Focused Web Crawling for E-Learning Content

Synopsis of the Thesis to be submitted in Partial Fulfillment
of the Requirements for the Award of the Degree of

Master of Technology

In

Computer Science and Engineering

Submitted by:
Udit Sajjanhar (03CS3011)

Under the supervision of

Prof. Pabitra Mitra

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

April 2008

Abstract

The work describes the design of the focused crawler for Intinno, an intelligent web based content management system. Intinno system aims to circumvent the drawbacks of existing learning management systems in terms of scarcity of content which often leads to the cold start problem. The scarcity problem is solved by using a focused crawler to mine educational content from the web. Educational content is mined from University websites in the form of course pages. We present a survey of various probabilistic models such as Hidden Markov Models(HMMs) and Conditional Random Fields(CRFs) for building a focused crawler and finally we describe the design of the system by applying CRFs.

Introduction

Motivation

A Learning Management System (or LMS) is a software tool designed to manage user learning processes [1]. LMSs go far beyond conventional training records management and reporting. The value-add for LMSs is the extensive range of complementary functionality they offer. Learner self-service (e.g. self-registration on instructor-led training), learning workflow (e.g. user notification, teacher approval, waitlist management), the provision of on-line learning, on-line assessment, management of continuous professional education, collaborative learning (e.g. application sharing, discussion threads), and training resource management (e.g. instructors, facilities, equipment), are some of the additional dimensions to leading learning management systems [2].

The current course management systems have a number of drawbacks which hinder their wide acceptance among teachers and students. One of them is the non availability of free content. LMS's assume that the content will be put up by users i.e. teachers and students. This leads to the cold start problem. Instructors who begin to make up a course don't have the material to start up. Materials presented may lack coverage of the subject area and thus fail to cater information needs of all students in a class. On the other hand, students while studying or reading a lecture have to waste a lot of their time in searching for relevant resources from the web.

We aim to build a system which solves the above problem to a large extent. The web interfaced educational digital library will solve the cold start problem faced by instructors. While putting up new course, assignment or a lecture, similar resources would be available from the digital library either by search or by recommendations.

Problem Definition

Web being a rich repository of learning content, we attempt to collect high volume of learning material from web using a web miner [3]. The type of content required for the digital library would include Courses, Assignment, Lectures & Tutorials, Animations & Videos and Quizzes & Questions. This content can be mined from the following sources:

(a) Websites hosting standardized, reviewed and open source course material like MIT Open Courseware, NPTEL India.
(b) Course websites of large international universities. We have considered US universities currently.
(c) Discussion Forums - Google Groups, Yahoo Answers
(d) Websites for animations/videos - Youtube, Google Video and metacafe
(e) Websites for general content - Wikipedia, Mathworld

Out of the above mentioned sources, course websites of different Universities are the richest source of learning content that is authenticated, since it is available on the University site. Also this type of content is the most difficult to mine due its non-structured nature. Crawling the whole university for course pages would be inefficient both in terms of Time and Space required. Hence we need a focused crawling [4] technique to efficiently mine relevant course pages starting from the university homepage.

System Overview

<u>Introduction</u>

It is often observed that University websites are structurally similar to each other. Humans are good at navigating websites to reach specific information within large domain-specific websites. Our system tries to learn the navigation path by observing the user's clicks on as few example searches as possible and then use the learnt model to automatically find the desired pages using as few redundant page fetches as possible. Unlike in focused crawling [4], our goal is not to locate the websites to start with. These are collected from web directories [5] and similar resource websites. We start from a listing of University homepages and after watching the user find the specific information from a few websites in the list, we automate the search in the remaining.

