Development of a National Syllabus Repository for Higher Education in Ireland

Arash Joorabchi and Abdulhussain E. Mahdi

Department of Electronic and Computer Engineering, University of Limerick, Ireland
{Arash.Joorabchi,Hussain.Mahdi}@ul.ie

Abstract. With the significant growth in electronic education materials such as syllabus documents and lecture notes available on the Internet and intranets, there is a need for developing structured central repositories of such materials to allow both educators and learners to easily share, search and access them. This paper reports on our on-going work to develop a national repository for course syllabi in Ireland. In specific, it describes a prototype syllabus repository system for higher education in Ireland that has been developed by utilising a number of information extraction and document classification techniques, including a new fully unsupervised document classification method that uses a web search engine for automatic collection of training set for the classification algorithm. Preliminary experimental results for evaluating the system's performance are presented and discussed.

1 Introduction

Syllabus documents are important and valuable educational materials in that they serve as one of the first initial contact points between the student and instructor/tutor and reflect a form of agreement between the student and the educational institute in terms of their expectations in relation to required prior learning, covered topics, assessment, qualification, regulations, and policies [1]. Currently, there is a lack of a centralised syllabus repository for higher education institutes in Ireland. This has resulted in inefficient storage and retrieval methods of often out-of-date syllabi, and prevented reusability of existing syllabus documents. This has necessitated the development of a structured repository that can hold syllabus documents covering the majority of courses offered by higher education institutes in Ireland. Such repository would benefit all parties involved. It would give students access to up-to-date syllabi and allows them to compare similar courses provided by different institutes and choose a course that matches their education background and interests them most. It would facilitate sharing and reuse of syllabi by helping course developers/tutors find candidate materials to reuse. It would also enable the institutes to gain competitive edge by facilitating comparisons of similar courses offered by different institutes and development of syllabi aimed at bridging knowledge and skills gaps in industry.

This paper describes the first prototype for an Irish syllabus repository system. The rest of the paper is organised as follows: Section 2 discusses the challenges in developing a structured syllabus repository and related work done to overcome some of the drawbacks. Section 3 describes the developed system and its various components in

B. Christensen-Dalsgaard et al. (Eds.): ECDL 2008, LNCS 5173, pp. 197–208, 2008.
© Springer-Verlag Berlin Heidelberg 2008

details. Section 4 describes the evaluation process carried out to assess the performance of the system, presenting and discussing some preliminary and experimental results. Section 5 concludes the paper and summaries our findings.

2 Challenges and Related Work

In this Section, we briefly review existing work and up-to-date developments in the fields of centralised repository systems, information extraction and electronic classification of documents as applied to syllabi, highlighting three major challenges in the development of a structured syllabus repository.

2.1 Unstructured Data

Electronic syllabus documents have arbitrary sizes, formats and layouts. These documents are intended for human readers, not computers and may contain complex layout features to make them easier to read (e.g., hidden tables for formatting, nested tables, tables with spanning cells), which make the information extraction task more complex and error-prone [2]. These characteristics makes electronic syllabus documents categorized as unstructured documents requiring sophisticated information extraction algorithms to automatically extract structured information form them. In this context, McCallum [3] gives a good overview of information extraction methods and discusses their application in syllabus domain. Yu and co-workers [4] have used the GATE natural language processing tool [5] to extract name entities such as persons, dates, locations, and organizations from the syllabus documents. This was followed by using Choi's C99 segmenter [6] to find the topic change boundaries in the text and classify the content between identified boundaries into one of the syllabus components (e.g., objectives section) by heuristic rules. Thompson et al. [7] explored the use of class HMMs to generate classificatory meta-data for a corpus of HTML syllabus documents collected by a web search engine.

