web; thus, building genre-specific structured repositories presents the opportunity to use such data in innovative applications. We hope that our application of machine learning techniques to extract and obtain a structured repository of genre-specific data will encourage the creation of similar systems for other genres.

2 Related Work

There are a few ongoing research studies on collecting and making use of syllabi. Started from reading lists of a small set of manually collected computer science syllabi in the operating systems, information retrieval, and data mining courses, Neves [5] developed three coherent literature collections to test his ideas of literature-based discovery. A small set of Digital Library course syllabi was manually collected and carefully analyzed, especially on their reading lists, in order to define the Digital Library curriculum [6]. The MIT OpenCourseWare project manually collects and publishes 1,400 MIT course syllabi for public use. A lot of effort from experts and faculty is required in manual collecting approaches, which is the issue that our approach tries to address.

Some have addressed the problem of lack of standardization of syllabi. Along with a defined syllabus schema, SylViA [7] supports a nice interface to help faculty members construct their syllabi in a common format. More work has been done on defining the ontology or taxonomy of a variety of objects. For example, the ontology of a learner, especially in a remote learning environment, describes the features of the learner and distributed fragments of his/her information such as e-portfolios from the college and the first job, respectively [8]. Our proposed syllabus taxonomy also describes the features of a course, such as the course instructor, textbooks, and topics to be covered. We will use these features to provide additional services such as recommending resources from the National Science, Technology, Engineering, and Mathematics Education Digital Library (NSDL)[7] resources for students of a particular course.

In order to fulfill a general goal of the Semantic Web, annotation and semantic search systems have been successfully proposed for other genres (such as television and radio news [9]). Such systems vary more or less with different genres due to their own characteristics and service objectives. To our knowledge, there is no specific annotation and semantic search system for the syllabus genre.

Much work has been done on metadata extraction from other genres such as academic papers. For example, Han *et al.* [10] described a Support Vector Machine (SVM) classification-based method for metadata extraction from a paper's header field. Takasu *et al.* [11] described bibliography field extraction using a Hidden Markov Model. We will describe more related studies when introducing our extraction work in Section 4.

3 Syllabus Taxonomy

Our syllabus taxonomy is designed to help reconcile different vocabularies for a syllabus used by different instructors. For example, instructors often start a course description with headings such as *'Description'*, *'Overview'*, or *'About the Course'*. Such

[7] http://nsdl.org/

variations make it difficult to reuse information from these syllabi. It is also very hard to locate a particular syllabus section because the section headings are not uniquely named. In order to facilitate processing of syllabi by different applications, we propose a syllabus taxonomy[8] developed with the aid of the ontology editor Protégé[9].

The first level of the taxonomy is shown below. Among these 17 properties, some are data types of a syllabus such as `title` (a course title) and `description` (a course description) while others are object types such as `teachingStaff` and `specificSchedule` that utilize other vocabularies at a deeper level. For example, a `courseCode` is defined as an abbreviation of the department offering the course and a number assigned to the course, and a `prerequisite` is composed of one or more `courseCode` objects. It is also worth noting that we define a `specificSchedule` as topics and specific dates to cover them, and a `generalSchedule` as semester, year, class time, and class location. We also add a `category` property to a syllabus to aid categorization.

The defined taxonomy will help our extraction of the list of property values (excluding the category and the policy properties) from each syllabus, and to make the collection of structured syllabi available in RDF.

Data Types

1. policy *
2. affiliation
3. category *
4. title
5. objective
6. description
7. courseWebsite

Object Types

1. assignment
2. resource
3. courseCode
4. grading
5. teachingStaff
6. specificSchedule
7. prerequisite
8. textbook
9. exam
10. generalSchedule

4 Information Extraction

Information extraction aims to extract structured knowledge, especially entity relationships, from unstructured data. In our case, we extract relations on a course such as an instance of the TEACH relation *"(Mary, Data Structure, Fall 2006)"* from a syllabus, *"(Mary teaches the Data Structure course in Fall 2006)"*. There are plenty of research studies, reviewed in [12], that applied machine learning technology to the information extraction task. These approaches can be broadly divided into rule-based approaches such as Decision Tree, and statistics-based approaches such as Hidden Markov Model (HMM). The extraction task usually involves five major subtasks: segmentation, classification, association, normalization, and deduplication [12]. For our extractor, the segmentation task includes mainly two steps – name entity recognition and topical segmentation – while the deduplication task is integrated into the classification task.