There are two phases to this task: first is the training phase, where the user teaches the system by clicking through pages and labeling a subset with a dynamically defined set of classes, one of them being the Goal class. The classes assigned on intermittent pages along the path can be thought of as "milestones" that capture the structural similarity across websites. At the end of this process, we have a set of classes C and a set of training paths where a subset of the pages in the path are labeled with a class from C. All unlabeled pages before a labeled page are represented with a special prefix state for that label. The system trains a model using the example paths, modeling each class in C as a milestone state. The second phase is the crawling phase where the given list of websites is automatically navigated to find all goal pages.

Formally, we are given a website as a graph $W(V,E)$ consisting of vertex set V and edge set E, where a vertex is a webpage and an edge $e = <u, v>$ is a hyperlink pointing from a webpage u to a webpage v. The goal pages P_G constitute a subset of pages in W reachable from starting seed page P_S. We have to navigate to them starting from P_S visiting fewest possible additional pages. Let $P : P_1, P_2, \ldots, P_n$ be one such path through W from the start page $P_1 = P_S$ to a goal page P_n

$\in P_G$. The ratio of relevant pages visited to the total number of pages visited during the execution is called the **harvest rate**. The objective function is to maximize the harvest rate.

There are two parts to solving this problem.

1. *Recognizing a page as the goal page*. This is a classification problem where given a webpage we have to classify it as being a goal page or not. Often the page alone may not hold enough information to help identify it as the goal page. We will need to consider text around the entire path leading to the goal page in order to decide if it is relevant or not. For example, if we want to get all course pages starting from a university root page, then it is necessary to follow a path through departments' homepages and then through professors' homepage. A course page on its own might be hard to classify.

2. *Foraging for goal pages*. This can be thought as a crawling exercise where, starting from the entry point, we want to visit as few pages as possible in finding the goal pages. This problem is different from the previous work on focused crawling[4] where the goal is to find all web pages relevant to a particular broad topic from the entire web. In our case, we are interested in finding course pages starting from a University homepage. We exploit the regularity in the structures of University websites to build more powerful models than is possible in the case of general-purpose focused crawlers.

Possible Approaches

One possible method of solving the problem is to train a classifier that can discriminate the goal pages from the non-goal pages. Then, extract from the classifier the set of prominent features to serve as keywords to a search engine that indexes all the websites of interest. By restricting the domain to each given starting URL in turn, we issue a keyword search to get a set of candidate pages. We further classify these pages to identify if these are goal pages or not. However this method cannot provide high accuracy for the simple reason that the goal page itself may not hold enough information to correctly identify it as the goal page. The path leading to the goal page is important too.

A Focused crawler must use information gleaned from previously crawled page sequences to estimate the relevance of a newly seen URL. Therefore, good performance depends on powerful modeling of context as well as the current observations. Probabilistic models, such as Hidden Markov Models(HMMs)[6] and Conditional Random Fields(CRFs)[7], can potentially capture both formatting and context. Thus a second approach and the one that we use is to treat this as a sequential labeling problem where we use Hidden Markov Models (HMMs) and the Conditional Random Fields to learn to recognize paths that lead to goal states. We then superimpose ideas from Reinforcement Learning [8] to prioritize the order in which pages should be fetched to reach the goal page. This provides an elegant and unified mechanism of modeling the path learning and foraging problem.

Probabilistic Models

HMMs is one of the most common methods for performing such labeling tasks i.e.to identify the most likely sequence of labels for the words in any given sentence. HMMs are a form of generative model, that defines a joint probability distribution p(X,Y) where X and Y are random variables respectively ranging over observation sequences and their corresponding label sequences. In order to define a joint distribution of this nature, generative models must enumerate all possible observation sequences – a task which, for most domains, is intractable unless observation elements are represented as isolated units, independent from the other elements in an observation sequence. More precisely, the observation element at any given instant in time may only directly depend on the state, or label, at that time. This is an appropriate assumption for a few simple data sets, however most real-world observation sequences are best represented in terms of multiple interacting features and long-range dependencies between observation elements.

