Automatic Syllabus Classification using Support Vector Machines

Xiaoyan Yu, Manas Tungare, Weiguo Fan, Yubo Yuan,
Manuel Pérez-Quiñones, Edward A. Fox
Virginia Tech
{xiaoyany, manas, wfan, ybyuan, perez, fox}@vt.edu
William Cameron, Lillian Cassel
Villanova University
{william.cameron, lillian.cassel}@villanova.edu

Abstract

Syllabi are important educational resources. Gathering syllabi that are freely available and creating useful services on top of the collection presents great value for the educational community. However, searching for a syllabus on the Web using a generic search engine is an error-prone process and often yields too many irrelevant links. In this chapter, we describe our empirical study on automatic syllabus classification using Support Vector Machines (SVM) to filter noise out from search results. We describe various steps in the classification process from training data preparation, feature selection, and classifier building using SVMs. Empirical results are provided and discussed. We hope our reported work will also benefit people who are interested in building other genre-specific repositories.

Index Terms

SVM, Syllabus, Text Classification, Feature Selection

I. 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 adding updated course information or 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. In addition, a life-long learner identifies basic topics of a course and popular textbooks by comparing syllabi from different universities. A syllabus is thus an essential component of the educational system.

Free and fast access to a collection of syllabi could have a significant impact on education. Unfortunately, searching for a syllabus on the Web using a generic search engine is an error-prone process and often yields too many irrelevant links. As an alternative, the MIT OpenCourseWare[1] project, which provides free access to MIT course materials, 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 much 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 collection of syllabi to work on. Hence, to break out of this deadlock, we decided to seed our repository with syllabi acquired from the Web in order to bootstrap the process. We restrict our focus to computer science syllabi offered by universities in the USA as a starting point of our

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

Class	Definition	Syllabus	Out-Links
Full	a syllabus without links to other syllabus components.	T	F
Partial	a syllabus along with links to other syllabus components somewhere else.	T	T
Entry Page	a page that contains a link to a syllabus.	F	T
Noise	all others.	F	N/A

TABLE I
Class Definitions.

proof-of-concept project. The methodology and the system could be extended easily to other disciplines and locations.

This paper presents our progress regarding automatic classification towards building a syllabus collection. A classification task usually can be accomplished by defining classes, selecting features, preparing a training corpus, and building a classifier. In order to build quickly an initial collection of CS syllabi, we obtained more than 8000 possible syllabus pages by automatically searching on Google [1]. After randomly examining the set, we found the result set very noisy. To help with the task of properly identifying true syllabi, we defined four syllabus class types, shown in Table I, and then proposed syllabus feature characteristics for each class. We prepared a variety of training data in terms of their sizes and their distributions. Finally, we applied Support Vector Machines (SVM) [2] to learn classifiers to produce the syllabus repository.

There are many other genres of data on the Web. We hope that our application of machine learning techniques to obtain a repository of genre-specific data will encourage the creation of similar systems for other genres.

II. Class Definition

The four classes of syllabi are defined in Table I. A syllabus component is one of the following information: course code, title, class time and location, offering institute, teaching staffs, course description, objectives, web site, prerequisite, textbook, grading policy, schedule, assignment, exam and resources. We consider only the full and the partial classes as syllabi. The reason we treat a partial syllabus as a syllabus is that we can complete a partial syllabus by following outgoing links from it, which would be helpful for a variety of services. For example, in order to recommend papers or textbooks for a course using a partial syllabus, it is inaccurate just to extract frequent words from its syllabus since more features of the course are described in other pages. Therefore, we would like to recognize partial syllabi and then retrieve more complete information from them. Similarly, we also need to differentiate between an entry page and a noise page, although we consider neither of them as syllabi.

III. Feature Selection

In a text classification task, a document is represented as a vector of features usually from a high dimensional space that consists of unique words occurring in documents. A good feature selection method reduces the feature space so that most of learning algorithms can handle and contribute to high classification accuracy. We applied three feature selection methods in our study: general feature selection,

genre-specific feature selection, and a hybrid of the two.

