An Automated Syllabus Digital Library System for Higher Education in Ireland

Arash Joorabchi and Abdulhussain E. Mahdi

Department of Electronic and Computer Engineering, University of Limerick, Ireland

Abstract

Purpose - 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 robust central repositories of such materials to allow both educators and learners to conveniently share, search and access them. This paper reports on our work to develop a national repository for course syllabi in Ireland.

Design/methodology/approach – The paper describes a prototype syllabus repository system for higher education in Ireland, which 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.

Findings – Preliminary experimental results for evaluating the performance of the system and its various units, particularly the information extractor and the classifier, are presented and discussed.

Originality/value – In this paper, we identify three major obstacles associated with creating a large-scale syllabus repository, and provide a comprehensive review of published research work related to addressing these problems. We also identify two different types of syllabus documents and describe a rule-based information extraction system capable of extracting structured information from unstructured syllabus documents. Finally, we highlight the importance of classifying resources in a syllabus digital library, introduce a number of standard education classification schemes, and describe our unsupervised automated document classification system which classifies syllabus documents based on an extended version of the International Standard Classification of Education (ISCED 1997).

Keywords: syllabus repository, information extraction, document classification, digital library.

Article Type: Research paper

1 Introduction

In general, a syllabus is a summary of a course/module of study providing information about various aspects of the course, such as aims, objectives, learning outcomes, schedule, recommended resources, and assessment methods. Syllabus documents are important and valuable educational materials in that they serve as initial contact points between a student and an instructor/tutor and represent some 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 - *See* (Marcis & Carr 2003) *for a survey on student views regarding course syllabi*. Currently, there is a lack of a properly structured and centralised repository for electronic syllabus documents for higher education in Ireland. This has resulted in inefficient storage and retrieval methods of often out-of-date syllabi, and prevented reusability of existing syllabus documents. Hence, there is a recognised need for the development of a national 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.

With above in mind, the Higher Education Authority in Ireland has recently initiated a number of related projects nationally. One such major project is the IDEAS (IDEAS 2007) which is led by the University of Limerick. The aim of this project is to develop an information portal for continuing professional development of non-traditional students in Ireland up to and including postgraduate masters and professional doctorates. The portal is to provide an expert self-assessment system to assess qualifications and experience and produce customised advice on knowledge acquisition and career progression. To facilitate this, information on modules/courses appropriate to individually customised plans have to be appropriately extracted and processed. The latter process necessitated the development of a national syllabus repository to act as the infrastructure for IDEAS system, and enables the IDEAS search engine to perform semantically rich searches over its comprehensive collection of syllabus records to retrieve syllabi of courses/modules that match a student's requirements.

This paper describes the first prototype for an Irish syllabus repository system that has been developed within the IDEAS framework. 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 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. They can include a single or multiple module descriptions and be in different formats, e.g. PDF, Microsoft Word and HTML. The number of layouts is virtually unlimited as each institute uses its own layouts which may also change over time. In some cases, each department within an institute may create its own templates for presenting existing syllabus documents in various formats as required for different processes. The electronic syllabus 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, and tables with spanning cells), which make the information extraction task more complex and error-prone (Embley et al. 2006). These characteristics makes electronic syllabus documents categorized as unstructured documents requiring sophisticated information extraction algorithms to automatically extract structured information (e.g. module name, objectives, pre-requisites, and time-table) form them. In general, information extraction approaches can be divided into two main categories (Appelt & Israel 1999): rule-based methods and machine learning methods. In rule-based systems, a human expert discovers the domain patterns through inspection of a corpus in a target domain and constructs and tunes extraction rules manually. In machine learning approaches, however, statistical algorithms such as Hidden Markov Model (HMM) are deployed to discover extraction patterns in training data. In this context, McCallum (McCallum 2005) gives a good overview of information extraction methods and discusses their application in syllabus domain. Yu and co-workers (Yu et al. 2007) have used the GATE natural language processing tool (Cunningham et al. 2002) to extract name entities such as persons, dates, locations, and organizations from the syllabus documents. This was followed by using a text segmenter 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 and co-workers (Thompson et al. 2003) explored the use of class HMMs to generate classificatory meta-data for a corpus of HTML syllabus documents collected by a web search engine. In order to avoid the problem of unstructured syllabus documents, some research has been carried out on creation of metadata standards and schema for syllabus domain and development of software systems that allow faculty members/course developers to easily generate and publish structured syllabus documents using markup languages, such as XML - *see* (Cebeci et al. 2006; Ida et al. 2005) *for examples*. As reported in the literature, D.A. Black from Seton Hall University is the first researcher who developed an XML schema for syllabi called SyML – the Syllabus Markup Language in 2002. However, lack of an internationally-recognised open standard still remains a key problem for interoperability and sharing of syllabi data between various systems. The DC-Ed (DCMI education community 1999) and the IEEE-LOM (IEEE-LTSC-WG12 2002) standards developed, by Dublin Core Metadata Initiative (DCMI) Education Community and IEEE Learning Technology Standards Committee respectively, specify the syntax and semantics of Learning Objects Metadata. However, they do not provide the specific metadata elements required for describing syllabus objects. Developing a widely accepted standard for syllabus genre and its machine-readable bindings (e.g., XML, and RDF) provides the necessary framework for building software systems that allow faculty members to create fully interoperable structured syllabus documents.