[8] http://doc.cs.vt.edu/ontologies

[9] http://protege.stanford.edu/

Thompson *et al.* [13] have tried completing these tasks with an HMM approach on course syllabi for five properties: course code, title, instructor, date, and readings. They manually identified the five properties on 219 syllabi to train the HMM. However, it would take us much more effort to label 16 properties for a collection of unstructured syllabi. Therefore, we needed a method that is unsupervised, i.e., not requiring training data. In the following subsections, we explain our choice in detail.

In addition, the association task in [12], which determines which extracted information belongs to the same record given a web page with a list of courses, is not required in our extractor, since we aim to extract information given a syllabus. The normalization task, which forms extracted information in a standard format such as presenting *"3:00pm-4:00pm"* and *"15:00-16:00"* uniformly as *"15:00-16:00"* for the class time, will be performed in the future since it does not affect extraction accuracy.

4.1 Name Entity Recognition

Name Entity Recognition (NER), a sub-task of information extraction, can recognize entities such as persons, dates, locations, and organizations. An NER F-Measure of around 90%, (a combination of the precision and the recall of recognition), has been achieved since the 7th Message Understanding Conference, MUC [10], in 1998. We therefore chose to make our name entity recognizer based on ANNIE[11] of the GATE natural language processing tool [14], which has been successfully applied to many information extraction tasks such as in [9] and is easily embedded in other applications. Table 1 shows the name entities with examples extracted through the recognizer. Our recognizer can also recognize course codes by matching them to the pattern of two to five letters followed by zero or more spaces and then two to five digits.

Entity	**Property**	**Example**
People	name of teaching staff	Dr. Mary Peters
Date	semester class time, office hour schedule date	Fall 2006 Tue/Thurs, 1:00pm -2:00pm Nov 01
Organization	university department	University of A Department of CS
Location	mailing list, contact email, home page, course web site	mary@a.edu www.a.edu/cs2604

Table 1. Name entities extracted from a syllabus.

[10] http://www-nlpir.nist.gov/related_projects/muc/

[11] http://www.aktors.org/technologies/annie/

4.2 Topical Segmentation

A course syllabus might describe many different aspects of the course such as topics to be covered, grading policy, and readings. Because such information is usually expressed in sentences, NER is not applicable for such an extraction task. In order to extract such information, it is essential to find the boundary of a topic change and then to classify the content between identified boundaries into one of the syllabus data/object types. The first half falls in the topical segmentation task and the other half will be described in the next section. Much research work has already been done on topical segmentation. We chose C99 [15] because it does not require training data and has performance comparable to the supervised learning approach which requires training data [16]. C99 measures lexical cohesion to divide a document into pieces of topics. It requires a pre-defined list of preliminary blocks of a document. Each sentence in a document is usually regarded as a preliminary block. It calculates cosine similarity between the blocks by stemming and removing stop words from each block. After the contrast enhancement of the similarity matrix, it partitions the matrix successively into segments.

C99 is not good, however, at identifying a short topic, which will be put into its neighboring segment. Therefore, we do not expect the segmenter to locate a segment with only a single syllabus property, but expect it not to split a syllabus property value into different segments. It is also critical to define a correct preliminary block which is the building block of a topical segment of C99. We defined a preliminary block at the sentence or the heading level. A heading is a sequence of words just before a syllabus property. It is usually short, and occupies a line or separates with contents of the heading by the delimiter ':'. We first located possible headings and sentences. If two headings were found next to each other, the first one was treated as a preliminary block; otherwise a heading and the following sentence form a preliminary block in case they are partitioned into different segments.

4.3 Classification

Given the topical segments and name entities of a syllabus, the final step is to associate them with the list of interesting syllabus properties through classification. The algorithm for this final step is shown in Figure 3 and the details are explained below.