A. General Features

Yang *et al.* [3] conducted a comparative study on five feature selection methods generally for text categorization tasks (a task similar to text classification). They selected words as features based on document frequency (DF), information gain (IG), mutual information (MI), a χ^2-test (CHI), and term strength (TS). The DF of a word is the number of documents in which the word appears. IG measures the number of information gained for class prediction by knowing the presence or absence of a word in a document. MI calculates the association of a word and a class. They found that IG and MI performed best in their study. They also concluded that DF is a good choice since it's performance was similar to the one deemed best and it is simple in terms of time complexities. Therefore, we chose DF as our general feature selection method. In [3], the best classification performance by means of the DF method was achieved at the reduction of the feature space to 2000–4000 unique words. Hence, we selected words whose DFs are not less than 10, 20, and 30, which reduced 63963 unique words in the training corpus into the 2000–4000 range. After removing words that were too specific to the training corpus, such as URLs of university websites, we obtained 3836 features at DF=10, 2325 features at DF=20, and 1754 features at DF=30.

B. Genre Features

Each class defined in Table I has its own characteristics other than general features. An entry page would contain a link with the word *'syllabus'* or prefixed with *'syl'* or a link whose anchor text contains the *'syllabus'* keyword. Many keywords such as *'prerequisite'* occur in a full syllabus. These keywords also occur in a partial syllabus, but often along with a link. In addition, the position of a keyword within a page matters. For example, a keyword within the anchor text of a link or around the link would suggest a syllabus component outside the page. A capitalized keyword at the beginning of a page would suggest a syllabus component with a heading in the page. Motivated by the above observations, we manually selected 84 features, grouped into categories listed below with the number of features in each category within parentheses, to classify our data set into the four classes. We used both content and structure features for syllabus classification, as they have been found useful in the detection of other genres [4]. These features mainly concern the occurrences of keywords, the positions of keywords, and the co-occurrences of keywords and links.

Content (with 15 features)

- the relative number of occurrences of the syllabus keyword in the document to the total number of words in the page (1);
- the relative number of occurrences of words or phrases satisfying the patterns defined in Table II (14).

Structure (with 69 features)

- the document type: HTML, PDF, PostScript and plain text (1);
- whether the URL of the document contains the *'syllabus'* keyword (1);
- the number of links in the document (1);
- the average positions of links in terms of link-in-line and line-in-file (2);
- the relative number of links which contain the *'syllabus'* keyword to the total number of links in the page, the average positions of such links in terms of word-in-line and line-in-file (3);
- the relative number of links with which the syllabus keyword and each identified pattern respectively are in the same line and also the total number of such links (16);
- the average positions of the syllabus keyword and each identified pattern respectively in terms of word-in-line and line-in-file (30);
- the relative number of occurrences of the *'syllabus'* keyword and words or phrases satisfying the patterns defined for each property that are capitalized (15).

We also defined in Table II patterns that often occur in a syllabus. The patterns of all but the course code are stems of keywords or phrases of the properties. A course code pattern consists of uppercase letters and digits.

<table>
<tr><th>Description</th><th>Regex</th></tr>
<tr><td>syllabus keyword</td><td>syll</td></tr>
<tr><td>course description</td><td>description|overview|abstract|summary|catalog|about the course</td></tr>
<tr><td>course objective</td><td>objective|goal|rationale|purpose</td></tr>
<tr><td>assignment</td><td>assignment|homework|project</td></tr>
<tr><td>textbook</td><td>text|book|manual|ISBN</td></tr>
<tr><td>prerequisite</td><td>pre−?requi</td></tr>
<tr><td>grading</td><td>grading</td></tr>
<tr><td>policies</td><td>polic|cheating|integrity</td></tr>
<tr><td>course schedule</td><td>lecture[∧r]|topic|reading|schedule|content|outline|reference</td></tr>
<tr><td>instructor</td><td>instructor|lecturer|teacher|professor|head|coordinator|office hour</td></tr>
<tr><td>TA</td><td>teaching assistant|grader|ta</td></tr>
<tr><td>exam</td><td>exam|test</td></tr>
<tr><td>affiliation</td><td>college|department|university</td></tr>
<tr><td>semester</td><td>fall|summer|spring|winter</td></tr>
</table>

TABLE II

Common keyword patterns for the syllabus genre.