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 and co-workers (Matsunaga et al. 2003) 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. Assis and co-workers (Assis et al. 2007) 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 and co-workers (Xiaoyan et al. 2007) proposed utilising a generic search engine to search for syllabus documents and filter the search results by using an SVM classifier. Cohen's syllabus finder (Cohen 2006) is a specialised search engine for syllabi. It optimises user's search query, sends it to Google search engine, and combines the returned result with the results of simultaneous searches on in-house databases (e.g., a database of educational institutions, so it can identify which university or college a syllabus comes from).

2.3 Classification

A library classification is a system of coding and organising library materials according to their subjects that simplifies subject browsing. Library classification systems have been used by catalogers to classify books and other materials in physical libraries for over a century. The two major classification systems used today in libraries around the world are the Dewey Decimal Classification system (DDC) and the Library of Congress Classification system (LCC). Since their introduction in the late 18th century, these two systems have undergone numerous revisions and updates. Large-scale digital libraries, such as our targeted syllabus repository, are intended to hold thousands of items just like physical libraries, and therefore require deploying flexible query and information retrieval techniques that allow users to easily find the items they are looking for. In order to provide highly refined search results, the system needs to go beyond the traditional keyword-based search techniques which yield a large volume of indiscriminant search results irrespective of their content. Classification of materials in a digital library based on a pre-defined scheme improves the accuracy of information retrieval significantly and allows users to browse the collection by subject. However, manual classification is a tedious and time-consuming job requiring an expert cataloger in each knowledge domain represented in the collection and, therefore, deemed unfeasible in many cases. 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 (Sebastiani 2002) provides an overview of common machine learning-based methods for ATC, such as naive Bayes, k-NN, and SVM techniques. These 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 such as International Standard Classification of Education(ISCED 1997).

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. As indicated, the core system which resides on a dedicated server is effectively a meta-data generator comprising a Pre-processing Unit, an Information Extractor, a Classifier, and a Post-processing Unit.