2.2 Bootstrapping

A national syllabus repository for course syllabi for a given country needs to provide a rich collection of syllabi in a wide range of disciplines in order to attract the attention of all concerned in the higher education institutions in that country, motivating them to put in additional efforts to add their new syllabi to the repository and keep the existing ones up-to-date. In addition, the repository system should have a built-in mechanism for automatic collection of documents. Over the last few years, a number of techniques have been proposed for automatic collection of syllabus documents, particularly via searching the Internet and using the collected syllabi for bootstrapping a syllabus repository. Matsunaga et al. [8] developed a web syllabus crawler that uses the characteristics of syllabus web pages such as their keywords and the link structure between them to distinguish syllabus pages from other web pages. de Asis and co-workers [9] described a focused crawler for syllabus web pages that exploits both genre and content of web pages using cosine similarity function to determine the

similarly between the fetched web pages. Xiaoyan et al. [10] proposed utilising a generic search engine to search for syllabus documents and filter the search results by using an SVM classifier.

2.3 Classification

Large-scale digital libraries, such as our targeted syllabus repository, are intended to hold thousands of items, and therefore require deploying flexible query and information retrieval techniques that allow users to easily find the items they are looking for. Automated Text Classification or Categorization (ATC), i.e. automatic assignment of natural language text documents to one or more predefined categories or classes according to their contents, has become one of the key techniques for enhancing information retrieval and knowledge management of large digital collections. Sebastiani in [11] provides an overview of common methods for ATC, such as the Naive Bayes, k-NN, and SVM based techniques. Text classification algorithms have been successfully used in a wide range of applications and domains, such as spam filtering and cataloging news articles and web pages. However, to the best of our knowledge, ATC methods are yet to be adapted adequately for automatic classification of a large collection of syllabi based on a standard education classification scheme.

3 System Description

With the above in mind, we have recently developed a prototype for a national syllabus repository system for higher education in Ireland. The prototype is depicted in Fig.1, where the main processing stages and components of the system are illustrated.

A hot folder application communicating to an FTP server has also been added in order to provide an easy-to-use means for individuals and institutions to up-load their syllabi to the system. The application allows authorised contributors to easily add their syllabi to the repository by simply drag-and-drop their syllabus documents onto the hot-folder icon. The hot-folder application creates a zip package including all the submitted documents, along with a manifest file that contains some metadata about the package, such as submission date/time, institution name, and identity and contact details of submitting person. The package is then uploaded to the repository FTP server, whose contents are scanned and processed at regular pre-defined intervals by the meta-data generator module. The meta-data generator processes submitted syllabus documents generating meta-data for each and storing it along with the original document as a new record in the system's database shown in the figure. The following sections describe the design, implementation and operation of the four main components of the systems' meta-data generator.

3.1 The Pre-processing Unit

Having inspected a large number of documents originating from participating institutions, we discovered that the majority of existing syllabus documents are in PDF or MS-Word format. In order to reduce the complexity of the targeted system in