First of all, (lines 1–9 in Figure 3) identify a course code, a semester, and a course affiliation (university and department) at the top of a syllabus, i.e., in the first segment. A course title is a heading and follows a course code. Second, (lines 11–18) indicate information about teaching staff by a heading with keywords such as 'instructor', 'lecturer' and more in Table 2. It might include their names, email addresses, Website URLs, phone numbers, and office hours. They should fall in the same segment. Third, (lines 19–20) identifies a course Web site by looking for the course code inside. Finally, (lines 21–25) looks for other syllabus properties: each starts with the heading of the property and falls into a single topical segment. A heading is identified based on a list of keywords, as shown in Table 2. For example, a course description heading might contain 'description', 'overview', 'abstract', 'summary', 'catalog', and 'about the course'.

```
Input: a people list (P), a date list (D), an organization list (O), a location list (L),
a course code list (C), a segment list and a property pattern list (PP).
Output: a list of property names and extracted values, E.

Begin
     For the first segment
          If a code c in C falls into this segment
          Then E← ('courseCode',c)
              If the words following the code is a heading
              Then E←('title', the words)
          If an organization o in O falls into this segment
          Then E←('courseAffiliation', o)
          If an semester item d in D falls into this segment
          Then E←('generalSchedule', d)
     For each segment
          If no entry of staff information is obtained
          Then If a person p in P falls in this segment
                Then If the teachingStaff pattern occurs before the occurrence of this
                person
                    Then E←('teachingStaff', ts)where Start_Pos(ts) =
                    Start_Pos(p)
                         If there are more items in D and L falling
                              in this segment
                         Then
                              End_Pos(ts) = max(End_Pos(these items))
                    Else
                              End_Pos(ts) = End_Pos(the segment)
     If a URL in L falling in this segment contains the course code extracted already
    Then E←('courseWebsite', the URL)
    If the segment starts with a heading
    Then for each pattern pp in PP
          If pp occurs in the heading
           Then E←(pn, the segment without the heading) where pn is the property name for
                    the pattern pp.
                 Extraction is completed for this segment.
End
```

Fig. 3. Classification algorithm to associate topical segments and name entities with syllabus properties.

Property	**Regex**
description	description|overview|abstract|summary|catalog|about the course
objective	objective|goal|rationale|purpose
assignment	assignment|homework|project
textbook	text|book|manual
prerequisite	prerequi
grading	grading
specificSchedule	lecture|topic|reading|schedule|content|outline
teachingStaff	instructor|lecturer|teacher|professor|head|coordinator|teaching assistant|grader
exam	exam|test
schedule	reference|reading|material|lecture[$'$]

Table 2. Heading Patterns for Syllabus Properties.

4.4 Evaluation

To evaluate the accuracy of the information extraction and conversion process, we randomly selected 60 out of over 700 syllabi manually identified from our potential syllabus collection [3], all in HTML format. The free text of each syllabus document (obtained by removing all HTML tags), was fed into our extractor.

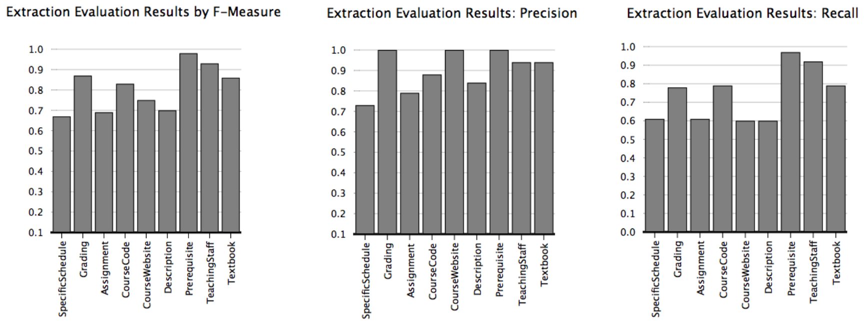
One of the co-authors, an expert in the syllabus genre, judged the correctness of extraction manually by the following procedure: our judgment criterion was that a piece of information for a syllabus property is considered extracted correctly if it is identified at the correct starting position in the syllabus as obtained via manual inspection. It was considered acceptable to include extra information that did not affect the understanding of this piece of information. For example, we judged a course title that also contained semester information, as a positive extraction.

We calculated the F-measure [13] on each property of interest, over the syllabi with this property. The F-Measure is a widely accepted evaluation metric on information extraction tasks. It is a combination of precision and recall, expressed as $F = 2 * Precision * Recall/(Precision + Recall$. Precision on a property is the ratio of the number of syllabi with the property correctly extracted over the total number of syllabi with the property extracted. Recall on a property is the ratio of the number of syllabi with this property correctly extracted over the total number of syllabi with this property. The higher the F value, the better the extraction performance.

Our extractor is more effective on some properties than others. The performance on the more effective properties is shown in Figure 4. For example, we achieved high accuracy on the `prerequisite` property at the F value of 0.98 since this field usually starts with the heading keyword *'prerequisite'* and contains a course code. On the other hand, by examining the false extractions on the properties with low accuracy, we summarize our findings as follows.

