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], and Lillian Cassel[2]

[1] Virginia Tech, Blacksburg VA 24061, USA
{xiaoyany,manas,wfan,perez,fox}@vt.edu
http://syllabus.cs.vt.edu/
[2] Villanova University, Villanova PA 19085, USA
{william.cameron,lillian.cassel}@villanova.edu

Abstract. Syllabi are important documents created by instructors for students. 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 for syllabus construction. In this paper, we propose an intelligent approach to automatically annotate freely-available syllabi from the Web to benefit the educational community through supporting services such as semantic search. We discuss our detailed process for converting unstructured syllabi to structured representations through entity recognition, segmentation, and association. Our evaluation results demonstrate the effectiveness of our extractor and also suggest improvements. We hope our work will benefit not only users of our services but also 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 adapting information from other relevant syllabi. 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 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.

Supporting activities like those mentioned above can be facilitated if metadata is extracted from syllabi. However, two obstacles hinder this, especially with respect to the syllabus genre. First, no metadata standard is specific to the syllabus genre, although markup schemes, such as IEEE LOM [1], exist for educational resources. Thus, while we are able to annotate a document as a syllabus by the LOM's resource type property, we are unable to annotate a piece of information inside a syllabus as a textbook using any of the available metadata standards. Second, it requires too much effort to manually annotate information inside syllabi, and no approach is available to automate the process of information extraction from the syllabus genre. Motivated by these two observations,

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 337–346, 2007.

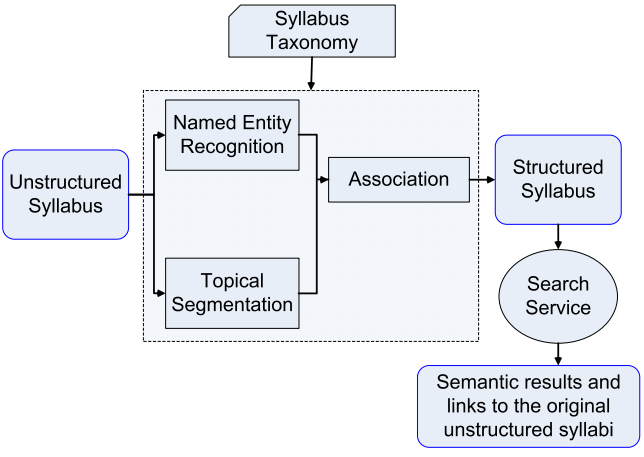

Fig. 1. Workflow from a unstructured syllabus to a structured syllabus

we propose a taxonomy and an extraction approach specific to the syllabus genre, build a structured syllabus digital library (DL) by extracting metadata from each syllabus, and support semantic search of the syllabi through the DL (and thus through the Semantic Web).

Figure 1 shows the flow of the transformation from an unstructured syllabus to a structured syllabus and then the retrieval of structured syllabi. Following our syllabus taxonomy (Section 2), semantic information can be extracted from a syllabus, which becomes part of the Semantic Web. The named 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. Finally, the association module associates a list of syllabus properties with the segmented values, and stores them in the structured syllabus repository. These three modules work together for the information extraction task (Section 3). The search service (Section 4) indexes structured syllabi and provides semantic search results through both RDF[1] and links to the raw syllabi.

There are many other types of unstructured data on the Web; thus, success with our genre-specific structured repository suggests that there are opportunities to use such other data in similar innovative applications. We hope that our application of machine learning techniques to extract and obtain structured genre-specific data will encourage the creation of other similar systems.

2 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 variations

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

make it difficult to reuse information from these syllabi. It also is 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[2] and show the first level of the taxonomy in Table 1. Among these 15 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, while a `prerequisite` is composed of one or more `courseCode` objects. It also is 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.

The taxonomy will help both our extraction of the list of property values from each syllabus, and our making the collection of structured syllabi available in RDF.

Table 1. First level of syllabus taxonomy

Data Type	affiliation, title, objective, description, courseWebsite
Object Type	assignment, resource, courseCode, teachingStaff, grading, specificSchedule, prerequisite, textbook, exam, generalSchedule

3 Information Extraction

Information extraction aims to extract structured knowledge, including entity relationships, from unstructured data. In our case, for example, we would extract relations 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 [2], that have 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 four major subtasks: segmentation, association, normalization, and deduplication [2]. For our extractor, the segmentation task includes mainly two steps – named entity recognition and topical segmentation – while the deduplication task is integrated into the association task. In addition, the normalization task, which puts extracted information into 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.

Thompson *et al.* [3] 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

[2] http://syllabus.cs.vt.edu/ontologies

would take us much more effort to label 15 properties for a large 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 approach in detail.