IV. Training Data Preparation

In a supervised learning procedure, it is important to prepare a labeled training set for model building. In order to build quickly an initial collection of CS syllabi, we obtained around 8000 possible syllabus pages by querying for the top 100 computer science department web sites within the `.edu` top-level domain and then querying within these department web sites for the top 100 potential syllabi with the ‘syllabus’ keyword. We randomly sampled 1020 documents (HTML, PDF, PostScript or Text). Three team members classified these into the four categories. A document was classified into a category iff all three raters unanimously agreed. We observed 499 full, 208 partial, 138 entry and 175 noise pages in the sample set. We took this sample set as our training corpus where the total number of unique words was 63963. We obtained seven representations of the corpus by means of seven feature selection methods as described in Section III and referred to each as a corpus representative. In order to measure stable classification performance, we performed 10 stratified splits of each corpus representative, and took each of them as a testing set and the remaining splits as a training set candidate. That is, we formulated 10 training and testing data pairs with the ratio of 9:1 and kept the class distribution of the whole training corpus in each data set. We also varied each training set candidate with respect to the class distribution and training sizes.

A. Class Distribution Settings

The distribution of our corpus among the four classes is not uniform. As we observed, nearly half of the documents are full syllabi. We would like to find out whether such a distribution affects classification performance. We produced a new uniform training set candidate of each stratified candidate using sampling with replacement with a bias towards uniform class distribution. We will compare these two different sampling methods and their effects on classification performance.

B. Training Size Settings

In order to investigate the effect of different training sizes on classification accuracy, we performed 10 stratified splits on each training set candidate and produced 10 training sets with increasing sizes as follows. The i^{th} training set, tr_{i-1}, consists of the first i splits where i is from one to ten. The last training set is then the entire training set candidate itself. Since we had 10 training set candidates for each distribution, we obtained 200 training sets for each corpus representative.

V. Support Vector Machines

There are various classification methods available. In previous literature [5], [6], it has been found that Naïve Bayes and Support Vector Machines (SVM) are the most commonly used and effective text classification methods in general. Our previous work [7] in syllabus classification also showed that SVM performs better than Naïve Bayes on a syllabus classification task. Therefore, we conducted this empirical study with the SVM method only.

SVM was first introduced in [2]. It is a two-class classifier that finds the hyperplane maximizing the minimum distance between the hyperplane and training data points. Specifically, the hyperplane $\omega^T x + \gamma$ is found by minimizing the objective function:

$$\frac{1}{2}||(\omega)||^2 \text{ such that } D(A\omega - e\gamma) >= e.$$

The distance is $\frac{2}{||\omega||^2}$. D is a vector of classes of training data, i.e., each item in D is $+1$ or -1. A is the matrix of features values of training data. e is the vector of ones. After ω and γ are estimated from training data, a testing item x will be classified as $+1$ if

$$\omega^T x + \gamma > 0$$

and -1 otherwise.

In some cases, it is not easy to find such a hyperplane in the original data space, in which case the original data space has to be transformed into a higher dimensional space by applying kernels. In this work, we focused on SVM without kernels.

In order to employ SVM on multi-class classification, we first conducted pairwise classification [8] and then decided a document's class by pairwise coupling [9]. In addition, sequential minimal optimization (SMO) [10], a fast nonlinear optimization method, was employed during the training process to accelerate training. The implementation of SVM discussed above is based on the Weka package [2].

VI. Evaluation

A. Performance Measures

We employed F_1 as the main performance measure. F_1 is a measure that trades off precision and recall, to provide an overall measure of classification performance. For each class on a training set, the definitions of the measures are:

- Precision: the percentage of the correctly classified positive examples among all the examples classified as positive.
- Recall: the percentage of the correctly classified positive examples among all the positive examples.
- F_1: $2*$ Precision * Recall / (Precision + Recall).

A higher F_1 value indicates better classification performance.