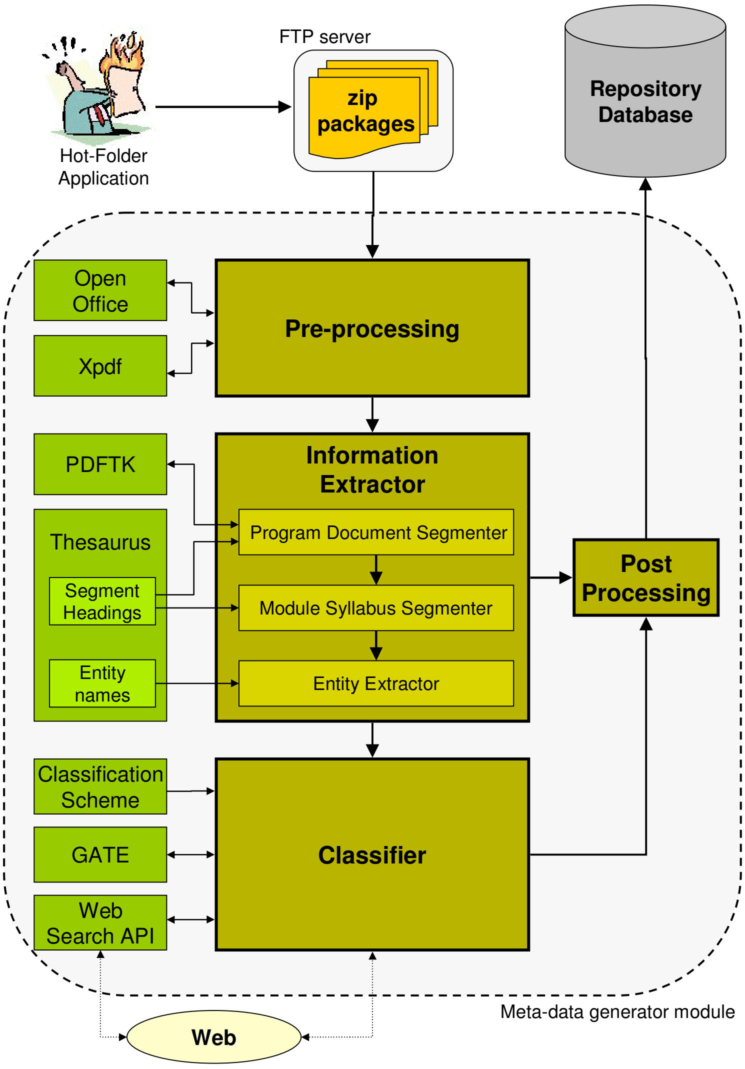

Figure 1. Overview of developed repository system.

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 each submitted syllabus document, generates meta-data for it, and stores the results along with the original document as a new record in the system's database, as shown in figure 1. 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

Our initial assumption was that syllabus documents we would be dealing with could be in any format. However, 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 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 portable format as a first step in generating the meta-data. Currently our system uses a PDF format as the unified format, due to a number of reasons. Firstly, PDF is a de facto open standard (the full PDF 1.7 specification has been submitted to ISO for approval as a formal, open standard, named ISO 32000) and therefore is fully usable as a free document exchange format, as compared to the proprietary MS-Word format. Secondly, a large proportion of documents received so far are already in PDF format. Thirdly, PDF files can be viewed on nearly every operating system and PDF viewers, such as Adobe Reader are freely available. Finally, information Extraction from PDF-formatted documents is an active field of research and various tools and techniques have been developed to support the extraction process.

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 main server to be processed one at a time using FIFO approach. After unzipping each package, all non-PDF files (except the manifest file) are converted to PDF using Open Office Suite (*OpenOffice.org 2.0* 2007). All PDF documents are then converted into pure text with as much as possible preserved layout using the Xpdf application (*Xpdf 3.02* 2007). 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”, “aim”, “module aims”, “description”, “module 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 (Steward 2004) 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. Topical segmentation can significantly improve the accuracy of both named entity extraction and information retrieval by reducing the scope of search from the entire document to only one segment of document that is most expected to find the required data in. For example, if we are looking for the number of credits assigned to a given module, then the scope of search will become limited to the header segment. Reducing the scope of search reduces the number of potential candidates and therefore increases the probability of choosing the correct candidate.

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 such as names of persons, organisations and locations. An NEE system is usually interested in a specific set of named entities in relation to the domain/genre that it works in. For example, a named entity extraction system working in news domain would be interested in extracting named entities such as people, cities, countries, companies, government organizations, committees, and events; whereas a named entity extraction system working in a biomedical domain would be interested in extracting named entities such as genes, organisms, malignancies, and chemicals.

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”, “course name”, 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. The output of the NEE coupled with the outputs of the MSS and PDS are passed to the Post-processing Unit to be used for creating new syllabus records in the repository database.