3.1 Named Entity Recognition

Named Entity Recognition (NER), a sub-task of information extraction, can recognize entities such as persons, dates, locations, and organizations. An NER F_1 (a combination of the precision and the recall of recognition) of around 90% commonly has been achieved since the 7^{th} Message Understanding Conference, MUC [3], in 1998. We therefore chose to base our named entity recognizer on a proven routine, ANNIE[4], part of the GATE natural language processing tool [4]. It has been successfully applied to many information extraction tasks such as in [5] and is easily embedded in other applications. Our recognizer also can 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.

3.2 Topical Segmentation

A course syllabus might describe many different aspects of the course such as topics to be covered, grading policies, and readings. Because such information is usually expressed in arbitrary sentences, NER is not applicable for that part of the extraction task. In order to extract such information, it is essential to find the boundaries indicating 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 [6] because it does not require training data and has performance comparable to the supervised learning approach which requires training data [7]. 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. C99 calculates the 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 also is 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 often occupies a line. At other times the heading and its contents are separated 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.

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

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

```
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
1 For the first segment
2    If a code c in C falls into this segment
3    Then E← ('courseCode',c)
4            If the words following the code is a heading
5            Then E←('title', the words)
6       If an organization o in O falls into this segment
7    Then E←('courseAffiliation', o)
8    If an semester item d in D falls into this segment
9    Then E←('generalSchedule', d)
10 For each segment
11   If no entry of staff information is obtained
12   Then if a person p in P falls in this segment
13        Then if the teachingStaff pattern occurs before the occurrence of
               this person
14             Then E←('teachingStaff', ts)where Start_Pos(ts) = Start_Pos(p)
15                  If there are more items in D and L falling in this segment
16                  Then End_Pos(ts) = max(End_Pos(these items))
17                  Else
18                       End_Pos(ts) = End_Pos(the segment)
19   If a URL in L falling in this segment contains the course code extracted
     already
20   Then E←('courseWebsite', the URL)
21   If the segment starts with a heading
22   Then for each pattern pp in PP
23             If pp occurs in the heading
24             Then E←(pn, the segment without the heading) where pn is the
                    property name for the pattern pp.
25                  Extraction is completed for this segment.
End
```

Fig. 2. The algorithm to associate topical segments and named entities with syllabus properties

Table 2. Heading Patterns for Syllabus Properties

Property	Regular Expression (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[∧r]

3.3 Association

Given the topical segments and named entities of a syllabus, the final step is to associate them with the list of interesting syllabus properties. The algorithm for this final step is shown in Figure 2 and the details are explained below.

First of all, lines 1–9 in Figure 2 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 identify a course Web site by looking for the course code inside. Finally, lines 21–25 look 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', or 'about the course'.

3.4 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 [8], 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_1 on each property of interest, over the syllabi with this property. The F_1 is a widely accepted evaluation metric on information extraction tasks. It is a combination of precision and recall, expressed as $F_1 = 2 * Precision * Recall / (Precision + Recall)$. 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_1 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 3. For example, we achieved high accu-

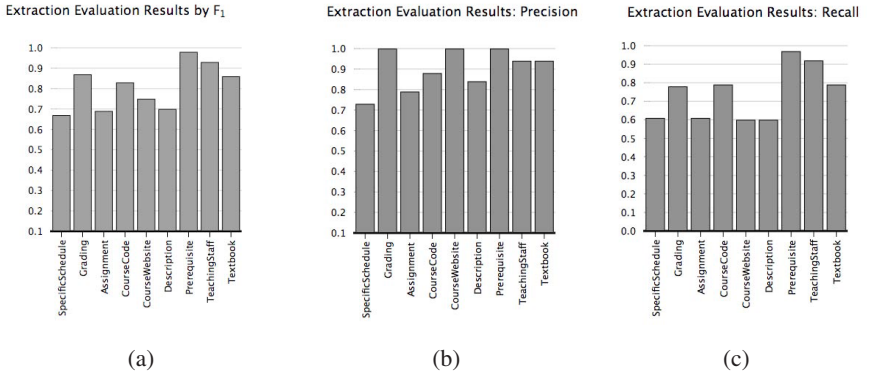

Fig. 3. Extraction evaluation results by F_1

racy on the `prerequisite` property at the F_1 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, namely 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 to be 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_1 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.

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, not the ending position: the `schedule` property is the best example of this observation. We should use HTML tags to ascertain the structure of HTML documents. 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 also will help extraction of information like textbooks, which are commonly presented in an HTML list. Creating an exhaustive set of patterns for all properties is a tedious and error-prone process. Thus, we started off with a smaller subset of patterns and properties.