Since we used thousands of settings in the experiment, we employed several average measures to facilitate our analysis. The uses and formulas of these aggregated average measures are shown in Table III. These aggregated measures will be used in later graph comparison to illustrate the effects of different experimental settings.

B. Results and Discussions

We conducted analyses based on the average metrics shown in Table III and the results are summarized in five primary findings below.

Measures	Uses	Formulas
$\mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{c}_p\mathrm{avgF}_1$	measure the stable performance of a setting with the i^{th} size, the j^{th} distribution, and the kl^{th} feature selection method and the p class. The details of the settings are in Table IV.	N/A
$\mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{avgF}_1$	measure the impact of re-sampling towards uniform class distributions on different training sizes and feature selection methods	$\frac{1}{4}\sum_p \mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{c}_p\mathrm{avgF}_1$
$\mathrm{ds}_j\mathrm{c}_p\mathrm{avgF}_1$	measure the impact of re-sampling towards uniform class distributions on different classes.	$\frac{1}{10}\sum_i \frac{1}{7}\sum_k \mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{c}_p\mathrm{avgF}_1$
$\mathrm{tr}_i\mathrm{f}_k\mathrm{avgF}_1$	measure the impact of training sizes on feature selection methods.	$\frac{1}{2}\sum_j \frac{1}{4}\sum_p \frac{1}{N_k}\sum_l \mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{c}_p\mathrm{avgF}_1$
$\mathrm{f}_{kl}\mathrm{avgF}_1$	measure the impact of different DF thresholds sizes on classification performance.	$\frac{1}{10}\sum_i \frac{1}{2}\sum_j \frac{1}{4}\sum_p \mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{c}_p\mathrm{avgF}_1$
$\mathrm{ds}_j\mathrm{f}_k\mathrm{c}_p\mathrm{avgF}_1$	measure the impact of different feature selection methods on different classes with a varied or uniform distribution	$\frac{1}{10}\sum_i \frac{1}{N_k}\sum_l \mathrm{tr}_i\mathrm{ds}_j\mathrm{f}_{kl}\mathrm{c}_p\mathrm{avgF}_1$

TABLE III

AVERAGE MEASURES.

Settings	Explanations
tr_i	Training sets are of ten different sizes increasing with the orderings.
ds_j	0: original distribution 1: uniform distribution
f_{kl}	01: hybrid feature selection with DF=10 02: hybrid feature selection with DF=20 03: hybrid feature selection with DF=30 11: general feature selection with DF=10 12: general feature selection with DF=20 13: general feature selection with DF=30 21: genre feature selection
c_p	0: full syllabus 1: partial syllabus 2: entry page 3: noise page

TABLE IV

A VARIETY OF SETTINGS FOR TRAINING DATA SETS.

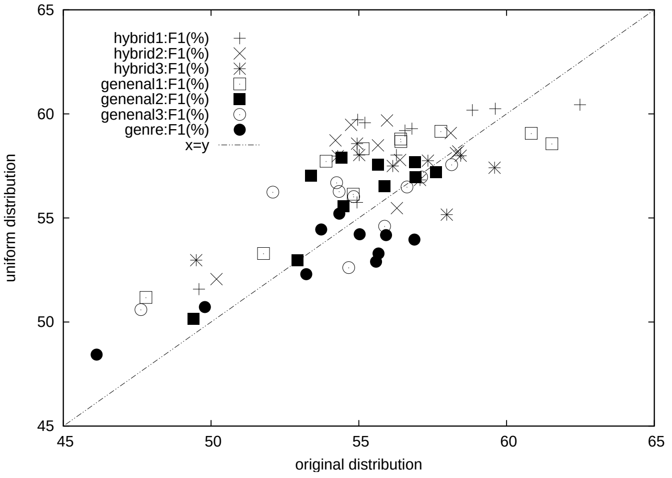


Fig. 1. Classification performance measured by F_1 on settings with different training data distribution. **hybrid**$_i$: hybrid feature selection with $DF = i \times 10$; **general**$_i$: DF as the feature selection method and $DF = i \times 10$; **genre**: manual selection of features specific to the syllabus genre. 10 data items are in each category, which represent performance with respect to 10 different training data sizes.