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) and higher education institutions in Ireland use the International Standard Classification of Education (ISCED) 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. The need for a more detailed national education classification standard than that provided by the ISCED has already been recognised by educational authorities within other jurisdictions. This has led some other countries to develop their own national classification of education standards such as the JACS in the UK (JACS 2007) and the ASCED in Australia (Trewin 2001). 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 classifier of our system is a new fully unsupervised syllabus document classification system which has been developed in-house. The underpinning approach of our classifier is the widely used Naïve Bayes algorithm (Mitchell 1997), implemented with the addition of a new web-based method for automatic creation of a classification training set, as described in the following sections.

3.3.1 Multinomial Naive Bayesian Classifier

The underlying theorem for the Naïve Bayesian text classification is the Bayes rule, expressed as:

$$P(A_i|B_j) = \frac{P(A_i)P(B_j|A_i)}{P(B_j)} \quad . \qquad (1)$$

The rule as expressed in (1) enables the calculation of the likelihood of event A_i given that B_j has happened. When applied to text classification, eq.1 can be rewritten as:

$$P(Class_i \mid Document_j) = \frac{P(Class_i)P(Document_j \mid Class_i)}{P(Document_j)} \quad , \qquad (2)$$

Such that the Rule is used to calculate the probability of each predefined $Class_i$ given $Document_j$, and the Class with the highest probability is allocated to $Document_j$. In eq.2, $P(Document_j)$ is a constant divider, common to every calculation and therefore can be safely removed from the equation.

In this model each documents is represented as a vector of words in a multidimensional space, where each dimension corresponds to a distinct word and the distance along that dimension is the number of times that word occurs in the document. In a common supervised setting, a set of manually classified training documents is used to parameterise the class prior probabilities, $P(Class_i)$, and the class-conditioned (word) probabilities, $P(Document_j \mid Class_i)$. The conditional probability of each word appearing in the vocabulary, w_k, in a given class, $Class_i$, is estimated by:

$$P(w_k \mid Class_i) = \frac{n_k + 1}{n + |Vocabulary|} \quad , \qquad (3)$$

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). The class prior probability, P($Class_i$), can be estimated by dividing the number of documents belonging to $Class_j$ by the total number of training documents. It follows that if a document, $Document_j$, is to be classified the most likely class, C_{NB}, for that document would be determined as:

$$C_{NB} = \arg\max_{i \in V} \left[P(Class_i) \prod_{k=1}^{|Document_j|} P(w_k \mid Class_i) f_{wk} \right], \qquad (4)$$

where V is a set of all possible target classes and f_{wk} is the frequency of word k in the test document.

Multiplying a large number of probabilities, which by definition have values between 0 and 1, can result in floating-point underflow. To avoid this problem, we perform all computations by summing logarithms of the probabilities rather than multiplying probabilities:

$$C_{NB} = \arg\max_{i \in V} \left[\log P(Class_i) + \sum_{k=1}^{|Document_j|} f_{wk} \log P(w_k \mid Class_i) \right]. \qquad (5)$$

Finally, the class with highest final probability is chosen as the target class for the document under processing. Reducing the size of the vocabulary by selectively choosing the words to provide as an input to the learning algorithm can improve both the accuracy and scalability of the classification. In this work, stop-word removal is used to reduce the size of vocabulary by excluding the words that appear frequently in all the training documents and, therefore, have no predictive value. We used a generic stoplist of 526 words, which contains common words such as "*the*", "*of*", and "*is*", and enhanced it with a syllabus domain-specific stoplist of 50 words, such as "student", "semester", "lecturer", etc. We do not use stemming or any other common feature selection/reduction methods. However, during the unsupervised training process a word frequency threshold is used for non-leaf nodes in the classification scheme hierarchy tree, as will be detailed in the next Section.