- The heuristic rule to identify a course title, (finding the heading next to a course code), is too specific to obtain high accuracy. Among the 60 syllabi we inspected, many have course titles and course codes separated by semester information.
- The extraction accuracy of a course title is also affected by that of a course code. Quite a few course codes do not match the pattern we defined. There is a larger variety of formats than we thought. For example, some course codes consist entirely of digits separated by a dot (such as '6136.123'), while some consist of two department abbreviations separated by a slash for two codes of the same course ('such as CS/STAT 5984').
- The `resource` property is identified with high precision at 0.8, but low recall at 0.42, because it is misclassified as other properties such as `textbook`. For example, many readings are present under the textbook section without an additional heading. In addition, some resources such as required software for the course are hard to identify simply from the heading. The same reason causes the `schedule`, `objective`, and `courseAffiliation` properties extracted with very high precision but low recall.
- The accuracy on the `exam` property is low in terms of recall and precision, both at the F value of nearly 0.5. It is mis-classified into `grading` sometimes, which

leads to low recall. On the other hand, the low precision is because the exam time which belongs to the `specificSchedule` property is mis-classified into an `exam` property.



(a) (b) (c)

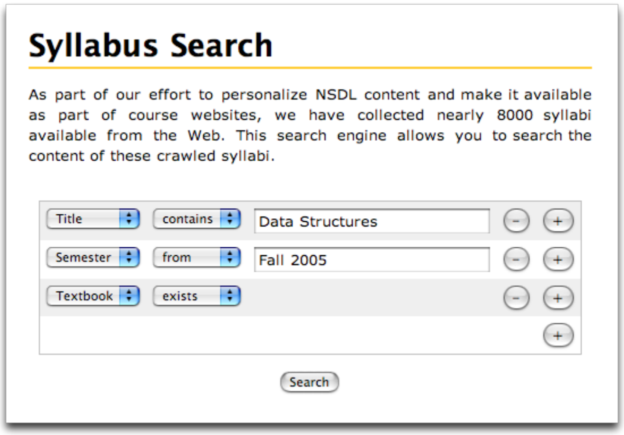
Fig. 4. Extraction evaluation results by F-Measure.

The evaluation results discussed above indicate challenges in the syllabus extraction task. First, there are many properties in a syllabus with varied presentations in varied syllabi. Trying to extract all of them at once will reduce the probability of obtaining high quality metadata on any of them. Therefore, we found it better to prioritize the few most important properties first and extract the rest later. Second, many properties' values contain long content, so the heading approach can only help in finding the starting position, but not the ending position: the `schedule` property is the best example of this observation. We should use HTML tags to ascertain structure of the HTML document. For example, schedules usually are included in an HTML table; we expect that if these tags are available during processing, the complete schedules can be extracted with high accuracy. This will also help extraction of information like textbooks, which are commonly presented in an HTML list. Creating an exhaustive list of patterns for all properties is a tedious and error-prone process. Thus, we started off with a smaller subset of patterns and properties.

5 Searching Syllabi

The availability of syllabi in a standard format with the appropriate metadata extracted from them makes several beneficial applications and services possible. We present one of these services, Semantic Search over syllabi, in detail below. Others are discussed in [17].

We have provided a semantic search service over our structured syllabus repository. This is different from other general-purpose keyword search engines in that our search



Syllabus Search

As part of our effort to personalize NSDL content and make it available as part of course websites, we have collected nearly 8000 syllabi available from the Web. This search engine allows you to search the content of these crawled syllabi.

Title contains Data Structures

Semester from Fall 2005

Textbook exists

Fig. 5. Syllabus search engine interface: advanced search dialog.

engine indexes a set of documents known with confidence to be syllabi, and provides extracted metadata to assist the user in various tasks.

For example, as shown in Figure 5, an instructor may query, from our advanced search dialog box, popular textbooks used in Data Structures courses since Fall 2005. The search results will highlight these keywords and also textbooks, plus a link to the original unstructured syllabus, and a link to the parsed syllabus in RDF format.

Our implementation is developed upon Lucene[12], a search engine development package. We index extracted metadata fields for each syllabus, and support basic search and advanced search functionalities. When a user types queries without specifying particular fields, our service searches all the indexed fields for desired syllabus. When the user specifies some constraints with the query through our advanced search dialog box, we only search in specific fields, which can find more accurate syllabi. For example, only a syllabus with textbooks will be returned for the case shown in Figure 5.