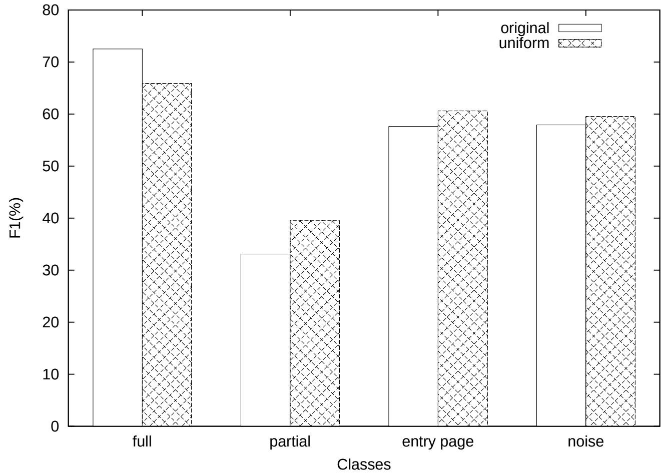


Fig. 2. Classification performance varied with classes.

1) The impact of class distribution on training data: Figure 1 shows the impact of class distribution measured by $tr_i\ ds_j\ f_{kl}$ avgF_1. The mean of the results is 0.56 and the standard deviation is 0.03. The best setting is the one with the largest training size, the original class distribution and the most features selected by hybrid feature selection method with $DF = 10$ ($\mathrm{tr}_9\mathrm{ds}_0\mathrm{f}_{0l}\mathrm{avgF}_1 = 0.63$). The worst is the one with the smallest training size, the original class distribution and the least features selected by genre feature selection method ($\mathrm{tr}_0\mathrm{ds}_0\mathrm{f}_{2l}\mathrm{avgF}_1 = 0.46$). Furthermore, although the performance variation of each setting on average is not significant, 67% of settings perform better after re-sampling towards a uniform class distribution. However, if only considering the settings with large training sizes such as tr_8 and tr_9, 71% settings perform worse after the re-sampling. In addition, we observed that 60% of settings with the genre feature selection method perform worse after re-sampling. Therefore, the class distribution on a training set has no impact on classification performance when the training data size is large to have enough samples for each class or far more than the number of features.

2) The performance with respect to each class: Figure 2 summarizes our further investigation on the performance of each class as measured by $\mathrm{ds}_j\mathrm{c}_p\mathrm{avgF}_1$. With uniform class distribution on training sets, a performance of 0.66 can be achieved for full syllabi, which is 65% better than the performance for the partial syllabi. Performance figures for the entry pages and the noise pages are close, both at around 0.60. Before re-sampling, the performance pattern on these four categories is the same but performance on full syllabi and partial syllabi differ even more. Therefore, on average, full syllabi are much easier to classify than partial syllabi by means of our training strategies. Furthermore, our classifiers favored classes with more examples in a training set. Figure2 shows that uniform re-sampling is beneficial when the sample size is small as in the case of partial, entry and noise pages. However, its usage hurts performance when training data is large as in the full syllabi case.

3) The correlation of feature selection methods and training size: We compared settings with different feature selection methods and different training sizes. Figure 3 shows the results measured by $\mathrm{tr}_i\mathrm{f}_k\mathrm{avgF}_1$. On average, the settings with the hybrid feature selection methods perform 1.8% better than those with the general feature selection methods and 7.6% better than those with the genre feature selection method. While the results imply that the larger the training size, the better the classification performance in general, nevertheless the variations of training sizes have dissimilar impact on different feature selection methods. As far as the settings with the hybrid and general feature selection methods are considered, their performances present the larger-size-better-performance trend especially when the training sizes are small (tr_0, tr_1 and tr_2) or large (tr_6, tr_7, tr_8 and tr_9). However, there is no much improvements with respect to the genre feature selection method especially considering tr_4 to tr_9. It is important to note that the number of