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 (Joachims 1997) measured the accuracy of Bayes classifier with a dataset of 20,000 Usenet articles originally collected by Ken Lang from 20 different newsgroups (1000 articles each), 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. As this and other experimental results show, increasing the size of training corpus improves the accuracy of the classifier substantially. However, manual classification of documents for the purpose of training a classifier is a tedious and expensive job. 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 (Zhu 2005) for a review of these techniques. 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 used in our system is an extended version of ISCED (ISCED 1997) represented using XML. 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 , such that the first digit represents 'broad field', the second digit represents the 'narrow field' and third digit resents the 'detailed field' of a given document. There are 9 broad fields, 25 narrow fields and about 80 detailed fields. 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", where the broad fields, narrow fields and detailed fields represent the branches of the upper three levels of the classification hierarchical tree, from top to bottom respectively, and the subject fields represent the leaves of the tree. Representing the hierarchy in XML format makes it machine readable and determines the parent-child relationship of fields. An instance of a field is represented by an XML element which has a number of attributes such as code, name, and description. All the nodes descend from the document root element and, with the exception of subject-field elements, they have one or more internal field elements as their children.

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 (Yahoo-API 2007).For example, if the subject field is "databases " which is assigned the code 482B in our classification scheme, then a query is submitted to the Yahoo SDK using the combined words "databases syllabus". The first hundred URL's in the returned results are passed to the Gate toolkit (Cunningham et al. 2002), which downloads all corresponding files (HTML, Text, PDF, or MS-Word), 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 words used in the names of subject fields have direct effect on the quality of search results, and using words that have a high information gain value improves the quality of search results. However, in developing our system, we did not investigate the effect of changing subject field names, as our objective was to measure the accuracy of the system with a standard classification scheme in its original form. The other factor affecting the quality of search results is the number of teaching and learning related documents in each field available online. For example, the quality of search results for computer related fields such as databases, programming languages, and artificial intelligence, is substantially higher than fields such as veterinary nursing or cereal science which are less populated. Our investigation showed that the number of relevant syllabus documents retrieved from the first hundred URL's of search result can vary between 20 and 40 depending on the two above-described factors. However, although the rest of the resulting documents are not syllabus documents, the majority of them can be classified as belonging to the subject field or its parent, i.e., its detailed field, which makes them useful for training the classifier. As an example, looking for syllabus documents in the subject field of "databases", a large number of the retrieved documents might not be database-related syllabus documents, but can still be classified to the subject field of databases as their main content covers some aspect of database systems. In addition, for the majority of cases where the retrieved document can not be classified to the subject field of databases, they can still be classified to the detailed field of "computer science" which is the parent of the subject field "databases" in our classification scheme.

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, as illustrated in Fig. 2. 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. Besides reducing the size of the vocabulary, this technique also prevents the occurrence of decimal frequency values below 1. Reducing the size of vocabulary and preventing decimal frequency values both contribute to improving the speed of the classification process. The method described above can be formalised as follows:

$$F(w_i) = \left\{ \begin{array}{ccc} 0 & if & FreqSum < |Fields| \\ RND\left(\dfrac{FreqSum}{|Fields|}\right) & if & FreqSum \geq |Fields| \end{array} \right\}, \qquad (6)$$

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

As stated, this 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.