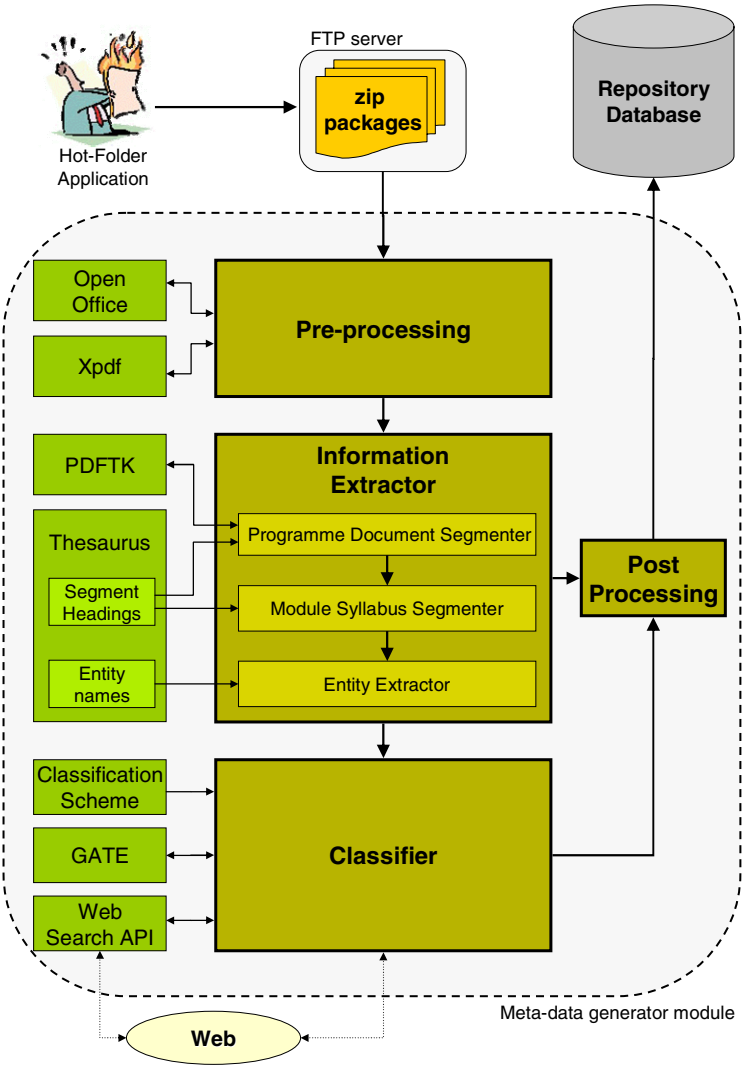

Fig. 1. Overview of developed repository system

terms of number of formats to be distilled and giving users the ability to access documents in their original format, our repository system has been designed to convert all the submitted documents to a unified format as a first step in generating the meta-data. Currently our system uses a PDF format as the unified format.

The pre-processing unit operates as follows: it checks the FTP server every one second and transfers any new zip packages to a queue on the repository's main server to be processed. After unzipping each package, all non-PDF files are converted to PDF using Open Office Suite [12]. All PDF documents are then converted into pure text with as much as possible preserved layout using the Xpdf application [13].

Finally the manifest file, PDF documents and the pure text representation of their content are passed to the information extraction component for distillation.

3.2 Information Extractor

Information extraction is the process of filling the fields and records of a database from unstructured or loosely formatted text. In our system, the role of the information extractor (IE) component is three-fold each of which is executed using one of the following sub-components.

3.2.1 Programme Document Segmenter

Most of the syllabus documents submitted to our system are envisaged to be in the form of complete course documents, each commonly referred to as the Definitive Programme Document (DPD). These documents are relatively large, usually comprising 100+ pages providing a detailed description of a full graduate or undergraduate programme of study. A DPD also contains the syllabi of all modules/subjects taught in a programme. The first sub-component in our IE component is a Programme Document Segmenter (PDS), whose main task is to find the boundaries, i.e. the start and the end, of each individual syllabus description inside a DPD. As discussed in Section 2.1, the number of variations in terms of both layout and content of syllabi is vast. However, inspection of a sample corpus containing a number of DPDs from a number of different institutes indicates that the syllabi inside these documents share a unified template. This feature yields a repeated pattern for the syllabi sections inside all DPDs, which is exploited by our PDS using a rule-base technique to define the boundaries of each individual syllabus inside a given DPD.

A module syllabus document is composed of a number of topical segments each describing a specific aspect of the course. Hence, our PDS incorporates a purpose-developed thesaurus which contains a list of potential terms/phrases that could be used for each segment's heading. For example, the segment that provides a description of the module's objectives can have any of these headings: "aims/objectives", "aims & objectives", "aims", "module aims", "description", etc. Using the segment heading entries in the thesaurus, the PDS identifies the location (i.e. page number) of each segment heading in the pure text version of the DPD under processing. Counting the number of times that each unique heading has been repeated identifies the number of individual module syllabi in the DPD. Having located all segment headings and identified the number of syllabi contained in a DPD, the PDS iterates through the segment headings to extract individual syllabi. The PDS designates the page where first heading appears as the start of the first syllabus in the processed DPD, and page where the same heading next appears as the start of the second syllabus, and so on, and uses corresponding page numbers to mark these boundaries. Locating the boundaries in terms of page numbers instead of line numbers is based on the fact that each individual syllabus starts in a new page. Therefore, locating the starting page and ending page is sufficient for extracting an individual syllabus. Hence, our PDS uses the assumption that page number corresponding to the end of each syllabus is equal to the page number corresponding to the beginning of next syllabus minus one. However, this assumption does not apply to the last module syllabus in the DPD as there are no more syllabi to follow. In order to avoid this problem, the page number where

the last heading appears is assigned to the ending boundary of the last syllabus. After identifying the first and last page numbers of each individual syllabus, the PDFTK toolkit [14] is used to split the individual module syllabi from the PDF version of the DPD under processing and store it in separate PDF files. Finally the individual syllabi in their both PDF and extracted pure text formats are passed to the Module Syllabus Segmenter sub-component of our IE for further processing. The PDS also sends a copy of each individual module syllabus in text format to the Classifier component to be classified.

