Finding Educational Resources on the Web: Exploiting Automatic Extraction of Metadata

Cynthia Thompson
School of Computing,
University of Utah
Salt Lake City, UT 84112

Joseph Smarr and **Huy Nguyen** and
Christopher D. Manning
Dept of Computer Science, Stanford University,
Stanford CA 94305-9040

Abstract

Searching for educational resources on the web would be greatly facilitated by the availability of classificatory metadata, but most web educational resources put up by faculty members provide no such metadata. We explore using text classification and information extraction techniques to automatically gather such metadata. Text classifications orthogonal to topic matter appear possible with high ($> 90\%$) accuracy, and exact-match information extraction has an F measure around 50%.

1 Introduction

In order to be able to do semantically rich queries over distributed heterogeneous data collections like the web, a key tool is the use of metadata to explicitly annotate documents with relevant information. This is the general goal of the semantic web [1], and such markup schemes exist for education resources, for example, IEEE LOM [2]. In the particular context of the Edutella project [3], learning resources such as lesson plans, tutorials, assignments, and so on are annotated with the educational topic to which they pertain, the education level of the intended audience, and so on. The presence of such attributes allows highly customized searches, as well as quick summaries of available documents.

A major challenge to building a metadata-rich repository is that someone has to manually annotate all the documents. This is a slow and costly process, and many producers of educational content are probably not interested in going back and annotating all their work. In the Edutella context, semantic metadata is available for documents within the Edutella peer-to-peer network, but it would be useful to be able to conveniently access the mass of other educational resources available on the web: there are numerous valuable educational resources available, but finding them using traditional keyword searches is hard, and most of them are not annotated with any useful metadata.

While searches can use available metadata when present, there is thus a clear need to develop tools that can perform some or all of this annotation automatically. Such tools would save content creators time and would allow content consumers to utilize the web as if it were part of the same metadata rich environment to which they were accustomed. In cases where fully automatic annotation cannot be accomplished with sufficient accuracy, there is still value in providing suggestions to human annotators. Or, one could highlight information that is relevant to the annotation decision, such as word features that a classifier has found to be relevant in text classification discrimination.

There are two major technical avenues for automatic metadata extraction. First is the classification of documents into appropriate categories on various dimensions (e.g., what language the document is written in, what type of learning resource it is, or what level of student it is intended for). Second is the extraction of text from documents for summary fields (e.g., title and author of the document, topics covered in a course description, or readings assigned on a syllabus). In both cases, it is reasonable to seek systems that work from limited training data: for any fine-grained topic there is only a limited amount of material on the web, and if people have to find and annotate most of those pages, then there is little to be gained. So methods that can quickly generalize are of particular interest. When dealing with web pages, another interesting question is whether HTML markup can be usefully exploited in addition to the text content for classification or extraction.

In this paper, we present some early results on this task of providing metadata for educational web pages, considering first text classification, and then information extraction.

2 Data Collection and Annotation

We downloaded text web pages for both classification and information extraction. While it would be useful to extend analysis to other formats such as PDF and postscript, the present experiments used just HTML pages. For classification, we collected 4 different types of resources: syllabi, assignments, tutorials, and exams. Such a text classification task is orthogonal to typical topic-based classification decisions, but it seems reasonable that good results should still be possible based on features present in the documents.

When collecting pages, we restricted ourselves to artificial intelligence and machine learning courses, since the syllabi in this case were also used for infor-

Table 1: Reachable Relevant Pages out of Top 20

Search Term	Num Returned	Num Relevant
syllabus	20.3K	15
class	442K	10
course	550K	7
introduction	552K	7

mation extraction (see below). For the most part, we found these pages by a web search engine using the terms "artificial intelligence" or "machine learning" with "course," "class," or "syllabus." We also used some directories and lists of such courses of which we were aware. We then proceeded from the main course page to find connected pages with assignments or online exams. For the tutorials, we used similar search terms and directories.

As noted previously, finding these resources via keyword search alone is difficult. We measured informally the difficulty of the task. We combined the search term "artificial intelligence" with each of "course," "class," "syllabus," and "introduction," and measured the number of useful pages that were easily reachable from the pages of the top 20 search terms returned. Table 1 summarizes the results. Even these figures are fairly lenient, as we included in the count both duplicates between the four terms and pages that we could find by following one or two links from the page returned by the search. An automated crawler would fare much worse in distinguishing the desirable pages.

Finally, for the extraction task, we took the syllabi collected above and used an annotation tool to tag them with a set of 5 tags, course number, course title, instructor, year, and readings. This tool also removed certain HTML material, such as SCRIPT blocks and comments; these features were also removed from the pages before classification was applied.

3 Classification

For the classification task, we used a classifier to distinguish between different resource types. Classification was done with a maximum entropy classifier, which here used just word-category features, and hence is simply a multiclass logistic regression model. The model uses a Gaussian prior on feature weights for regularization, and is fit by conjugate gradient search. The model is essentially similar to [4], and actually uses the classifier within the sequence model described in [5].