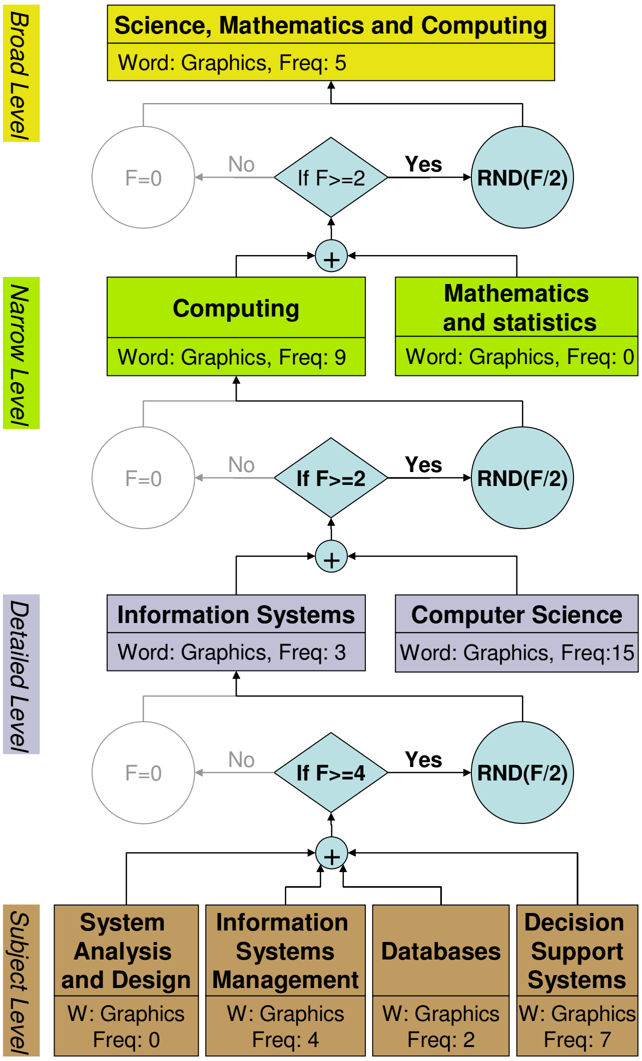

Figure 2. Flow diagram for the creation of word vector for broad, narrow and detailed fields of the hierarchy.

3.4 Post-processing

The task of Post-processing Unit is to store generated meta-data for each individual 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. The results used in the process include:

- syllabus submission date and time, submitting institute, and contact details of submitting person, as extracted from the manifest file by the Pre-processing Unit;
- original syllabus document in a PDF file format as produced by the PDS;
- syllabus segments as produced by the MSS;
- named entities extracted from the header segment of the syllabus as produced by the NEE.
- classification code as assigned to the syllabus document by the Classifier.

A middle-ware is developed to check the database for new confirmed syllabus records once a day and convert them to RDF triples to feed a semantic search engine. The latter engine provides rich semantic search capabilities to the repository's users.

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, were adopted to evaluate the performance of our system's IE:

$$R = \frac{\text{Correct Answers}}{\text{Total Possible Correct}} , \qquad (\mathbf{7})$$

$$P = \frac{\text{Correct Answers}}{\text{Answers Produced}} , \qquad (\mathbf{8})$$

$$F1 = \frac{2\ P\ R}{P + R} . \qquad (\mathbf{9})$$

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

Table 1. Information extraction performance

	P	***R***	***F*1**
Named Entities	0.94	0.74	0.82
Topical Segments	0.84	0.72	0.77

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:

(a) The module name and module code entities in 22 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 3.7% decrease in named entities recall.

(b) 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, corresponding recall decreases. Due to this issue, 211 out of total of 1200 possible named entities and 136 out of total of 1000 possible topical segments were not extracted, resulting in a consequent 17.61% decrease in named entities recall and 13.6% decrease in topical segments recall.

(c) In 53 cases, the named entities were 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 by 5.61%.

(d) 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 by 15.75%.

Table 2 summarizes the adverse affects, in terms of percentage decrease in Precision and Recall, caused by the four issues discussed above.

Table 2. Summary of issues affecting the performance of the information extractor

	1. Absence of a prefix	**2. Prefix not found in thesaurus**	**3. Broken lines in a table cell**	**4. Joining an unidentified segment to an identified one**
Named Entities Precision			-5.61%	
Named Entities Recall	-3.7%	-17.61%		
Topical Segments Precision				-15.75%
Topical Segments Recall		-13.6%		

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

$$P_m = \frac{\text{Total number of correctly classified documents in all classes}}{\text{Total number of classified documents in all classes}} \,. \qquad \textbf{(10)}$$

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. As these results show, the Classifier of our system seems to yield better results in classifying undergraduate courses. 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. For example, a syllabus document could be describing a module which contains both database design and web design components. 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. Also it should be noted here that this level of accuracy is achieved without using any manually classified training documents to train the Classifier.

5 Conclusion and Future Work

This paper presented an objective discussion on 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. It described a prototype syllabus repository system that has recently been developed for the above-mentioned purpose. We described the various components, design and operation of our system in details, particularly our new fully unsupervised syllabus classifier which utilises a web search engine for automatic collection of classification training set to eliminate the need for manual collection and indexing of training set. The rule-based information extraction component for distilling unstructured documents and converting them to structured documents and its sub-components were also described.