3.2.2 Module Syllabus Segmenter

The task of the Module Syllabus Segmenter (MSS) is to extract the topical segments making each individual syllabus document. It uses a similar method to the one used in the PDS for splitting individual syllabi from DPDs. Regular expressions created from the segment headings in thesaurus are matched against the pure text version of a given module syllabus document to find the locations of segment headings in terms of line numbers. The MSS then iterates through the headings to extract the individual segments. The line number where the first heading appears is taken as the start of the first segment and the line number where the second heading appears as the start of the second segment and so on. Accordingly, the line number corresponding to the end of each segment is equal to the line number corresponding to the start of next segment minus one, with the exception of last segment whose end extends to the end-of-file position. The topic of each identified segment is the same as the topic of the segment heading that it follows. For Example, the term "module objectives" in thesaurus belongs to the topic of objectives and therefore the topic of the text string between the "module objectives" heading and the next identified heading is classified as objectives. During the post processing phase, this text string is saved in the objectives field of a module syllabus record in the database. The module syllabus documents usually start with a header segment that provides some administrative information about the module, such as module title, module code, module provider, number of credits, and module prerequisites, in form of either a table or name-value pairs. In almost all of investigated cases, no descriptive header line precedes the header segment which makes the header-based segmentation method ineffective in case of header segments. To overcome this problem, our MSS uses a different feature of the header segments to identify the boundaries of such segments. Header segments, as their name implies, are always the first segment to appear in a module syllabus document. Based on this feature and the fact that at this stage we are only dealing with documents each containing the syllabus of an individual module, we can confidently assume that the string of text between the start-of-file position and the first segment heading identified by the header-based method of our MSS should be classified as the header segment.

After identifying and extracting all the segments, the MSS passes the results to the Named Entity Extractor sub-component of our IE to perform the final stage of information extraction process.

3.2.3 Named Entity Extractor

Named Entity Extraction is the task of locating and classifying atomic elements in natural language text (e.g., words, phrases, and numbers) that can be classified into predefined categories e.g., names of persons, organisations and locations.

The task of the Named Entity Extractor (NEE) sub-component of our IE is to extract syllabus related named entities such as module name and module code from the segmented syllabi. It focuses on extracting a set of common attributes in the majority of syllabi that would allow syllabus documents to be managed, located, and reused. These attributes include module code, module name, module level, number of credits, pre-requisites and co-requisites. All of these administrative attributes appear in the header segment of syllabus documents and, hence, this feature allows the NEE to reduce the scope of search to the header segment of syllabus which has already been extracted by the MSS. The thesaurus contains lists of potential terms that could be used for the name of each attribute. The rule is that these attribute names appear right before the attribute values and therefore can be used to locate corresponding attribute values. For example, the value of module name attribute can be preceded by terms such as "module name", "module title", "subject title", "subject name", "full title", and "course title". The NEE creates a group of regular expressions based on the potential attribute names in the thesaurus and matches them against the header segment of the syllabus to extract the required attribute values.

3.3 Classifier

The task of the Classifier component is to automatically assign a classification code to each individual course/module based on a predefined education classification scheme. Currently, the Higher Education Authority (HEA) uses the International Standard Classification of Education (ISCED) [15] to provide a framework for describing statistical and administrative data on educational activities and attainment in Ireland. This classification scheme is suitable for subject/discipline based classification of full undergraduate or postgraduate programmes. However, it does not provide the level of detail required for classifying individual modules. In order to standardise the classification of modules among all higher education institutes in Ireland, the HEA is currently considering the development of an Irish Standard Classification of Education scheme. The current version of the classifier component in our system classifies module syllabus documents based on an in-house developed, extended version of the ISCED, which we plan to replace by proposed Irish Standard Classification of Education scheme when such scheme becomes available.