We collected 385 pages in all: 131 assignment pages, 219 syllabus pages, 22 tutorial pages, and 13 exam pages. Given this data distribution, we are close to dealing with a two-class classification problem, so we looked at accuracy both for the two class case and for the more unbalanced data set. So for both cases, for five random splits of the data, we trained the classifier on 80% of the pages and tested on the remaining 20%. The training set accuracy on these splits was uniformly 100% accurate. We created learning curves for both cases, training on increasingly larger portions of the training set, and testing classification accuracy on the test set. The trends are shown in Figure 1. For the full training set in the multi-class case, the average test set accuracy was 92.5% and the number of word features was about 22,160 on average. For the binary case, the average accuracy was 98.6% and the number of word features was 20,235.

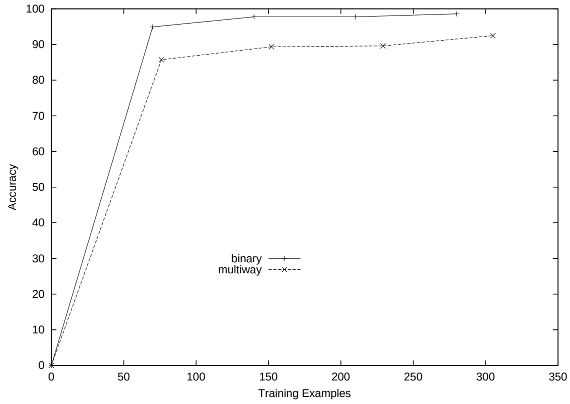


Figure 1: Classification Accuracy for Resource Type

Table 2: Multi-way accuracy per label

Category	Prec	Rec	F_1
assignment	89.6	91.6	91
syllabus	95.3	100	98
exam	50	27.5	35
tutorial	68.75	51.78	59

For the binary case, errors are fairly evenly split between mistaking assignment for syllabus and vice versa, with a slight tendency in some splits to mislabel assignment as syllabus more than the reverse mistake, but both types of mistakes were made in at least one training set size for every split. For the multi-class case, the results were more mixed. The precision, recall, and F_1 are shown in Table 2. For exam, all errors were due to their incorrect labeling as assignment. This is not surprising given the similar nature of the two. For tutorial, they either got mislabeled as assignment or syllabus, almost equally, but never as exam. We examined the weights learned in two of the splits: for assignment, the maximum weighted feature in both splits was *forbidden*, coming from statements prohibiting copying from other students or online sources!

4 Information Extraction

For many types of metadata, including course titles and instructors, the possible values for fields are not confined to a closed set, and are therefore beyond the extraction capabilities of classification. Information extraction is a promising alternative, since it allows us to accommodate variation in the values of the field, as well as exploit the surrounding context for extraction. Our general approach to information extraction is the use of class HMMs, in the general tradition of [6]. In this model, the mathematics of which is clearly described in [7], each hidden state generates not only a word, but also a class label: the name of a field to be extracted or 'Background'. At training time, the state sequence is partially constrained by observed class labels, but not full determined, and parameter estimation is done by the EM algorithm. For unknown words, the model uses a class-based model, based on features of words, such as capitalization and the presence of numbers. Figure 2 indicates one kind of HMM topology used in the experiments. This structure seeks to extract a single target field, uses a unigram model for the background, and attempts to model the target prefix and suffix with three states. Not shown are self-loops on every state, and forward arcs on the target states. We experimented some with target chains of different lengths based on the field being extracted, but since skipping is allowed in target chains, a longer chain structure can still model targets of shorter lengths, and this parameter did not have much effect.

We also experimented with a single large HMM which contained states corresponding to all of the target fields. The beginning and end of these target chains were fully connected to each other and eight background states (an ergodic context model), with parameter estimation used to find a suitable model structure.

Table 3 summarizes our experimental results. As discussed above, the data set was 219 syllabi coded for 5 fields, and we present the average of 10-fold cross-validation experiments (each test fold is quite small, and there is quite high variance in the results between folds). We used the same evaluation metric as [6] – reporting the F_1 score (harmonic mean) of precision and recall, based on exact target matching, calculated on the basis of correctly instantiating the fields of a metadata relation for the page. This is a fairly stringent evaluation criterion (for instance, a mostly correct evaluation missing a word gets no credit).