As indicated earlier, the main objective of this work was to develop a semi-automated framework for a national syllabus repository system that substantially reduces the amount of manual work required to convert unstructured syllabus documents to fully structured records and store them in the system's database. Within this context, the developed system provides a fully automated meta-data generation process, limiting the administrator's responsibility to the approval of generated meta-data and occasional rectification of a limited number of mistakes that could occur. This approach of deploying automatic information extraction and document classification techniques has been adopted in many similar systems in different domains with significant reported success. This is mainly due to the fact that this approach greatly reduces human effort and involvement without total elimination of manual intervention, which is necessary to minimise the risk of compromising the quality of generated data.

The performance of the system was evaluated using 200 syllabus documents from a number of participating institutes. Presented preliminary experimental results demonstrated good level of accuracy and robustness of the system for its designed purpose. Work is well underway to enhance the accuracy of the system by investigating the following two approaches. Firstly, the addition of table detection and sub-component extraction processes to the information extractor unit of the system, with the target of enhancing the accuracy of information extraction in terms of both precision and recall. Secondly, we are working on improving the accuracy of the classifier via the use of automatic filtration of training documents obtained from the search engine, in order to increase the percentage of highly relevant training documents.

Acknowledgments

This work is funded by the Higher Education Authority (HEA) in Ireland, under their Strategic Innovation Fund programme – Cycle II.

References

Appelt, D.E. & Israel, D. 1999, 'Introduction to Information Extraction Technology', *16th international joint conference on artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, viewed July 2008 <http://www.ai.sri.com/~appelt/ie-tutorial/IJCAI99.pdf >.

Assis, G.d., Laender, A., Gonçalves, M. & Silva, A.d. 2007, 'Exploiting Genre in Focused Crawling', in, *String Processing and Information Retrieval*, Springer Berlin / Heidelberg, pp. 62-73.

Cebeci, Z., Budak, F. & Tekdal, M. 2006, 'Working with XML for Flexible Management of Online Course Syllabi', *Information Technology Journal*, vol. 5, no. 2, pp. 322-328.

Cohen, D.J. 2006, 'From Babel to Knowledge: Data Mining Large Digital Collections', *D-Lib Magazine*, vol. 12, no. 3.

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. 2002, 'GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications', *40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, US., viewed July 2008 <http://gate.ac.uk/sale/acl02/acl-main.pdf>.

DCMI education community 1999, *DC-Ed.*, Dublin Core Metadata Initiative, viewed July 2008 <http://dublincore.org/groups/education/>.

Embley, D.W., Hurst, M., Lopresti, D. & Nagy, G. 2006, 'Table-processing paradigms: a research survey.' *International Journal on Document Analysis and Recognition*, vol. 8, no. 2-3, pp. 66-86.

Ida, M., Nozawa, T., Yoshikane, F., Miyazaki, K. & Kita, H. 2005, 'Syllabus database and Web service on higher education', *7th International Conference on Advanced Communication Technology (IEEE-ICACT 2005)*, vol. Vol. 1, Republic of Korea, pp. 415-418.

IDEAS 2007, *Individualised Digitisied Educational Advisory System*, Enterprise Research Center, University of Limerick, Ireland, viewed July 2008 <http://www.ideas.ie/ >.

IEEE-LTSC-WG12 2002, *The Learning Object Metadata standard*, IEEE Learning Technology Standards Committee, viewed July 2008 <http://www.ieeeltsc.org/working-groups/wg12LOM/>.

ISCED 1997, *International Standard Classification of Education - 1997 version (ISCED97)*, UNESCO, viewed July 2008 <http://www.uis.unesco.org/ev.php?ID=3813_201&ID2=DO_TOPIC>.

JACS 2007, *Joint Academic Coding System v 1.7*, HESA - Higher Education Statistics Agency, UK, viewed July 2008 <http://www.hesa.ac.uk/index.php?option=com_content&task=view&id=158&Itemid=233>.

Joachims, T. 1997, 'A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization', *Fourteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., Nashville, TN, USA, pp. 143-151.

Marcis, J.G. & Carr, D. 2003, 'A note on student views regarding the course syllabus', *Atlantic Economic Journal*, vol. 31, no. 1, pp. 115-115.