The underpinning approach of our classifier is the Multinomial Naïve Bayes algorithm, implemented with the addition of a new web-based method for automatic collection of a classification training set, as described in the following sections.

3.3.1 Multinomial Naive Bayes

The Multinomial Naïve Bayes algorithm , as described in [16], is expressed as:

$$C_{MNB} = \arg\max_{i \in V} \left[\log P(Class_i) + \sum_{k=1}^{|Document_j|} f_{wk} \log P(w_k \mid Class_i) \right], \qquad (1)$$

where V is a set of all possible target classes, The class prior probability, $P(Class_i)$, can be estimated by dividing the number of documents belonging to $Class_i$ by the total number of training documents, f_{wk} is the frequency of word k in document$_j$ and the class-conditioned (word) probability, $P(w_k \mid Class_i)$, is estimated by:

$$P(w_k \mid Class_i) = \frac{n_k + 1}{n + \mid Vocabulary \mid}, \quad (2)$$

where n_k is the number of times the word occurs in the training documents which belong to $Class_i$, n is the total number of words in the training documents which belong to the $Class_i$, and *Vocabulary* is a set of all distinct words which occur in all training documents. Each estimate is primed with a count of one to avoid probabilities of zero (Laplace smoothing).

3.3.2 Web-Based Unsupervised Training Method

A major difficulty with the use of supervised approaches for text classification is that they require a very large number of training instances in order to construct an accurate classifier. For example, Joachims [17] measured the accuracy of Bayes classifier with a dataset of 20,000 Usenet articles, called 20-Newsgroup collection. She reported that the Bayes classifier achieves the highest accuracy of 89.6% when trained with 13,400 documents (670 documents per class). The accuracy of her classifier dropped to 66% when 670 documents (33 documents per class) were used to train the classifier. Motivated by this problem, a number of researchers have attempted to develop/train classifiers using semi-supervised and unsupervised training methods with a limited number of training documents for the first type of methods, and no training documents for latter type of methods (*See* [18] *for examples*). Following this trend in developing our system, we have investigated the use of a new un-supervised web-based approach to train a Naïve Bayes classifier for classifying syllabus documents based on a hierarchical education classification scheme.

The classification scheme we used, is an extended version of ISCED [15]. The ISCED is a hierarchical scheme with three levels of classification: broad field, narrow field, and detailed field. Accordingly, the scheme uses a 3-digit code in a hierarchical fashion for classifying fields of education and training. We have extended this by adding a forth level of classification, subject field, which is represented by a letter in the classification coding system. For example a module assigned the classification code "482B" would indicate that module belongs to the broad field of "Science, Mathematics and Computing", the narrow field of "Computing", the detailed field of "Information Systems" and the subject field of "Databases".

The classifier starts the training process by reading the XML version of classification scheme and collecting a list of subject fields (leaf nodes). Then a search query, created from the name of the first subject field in the list combined with the keyword "syllabus", is submitted to the Yahoo search engine using the Yahoo SDK [19]. The first hundred URL's in the returned results are passed to the Gate toolkit [5], which downloads all corresponding files, extracts and tokenizes their textual contents. This process is repeated for all the subject fields in the hierarchy. The tokenised text documents resulting from this process are then converted to word vectors, which are used to train our system's classifier to classify syllabus documents at the subject-field level, and to create word vectors for the fields which belong to the upper three levels of the classification hierarchical tree.

The subject-field word vectors created by leveraging a search engine are used in a bottom-up fashion to construct word vectors for the fields which belong to the higher

levels of hierarchy. We illustrate this method with help of the following example. Let us assume that we want to create a vector of words for the detailed field of "information systems". There are four subject fields that descend from this field in our classification scheme: "systems analysis and design", "databases", "decision support systems", and "information systems management". We build a master vector by combining the vectors corresponding to these four subject fields and then normalise the word frequencies by dividing the frequency of each word in the master vector by the total number of subject field vectors used to create it, i.e. by 4 in this case. We then round-up the quotient to its nearest positive integer number. During the normalisation process, if the frequency of a word is less than the total number of vectors, that word is removed from the vocabulary. In specific, we use a feature reduction technique which reduces the size of vocabulary by removing all words whose frequency is below a certain threshold. The method can be formalised as:

$$F(w_i) = \begin{cases} 0 & if \quad FreqSum < |Fields| \\ RND\left(\frac{FreqSum}{|Fields|}\right) & if \quad FreqSum \geq |Fields| \end{cases} \tag{3}$$

$$FreqSum = \sum_{n=1}^{|Fields|} Freq(w_i \mid Field_n).$$

As stated, the method is used in our system to create word vectors for all the detailed, narrow and broad fields of the classification hierarchy in a bottom-up manner. In rare cases where a detailed or narrow field does not have any descendent, the web-based approach is used to create a word vector for it.

3.4 Post-processing

The task of Post-processing Unit is to store generated meta-data for each module syllabus document along with a copy of the original document as a new syllabus record in the repository's relational database. It uses the results produced by the Pre-processing Unit, the IE, and the Classifier to fill up the fields of new syllabus records.

4 System Evaluation and Experimental Results

Two hundred syllabus documents from five different institutes participating in our project were randomly selected to evaluate the performance of the information extractor component. The standard information extraction measures of Precision, *P*, Recall, *R,* and their harmonic mean, *F*1, [20] were adopted to evaluate the performance of our system's IE.

Table 1. Information extraction performance

	P_m	R_m	$F1_m$
Named Entities	0.91	0.73	0.81
Topical Segments	0.83	0.75	0.78

We apply the micro-average to the target performance measures of precision, recall, and F1 over two categories of named entities and topical segments. Micro-average can be calculated by regarding all sub-categories as the same category and then calculate its precision, recall, and F1 values. Table 1 shows the results.

Inspecting above results and examining the syllabus documents used to generate them indicate a number of issues which adversely affected the accuracy of the information extraction process:

- The module name and module code entities in few of the processed module syllabus documents appeared in a large font size at the beginning of the document with no prefix and, therefore, were not extracted resulting in a consequent decrease in named entities recall.
- Both the NEE and the MSS of our system use the thesaurus to identify the named entity prefixes and segment headings respectively. Hence, the occurrence of a named entity prefix or subject heading that do not appear in the thesaurus results in that named entity or segment not being extracted and, consequently, decreasing corresponding recall decreases.
- In a few cases, the named entity was longer than one line of text or broken down into a few lines within a table cell. This tends to confuse our IE and results in a partial extraction of such named entities, which in turn decreases the precision of named entities.
- In situations where an identified segment is followed by an un-identified one, the un-identified segment was assumed as being part of the previous identified segment. This problem tends to decrease the topical segments precision of our system.

For assessing the performance of our Classifier, we used the micro-average precision measure, P_m, which is computed as follows:

$$P_m = \frac{\text{Total number of correctly classified documents in all classes}}{\text{Total number of classified documents in all classes}} . \quad (4)$$

The performance of the classifier was measured using one hundred undergraduate syllabus documents and the same number of postgraduate syllabus documents. The micro-average precision achieved for undergraduate syllabi was 0.75, compared to 0.60 for postgraduate syllabi. Examining syllabi from both groups of documents indicates that some syllabi are describing modules/subjects which contain components belonging to more than one field of study. Classifying such documents, which effectively belong to more than one class, is more error-prone and requires the Classifier to recognise the core component of the module. Since the number of cross-subjects modules is substantially higher among the group of postgraduate courses compared to

those on undergraduate courses, the classification accuracy achieved for the first group of syllabus documents is about 15% lower than that of the second group.

5 Conclusion and Future Work

In this paper, we have discussed the necessity for developing a national syllabus repository for higher education in Ireland, reviewed similar reported works done by researchers in other countries, and described what we have achieved to-date in our venture to develop a national repository for course syllabi in Ireland. In future, we plan to improve the accuracy of the classifier by automatic filtration of training documents obtained by the search engine to increase the percentage of valid training documents. Also, we plan to investigate the potential enhancement of the information extractor component by adding a table detection & extraction sub-component to it.