This representation issue is one of the most fundamental problems when labeling sequential data. Clearly, a model that supports tractable inference is necessary, however a model that represents the data without making unwarranted independence assumptions is also desirable. One way of satisfying both these criteria is to use a model that defines a conditional probability p(Y |x) over label sequences given a particular observation sequence x, rather than a joint distribution over both label and observation sequences. Conditional models are used to label a novel observation sequence x by selecting the label sequence y that maximizes the conditional probability p(y|x). The conditional nature of such models means that no effort is wasted on modeling the observations, and one is free from having to make unwarranted independence assumptions about these sequences; arbitrary attributes of the observation data may be captured by the model, without the modeler having to worry about how these attributes are related.

Conditional random fields CRFs) are a probabilistic framework for labeling and segmenting sequential data, based on the conditional approach described in the previous paragraph. The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Additionally, CRFs avoid the label bias problem [7], a weakness exhibited by maximum entropy Markov models [9] (MEMMs) and other conditional Markov models based on directed graphical models. CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks [7, 10, 11].

Using CRFs for path classification

CRF models Pr(**y**|**x**) as a Markov random field, with nodes corresponding to elements of the structured object **y**, and potential functions that are conditional on (features of) **x**. In our implementation we use Linear Chain CRF where **y** is a linear sequence of labels from a fixed set *Y*, and the label at position *i* depends only on its previous label.

In our model the labels are: Homepage, Department Listing page, Department Homepage, Faculty Listing Page, Faculty Homepage, Course Page and a Null state. The Null state serves models the pages from where it is not possible to reach the destination page. We model each label as a dual-state - one for the characteristics of the page itself (**page-states**) and the other for the information around links that lead to such a page (**link-states**). Hence, every path alternates between a page-state and a link-state. The features for classification include: Nested Patterns [12] in the html DOM tree indicating that the page contains a list and the presence of pre-defined headings in the page. Nested Patterns are extracted from the DOM trees using suffix tree matching. Headings from the page are extracted using Reinforcement Learning.

The CRF model for the Publications dataset was trained using the mallet [14] toolkit. Training was performed on 122 sequences from 7 university domains. The training dataset included 78 positive and 44 negative sequences and the model was tested on 68 sequences. The test data included some sequences from domains that were not included in the training data. The Precision for the overall states classification was 86.2% and for the goal state classification it was 92.7%.

References

1. *Wikipedia* : http://www.wikipedia.com

2. J. Cole and H. Foster, *Using Moodle: Teaching with the Popular Open Source Course Management System* (O'Reilly Media Inc., 2007).

3. S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data* (Morgan-Kauffman, 2002).

4. S. Chakrabarti, M.H. Van den Berg, and B.E. Dom, "Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery," *Computer Networks,* vol. 31, nos. 11–16, pp. 1623–1640.

5. Yahoo Directory: http://dir.yahoo.com/Education/Higher_Education/Colleges_and_Universities/

6. Lawrence R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2), pages 257–286, February 1989.

7. John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001

8. Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

9. A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In International Conference on Machine Learning, 2000.

10. D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random fields. Proceedings of the ACM SIGIR, 2003.

11. F. Sha and F. Pereira. Shallow parsing with conditional random fields. Proceedings of Human Language Technology, NAACL 2003, 2003.

12. J. Wang and F.H. Lochovsky, “Wrapper Induction Based on Nested Pattern Discovery,” Technical Report HKUST-CS-27-02, Dept. of Computer Science, Hong Kong, Univ. of Science & Technology, 2002.

13. V.G.Vinod Vydiswaran and Sunita Sarawagi, *Learning to extract information from large websites using sequential models*(In COMAD, 2005. SIGKDD Explorations. Volume 6, Issue 2 - Page 66)

14. McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

15. Hongyu Liu , Evangelos Milios , Jeannette Janssen, Probabilistic models for focused web crawling, Proceedings of the 6th annual ACM international workshop on Web information and data management, November 12-13, 2004, Washington DC, USA