[2] http://www.cs.waikato.ac.nz/ml/weka/

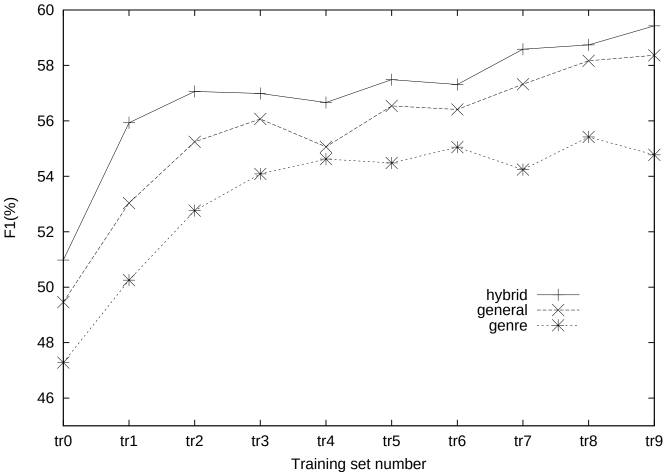


Fig. 3. Classification performance varied with training sizes and feature selection methods. $\mathbf{tr}_i$: the ith training set. The sizes of training sets increase when their ordering numbers increase.

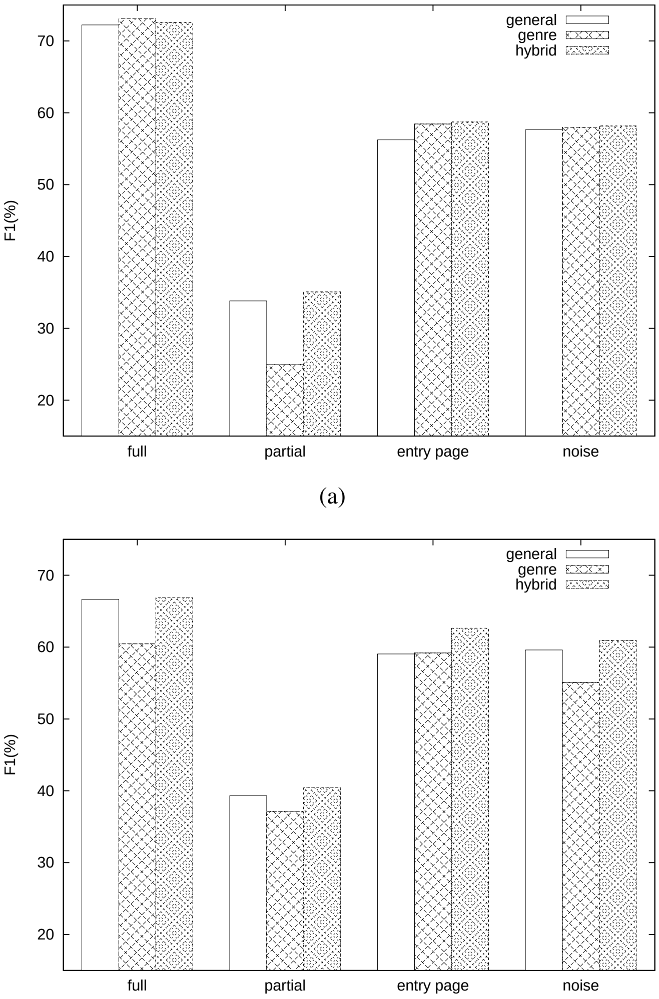
features from hybrid or general feature selection methods is larger than the maximum size of our training set, and the sizes of training sets tr_4 to tr_9 are far more than the number of genre features. Therefore, the settings with hybrid or general features might perform better given more training data. On the other hand, given a small size of training data, the genre feature selection method can achieve similar performance as general feature selection methods but with fewer computational resources.