Results are in general promising, but some fields are easier than others. Incorporating basic HTML markup tags boosted performance. This is not surprising, as certain HTML tags, such as `<title>` are strongly correlated with target fields, and targets frequently occur within `<i>` and `<b>` tags, or following `<li>` tags. The two chain lengths are for the length of the target chain, and of each of the prefix and suffix chains. Initially target lengths were chosen heuristically based on the complexity of the fields, but in retrospect simply choosing a uniform chain length of 4 would have made no real difference (given that skipping forward is allowed in target chains). The context chain length partly determines how much context can be modeled (but note that the context states have self-loops, unlike those of [6] – something that we have found useful in other experiments). Here, the data set is sparse enough that having more than one context state on each side of the target does not seem to have any useful discriminating power. We also recognized that certain fields, such as course number and course title, tend to appear near the top of the page. We attempted to exploit this (non-stationary) domain knowledge by additionally trying extracting the first segment labeled as a target, whereas the standard system returns the "best" segment (the one with the highest length-normalized generation probability within a window). This met with mixed success: it was very helpful for the course-number field, but didn't have positive value for the course-title field. Ways of choosing between targets picked out by the HMM deserves further thought. Finally, the all-fields-at-once HMM might be hoped to do better global modeling of the sequence of entities in a document, at the cost of having a less detailed model of the prefix and suffix contexts of individual fields. But at least for this data set, the two models seem to give roughly equal results overall.

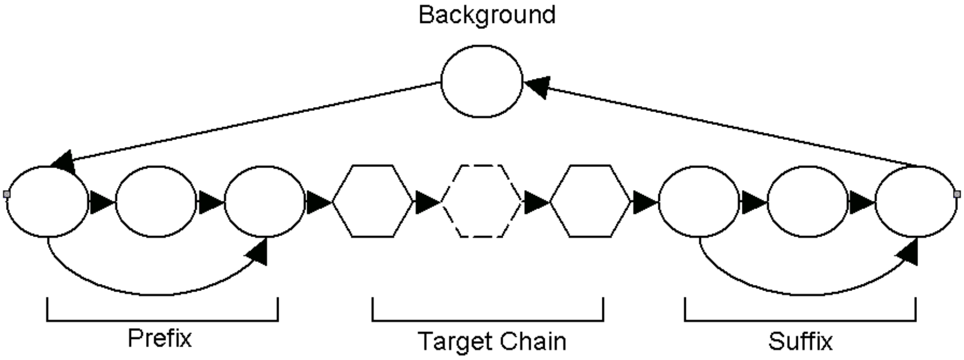
Table 3: HMM information extraction results.

Target Field	Targ/Conx Chain Len	Keep HTML?	First/ Best	F_1
One field at a time results				
Course number	4/1	Yes	First	78.3
Course number	2/3	Yes	First	68.8
Course number	4/1	Yes	Best	67.0
Course number	2/3	Yes	Best	63.5
Course number	2/3	No	Best	51.3
Course title	4/3	Yes	First	43.6
Course title	4/3	Yes	Best	52.5
Course title	4/3	No	Best	37.3
Instructor	3/3	Yes	Best	37.0
Instructor	4/1	Yes	Best	35.1
Instructor	3/3	No	Best	35.5
Date	4/1	Yes	Best	53.0
Date	4/3	Yes	Best	51.2
Reading	4/3	Yes	Best	21.0
All fields at once results				
Course number	4/NA	Yes	Best	55.5
Course title	4/NA	Yes	Best	53.2
Instructor	4/NA	Yes	Best	37.8
Date	4/NA	Yes	Best	49.7
Readings	4/NA	Yes	Best	31.1

5 Plans for Future Work

The results indicate that automatic extraction of metadata is feasible at least in certain cases, but much could be done to improve the utility of such an approach. Extracting summary information is clearly a more difficult task in general than classifying a page into one of a set of categories. Additional data would presumably help, though in many cases it would be unreasonable to expect more labeled data than was available here. One promising avenue is to exploit existing domain knowledge top-down to constrain classification and extrac-

Figure 2: Indicative HMM structure used in information extraction experiments.



tion. For example, if one knows the course numbering system at a university, that can be helpful in determining whether a course web page is intended primarily for undergraduates or graduate students. Another is to realize that word level data is likely to be too sparse for effective training in many cases, and to make more use of higher level notions, such as person names, which could be provided by a generic named entity recognizer. Eventually, the success metric of this approach is to be measured by whether it saves human annotators time reaching a given standard of annotation quality for documents, and whether it provides value in obtaining educational material from the web beyond simply using a search engine.

References

[1] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284** (2001) 35–43

[2] IEEE: Draft standard for learning technology – Learning Object Metadata – ISO/IEC 11404. Technical Report IEEE P1484.12.2/D1 (2002)

[3] Simon, B., Miklós, Z., Nejdl, W., Sintek, M., Salvachua, J.: Elena: A mediation infrastructure for educational services. In: Twelfth International World Wide Web Conference. (2003)

[4] Zhang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. Information Retrieval **4** (2001) 5–31

[5] Klein, D., Smarr, J., Nguyen, H., Manning, C.D.: Named entity recognition with character-level models. In: CoNLL 7. (2003) 180–183

[6] Freitag, D., McCallum, A.: Information extraction with HMM structures learned by stochastic optimization. In: Proceedings of AAAI. (2000) 584–589

[7] Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge (1998)