References

[1] Marcis, J.G., Carr, D.: A note on student views regarding the course syllabus. Atlantic Economic Journal 31(1), 115 (2003), http://dx.doi.org/10.1007/BF02298467

[2] Embley, D.W., Hurst, M., Lopresti, D., Nagy, G.: Table-processing paradigms: a research survey. International Journal on Document Analysis and Recognition 8(2-3), 66–86 (2006), http://dx.doi.org/10.1007/s10032-006-0017-x

[3] Mccallum, A.: Information extraction: distilling structured data from unstructured text. Queue 3(9), 48–57 (2005), http://dx.doi.org/10.1145/1105664.1105679

[4] Yu, X., Tungare, M., Fan, W., Yuan, Y., Pérez-Quiñones, M., Fox, E.A., Cameron, W., Cassel, L.: Using Automatic Metadata Extraction to Build a Structured Syllabus Repository. In: Proceedings of the 10th International Conference on Asian Digital Libraries (ICADL 2007), Ha Noi, Vietnam (December 2007), http://manas.tungare.name/publications/yu_2007_using

[5] 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, US (July 2002), http://gate.ac.uk/gate/doc/papers.html

[6] Choi, F.: Advances in domain independent linear text segmentation. In: Proceedings of the first conference on North American chapter of the Association for Computational Linguistics (NAACL 2000), Seattle, USA (April 2000), http://arxiv.org/abs/cs/0003083

[7] Thompson, C., Smarr, J., Nguyen, H., Manning, C.D.: Finding Educational Resources on the Web: Exploiting Automatic Extraction of Metadata. In: Proceedings of the ECML Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia (September 2003), http://nlp.stanford.edu/publications.shtml

[8] Matsunaga, Y., Yamada, S., Ito, E., Hirokawa, S.: A Web Syllabus Crawler and its Efficiency Evaluation. In: Proceedings of the International Symposium on Information Science and Electrical Engineering 2003 (ISEE 2003), Fukuoka, Japan (November 2003), https://qir.kyushu-u.ac.jp/dspace/bitstream/2324/2972/1/2003_d_2.pdf

[9] de Assis, G., Laender, A., Gonçalves, M., da Silva, A.: Exploiting Genre in Focused Crawling. In: String Processing and Information Retrieval, pp. 62–73. Springer, Heidelberg (2007)

[10] Xiaoyan, Y., Manas, T., Weiguo, F., Manuel, P.-Q., Edward, A.F., William, C., Guo-Fang, T., Lillian, C.: Automatic syllabus classification. In: Proceedings of the ACM IEEE Joint Conference on Digital Libraries, Vancouver, BC, Canada (June 2007), http://doi.acm.org/10.1145/1255175.1255265

[11] Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34(1), 1–47 (2002)

[12] OpenOffice.org 2.0 (sponsored by Sun Microsystems Inc., released under the open source LGPL licence, 2007), http://www.openoffice.org/

[13] Xpdf 3.02 (Glyph & Cog, LLC., Released under the open source GPL licence, 2007) http://www.foolabs.com/xpdf/

[14] Steward, S.: Pdftk 1.12 - the PDF Toolkit (sponsored by AccessPDF, Released under the open source GPL licence, 2004), http://www.accesspdf.com/pdftk/index.html

[15] International Standard Classification of Education - 1997 version (ISCED 1997) (UNESCO, 2006) [cited 2007 December], http://www.uis.unesco.org/ev.php?ID=3813_201&ID2=DO_TOPIC

[16] McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Proceedings of the AAAI 1998 Workshop on Learning for Text Categorization, Wisconsin, USA (1998), http://www.cs.umass.edu/~mccallum/papers/multinomial-aaai98w.ps

[17] Joachims, T.: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA. Morgan Kaufmann Publishers Inc., San Francisco (1997)

[18] Seeger, M.: Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh (2000), http://www.kyb.tuebingen.mpg.de/bs/people/seeger/papers/review.pdf

[19] Yahoo! Search Web Services Software Development Kit (Yahoo! Inc (2007), http://developer.yahoo.com/search/

[20] Appelt, D.E., Israel, D.: Introduction to Information Extraction Technology. In: Proceedings of the 16th international joint conference on artificial Intelligence (IJCAI 1999), Stockholm, Sweden (August 2, 1999), http://www.ai.sri.com/~appelt/ie-tutorial/IJCAI99.pdf