4) The impact of feature selection methods on different classes: We showed the performances of settings with different feature selection methods on each class in Figure 4. We also took into account the original and uniform class distributions separately and showed the results in Figure 4 (a) and (b) respectively. With the original class distributions, the settings with genre features perform 3.9% better than those with general features on entry pages, 1.2% better on full syllabi and 0.6% better on noise pages. This finding indicates that our genre feature selection method can select important features for the syllabus genre. It also suggests the need that more genre features should be defined to differentiate partial syllabi from other categories. For example, it would be useful to capture features to differentiate between outgoing links to syllabus components and links to other resources such as a website with detailed information about a required textbook. It is also interesting to note that uniform re-sampling impacts the performance of settings with genre features more strongly than it impacts other feature types, especially on full syllabi and partial syllabi with a performance difference of −17.8% and +48% as compared to the performance before re-sampling. Overall, the hybrid feature selection method seems to be the best option for all four classes in both sample distributions.

5) The impact of different DF thresholds on feature selection methods: We also conducted analysis on general and hybrid feature selection methods with respect to different DF thresholds. The performance with DF threshold at 10 achieved 1.8% better than that with 20 and 6.8% better than that with 30 regarding the general feature selection methods. The comparison on hybrid feature selection methods reveals the similar results in Figure 5. This suggests that more features, selected when DF threshold value is set to 10, contributes more values to our syllabus classification performance. However, when more features are selected, more training time and computation resources are also required. More features also requires more training data. Thus we can consider to set DF threshold as 30 to reduce the feature size for further investigation on more settings with the current training data.

VII. Related Work

A few ongoing research studies are involved with collecting and making use of syllabi. Neves das [11] started from reading lists of a small set of manually collected computer science syllabi in the operating



(a)

(b)

Fig. 4. Classification performance on different classes with class distributions: (a) original distribution; (b) uniform distribution.

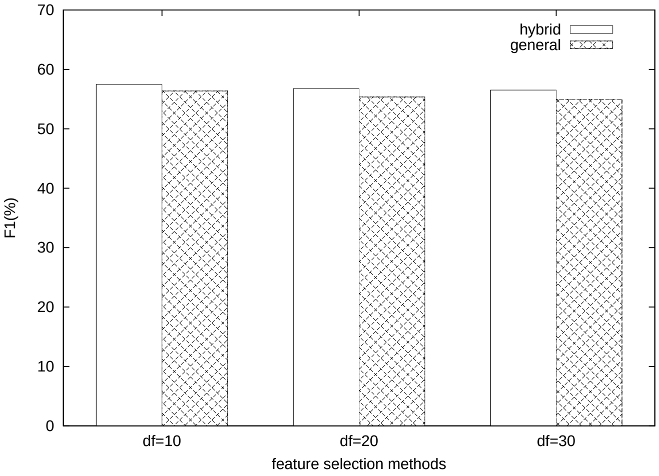


Fig. 5. Classification performance varied with different DF thresholds.

systems, information retrieval, and data mining courses, and 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 [12]. MIT OpenCourseWare manually collects and publishes 1,400 MIT course syllabi for public use. However, a lot of effort from experts and faculty are required in such manual collection building approaches, which is what our approach tries to address.

Furthermore, some effort has already been devoted to automating the syllabus collection process. A syllabus acquisition approach similar to ours is described in [13], but it differs in the way syllabi are identified. They crawled Web pages from Japanese universities and sifted through them using a thesaurus with common words which occur often in syllabi. A decision tree was used to classify syllabus pages and entry pages (for example, a page containing links to all the syllabi of a particular course over time). In [14], a classification approach was described to classify education resources – especially syllabi, assignments, exams, and tutorials. They relied on word features of each document and were able to achieve very good performance (F_1 score:0.98). Because we focused on classifying documents with a high probability of being syllabi into more refined categories, our best performance is 0.63 by the F_1 measure.

Our research also relates to genre classification. Research in genre classification aims to classify data according to genre types by selecting features that distinguish one genre from another, for example, identifying home pages [4] from web pages.