Our semantic search service would also benefit agent-based systems and other semantic web applications. For example, an application is to list popular books in a variety of courses especially in computer science. It will obtain different lists of syllabi in RDF format by the same query as the instructors's but with different course titles and then for each list rank the textbooks by their occurrences in the list.

6 Conclusions

In this paper, we proposed an intelligent approach to automatically annotate freely-available syllabi from the Internet to benefit the education community by supporting services such as semantic search. We discussed our detailed process of how to automatically convert unstructured syllabi to structured data through effective information recognition, segmentation, and classification. Our work indicates that an unsupervised

[12] http://lucene.apache.org/

machine learning approach can lead to generally good metadata extraction results on syllabi, which are hard to label manually for a training data set. The challenges of extraction on syllabus genre and suggestions for the refinement are also discussed. We hope that the experience of our approach in building genre-specific structured repositories will encourage similar contributions in other genres, eventually leading to the creation of a true Semantic Web.

7 Acknowledgments

This work was funded by the National Science Foundation under DUE grant #0532825.

References

1. Yu, X., Tungare, M., Cameron, W., Fan, W., Teng, G., Pérez-Quiñones, M., Fox, E.A., Cassel, L.: Syllabus library: Experiences in building a genre-specific structured repository. In: (To appear in) Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL 2007. (2007)
2. Hodgins, W., Duval, E.: Draft standard for learning technology - Learning Object Metadata - ISO/IEC 11404. Technical report (2002)
3. Tungare, M., Yu, X., Cameron, W., Teng, G., Pérez-Quiñones, M., Fox, E., Fan, W., Cassel, L.: Towards a syllabus repository for computer science courses. In: Proceedings of the 38th Technical Symposium on Computer Science Education (SIGCSE 2007). (2007)
4. Cassel, L., Hacquebard, A., Mcgettrick, A., Davies, G., Leblanc, R., Riedesel, C., Varol, Y., Finley, G., Mann, S., Sloan, R.: Iticse 2005 working group reports: A synthesis of computing concepts. ACM SIGCSE Bulletin **37**(4) (December 2005)
5. das Neves, F.: Stepping Stones and Pathways: Improving Retrieval by Chains of Relationships between Documents. PhD thesis, Virginia Tech Department of Computer Science (Sep 2004)
6. Pomerantz, J., Oh, S., Yang, S., Fox, E.A., Wildemuth, B.M.: The core: Digital library education in library and information science programs. D-Lib Magazine **12**(11) (November 2006)
7. de Larios-Heiman, L., Cracraft, C.: SylViA: The Syllabus Viewer Application. http://groups.sims.berkeley.edu/sylvia/ (Last Accessed: March 2006)
8. Dolog, P., Henze, N., Nejdl, W.: Reasoning and ontologies for personalized e-learning. Educational Technology and Society (2004)
9. Dowman, M., Tablan, V., Cunningham, H., Popov, B.: Web-assisted annotation, semantic indexing and search of television and radio news. In: Proceedings of the 14th international conference on World Wide Web (WWW '05), New York, NY, USA, ACM Press (2005) 225–234
10. Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.A.: Automatic document metadata extraction using support vector machines. In: JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital Libraries, Washington, DC, USA, IEEE Computer Society (2003) 37–48
11. Takasu, A.: Bibliographic attribute extraction from erroneous references based on a statistical model. In: JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital Libraries, Washington, DC, USA, IEEE Computer Society (2003) 49–60
12. Mccallum, A.: Information extraction: Distilling structured data from unstructured text. ACM Queue **3**(9) (November 2005)

13. Thompson, C.A., Smarr, J., Nguyen, H., Manning, C.: Finding educational resources on the web: Exploiting automatic extraction of metadata. In: Proc. ECML Workshop on Adaptive Text Extraction and Mining. (2003)
14. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust nlp tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia (July 2002)
15. Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proceedings of the first conference on North American chapter of the Association for Computational Linguistics, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2000) 26–33
16. Kehagias, A., Nicolaou, A., Petridis, V., Fragkou, P.: Text segmentation by product partition models and dynamic programming. Mathematical and Computer **39**(2-3) (January 2004) 209–217
17. Tungare, M., Yu, X., Teng, G., Pérez-Quiñones, M., Fox, E., Fan, W., Cassel, L.: Towards a standardized representation of syllabi to facilitate sharing and personalization of digital library content. In: Proceedings of the 4th International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL). (2006)