4 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

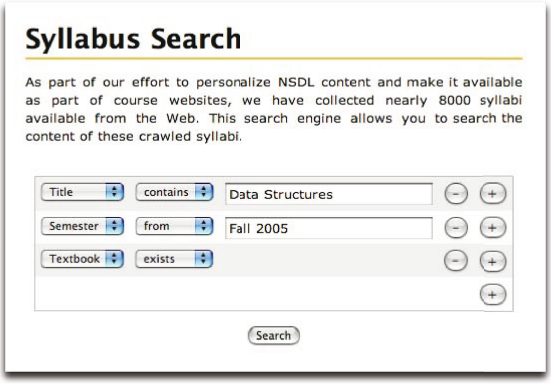

Fig. 4. Syllabus search engine interface: advanced search dialog

of these services, Semantic Search over syllabi, in detail below. Some others are discussed in [9].

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 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 4, 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 indicative keywords and also identified textbooks; there also will be a link to the original unstructured syllabus, and a link to the parsed syllabus in RDF format.

Our implementation is developed upon Lucene[5], 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 syllabi. When the user specifies some constraints with the query through our advanced search dialog box, we only search in specific fields, which can find syllabi with greater accuracy. For example, only a syllabus with textbooks will be returned for the case shown in Figure 4.

Our semantic search service also would 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.

5 Related Work

There are a few ongoing research studies on collecting and making use of syllabi. The MIT OpenCourseWare project manually collects and publishes 1,400 MIT course

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

syllabi in a uniform structure 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. Our previous work [10] helps with automating the syllabus acquisition process by identifying true syllabi from search results on the Web.

Some have addressed the problem of lack of standardization of syllabi. Along with a defined syllabus schema, SylViA [11] 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, such as the ontology of a learner, especially in a remote learning environment [12]. 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 educational resources to 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 genre (such as television and radio news [5]). Such systems vary in keeping with the different genre, due to their own characteristics and service objectives. To our knowledge, there is no specific annotation and semantic search system for the broad syllabus genre.

Much work has been done on metadata extraction from other genre such as academic papers. For example, Han *et al.* [13] described using Support Vector Machines for metadata extraction from a paper's header field.

6 Conclusions

In this paper, we proposed an intelligent approach to automatically annotate freely-available syllabi from the Internet and to benefit the education community through supporting services such as semantic search. We discussed our detailed process to automatically convert unstructured syllabi to structured data. Our work indicates that an unsupervised 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 the syllabus genre, along with suggestions for refinement, are discussed. We hope that the experience of our approach in building genre-specific structured repositories will encourage similar contributions for other genre, eventually leading to the creation of a true Semantic Web.

Acknowledgments

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

References

1. Hodgins, W., Duval, E.: Draft standard for learning technology - Learning Object Metadata - ISO/IEC 11404. Technical report (2002)
2. Mccallum, A.: Information extraction: Distilling structured data from unstructured text. ACM Queue 3(9) (November 2005)

3. 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)
4. 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 2002), Philadelphia (July 2002)
5. Dowman, M., Tablan, V., Cunningham, H., Popov, B.: Web-assisted annotation, semantic indexing and search of television and radio news. In: WWW 2005. Proceedings of the 14th international conference on World Wide Web, pp. 225–234. ACM Press, New York (2005)
6. 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, pp. 26–33. Morgan Kaufmann, San Francisco (2000)
7. Kehagias, A., Nicolaou, A., Petridis, V., Fragkou, P.: Text segmentation by product partition models and dynamic programming. Mathematical and Computer 39(2-3), 209–217 (2004)
8. 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: SIGCSE 2007. Proceedings of the 38th Technical Symposium on Computer Science Education, vol. 39, pp. 55–59. ACM Press, New York, NY, USA (2007)
9. Tungare, M., Yu, X., Teng, G., P érez 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)
10. Yu, X., Tungare, M., Fan, W., Pérez-Quiñones, M., Fox, E.A., Cameron, W., Teng, G., Cassel, L.: Automatic syllabus classification. In: Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL 2007, pp. 440–441 (2007)
11. de Larios-Heiman, L., Cracraft, C.: (SylViA: The Syllabus Viewer Application)
12. Dolog: Reasoning and ontologies for personalized e-learning. Educational Technology and Society (2004)
13. Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.A.: Automatic document metadata extraction using support vector machines. In: JCDL 2003. Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital Libraries, Washington, DC, USA, pp. 37–48. IEEE Computer Society Press, Los Alamitos (2003)