VIII. Conclusion

In this work, we described in detail our empirical study on syllabus classification. Based on observation of search results from the Web for the keyword query *'syllabus'*, we defined four categories: full syllabus, partial syllabus, entry page, and noise page. We selected features specific to the syllabus genre and tested the effectiveness of such a genre feature selection method as compared to the feature selection method usually used, DF thresholding. On average, DF thresholding performs better than genre feature selection method. However, further investigation revealed that genre feature selection method slightly outperforms DF thresholding on all classes except partial syllabi. Further study is needed regarding more features specific to the syllabus genre. For example, it might be worth including a few HTML tags (e.g. font size) as features. Another important finding is that our current training data size might be a factor that limits the performance of our classifier. However, it is always a laborious job to label a large set of data. Our future work will investigate the SVM variations on syllabus classification.

IX. Definitions

- A **syllabus component** is one of the following information: course code, title, class time and location, offering institute, teaching staffs, course description, objectives, web site, prerequisite, textbook, grading policy, schedule, assignment, exam and resources.
- A **full syllabus** is a syllabus without links to other syllabus components.
- A **partial syllabus** is a syllabus along with links to more syllabus components at another location.
- A **syllabus entry page** is a page that contains a link to a syllabus.
- **Text classification** is the problem of automatically assigning predefined classes to text documents.
- **Feature selection** for text documents is a method to solve the high dimensionality of the feature space by selecting more representative features. Usually the feature space consists of unique terms occurring in the documents.
- **Model training** is a procedure in supervised machine learning that estimates parameters for a designed model from data set with known classes.
- **Model testing** is a procedure performed after model training that applies the trained model to a different data set with known classes and evaluates the performance of the trained model.
- **Support Vector Machines (SVM)** is a supervised machine learning classification approach with the objective to find the hyperplane maximizing the minimum distance between the plane and the training data points.

Acknowledgments

This work was funded in part by the National Science Foundation under DUE grant #0532825. We'd also like to thank GuoFang Teng for the data preparation and the suggestions on the early work.

References

[1] M. Tungare, X. Yu, W. Cameron, G. Teng, M. Pérez-Quiñones, E. Fox, W. Fan, and L. Cassel, "Towards a syllabus repository for computer science courses." in *Proceedings of the 38th Technical Symposium on Computer Science Education (SIGCSE 2007)*, 2007.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*. New York, NY, USA: ACM Press, 1992, pp. 144–152. [Online]. Available: http://portal.acm.org/citation.cfm?id=130401

[3] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 412–420. [Online]. Available: http://portal.acm.org/citation.cfm?id=657137

[4] A. Kennedy and M. Shepherd, "Automatic identification of home pages on the web," in *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4*. Washington, DC, USA: IEEE Computer Society, 2005.

[5] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of the European Conference on Machine Learning, Springer*, 1998.

[6] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some effective techniques for naive bayes text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 11, pp. 1457–1466, November 2006.

[7] X. Yu, M. Tungare, W. Fan, M. Pérez-Quiñones, E. A. Fox, W. Cameron, G. Teng, and L. Cassel, "Automatic syllabus classification," in *(To appear in) Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL 2007*, 2007.

[8] J. Furnkranz, "Round robin rule learning," in *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 146–153. [Online]. Available: http://portal.acm.org/citation.cfm?id=645530.655685

[9] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*. Cambridge, MA, USA: MIT Press, 1998, pp. 507–513. [Online]. Available: http://portal.acm.org/citation.cfm?id=302744

[10] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," pp. 185–208, 1999. [Online]. Available: http://portal.acm.org/citation.cfm?id=299105

[11] F. das Neves, "Stepping stones and pathways: Improving retrieval by chains of relationships between documents," Ph.D. dissertation, Virginia Tech Department of Computer Science, Sep 2004.

[12] J. Pomerantz, S. Oh, S. Yang, E. A. Fox, and B. M. Wildemuth, "The core: Digital library education in library and information science programs," *D-Lib Magazine*, vol. 12, no. 11, November 2006.

[13] Y. Matsunaga, S. Yamada, E. Ito, and S. Hirokaw, "A web syllabus crawler and its efficiency evaluation," in *Proceedings of ISEE*, 2003.

[14] C. A. Thompson, J. Smarr, H. Nguyen, and C. Manning, "Finding educational resources on the web: Exploiting automatic extraction of metadata," in *Proc. ECML Workshop on Adaptive Text Extraction and Mining*, 2003.