Matsunaga, Y., Yamada, S., Ito, E. & Hirokawa, S. 2003, 'A Web Syllabus Crawler and its Efficiency Evaluation', *International Symposium on Information Science and Electrical Engineering 2003 (ISEE 2003)*, Fukuoka, Japan, pp. 565-568.

Mccallum, A. 2005, 'Information extraction: distilling structured data from unstructured text', *Queue*, vol. 3, no. 9, pp. 48-57.

Mitchell, T. 1997, 'Machine Learning', in McGraw-Hill, pp. 180-184.

OpenOffice.org 2.0 2007, sponsored by Sun Microsystems Inc., released under the open source LGPL licence, viewed July 2008 <http://www.openoffice.org/>.

Sebastiani, F. 2002, 'Machine learning in automated text categorization', *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1-47.

Steward, S. 2004, *Pdftk 1.12 - the PDF Toolkit*, sponsored by AccessPDF, Released under the open source GPL licence, viewed July 2008 <http://www.accesspdf.com/pdftk/index.html>.

Thompson, C.A., Smarr, J., Nguyen, H. & Manning, C. 2003, 'Finding Educational Resources on the Web: Exploiting Automatic Extraction of Metadata.' *In ECML Workshop on Adaptive Text Extraction and Mining, Croatia, 2003*, viewed July 2008 <http://nlp.stanford.edu/pubs/edutellaTR.pdf>.

Trewin, D. 2001, *Australian Standard Classification of Education (ASCED)*, Australian Bureau of Statistics, viewed July 2008 <http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/1272.02001?OpenDocument>.

Xiaoyan, Y., Manas, T., Weiguo, F., Manuel, P.-Q., Edward, A.F., William, C., GuoFang, T. & Lillian, C. 2007, 'Automatic syllabus classification', paper presented to the *ACM IEEE Joint Conference on Digital Libraries*, Vancouver, BC, Canada, June.

Xpdf 3.02 2007, Glyph & Cog, LLC., Released under the open source GPL licence, viewed July 2008 <http://www.foolabs.com/xpdf/>.

Yahoo-API 2007, *Yahoo Search Web Services Software Development Kit*, Yahoo! Inc, viewed July 2008 <http://developer.yahoo.com/search/>.

Yu, X., Tungare, M., Fan, W., Yuan, Y., Pérez-Quiñones, M., Fox, E.A., Cameron, W. & Cassel, L. 2007, 'Using Automatic Metadata Extraction to Build a Structured Syllabus Repository', *10th International Conference on Asian Digital Libraries (ICADL 2007)* Ha Noi, Vietnam, viewed July 2008 <http://manas.tungare.name/publications/yu_2007_using>.

Zhu, X. 2005, *Semi-supervised learning literature survey*, Report Number 1530, Department of Computer Sciences, University of Wisconsin, Madison, viewed July 2008 <http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf>.

About the Authors

Arash Joorabchi is a PhD candidate in the Department of Electronic & Computer Engineering, University of Limerick – Ireland. He earned his 1st Class Honours BSc in Computer Science from Griffith College Dublin, Ireland in 2006. His main research areas are automated information extraction and text categorization techniques and their applications in semantic web and digital libraries.

Abdulhussain E. Mahdi is a Senior Lecturer in the Department of Electronic & Computer Engineering, University of Limerick – Ireland. He is a Chartered Engineer (CEng), Member of the Institution of Engineering and Technology - UK (MIET), Member of the Engineering Council - UK, and Founder Member of the International Compumag Society (ICS). Dr Mahdi is a graduate in Electrical Engineering from University of Basrah (BSc 1st Class Hon. 1978) and earned his PhD in Electronic Engineering at University of Wales – Bangor, UK in 1990. He is also a SEDA-UK Accredited Teacher of Higher Education (University of Plymouth, UK 1998). His research interests include: speech/NL processing and applications in telecom, rehabilitation and information extraction, domain transformation and time-frequency analysis. He has authored and co-authored more than 99 refereed journal articles, book chapters and international conference articles, and has edited one book. His published work has been cited in more than 62 journal articles.