A Generalized Knowledge Representation Architecture for Intelligent Tutoring System

A Thesis submitted in Partial Fulfillment of the Requirements for the Award of the Degree of

Master of Technology

In

Computer Science and Engineering

Submitted By:

Mayank Jain (03CS3010)

Under the supervision of

Prof. Pabitra Mitra

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

May 2008

Certificate

This is to certify that the thesis titled **A Generalized Knowledge Representation Architecture for Intelligent Tutoring System**, submitted by **Mayank Jain**, to the Department of Computer Science and Engineering, in partial fulfillment for the award of the degree of **Master of Technology** is a bonafide record of work carried out by him under our supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of the institute and, in our opinion, has reached the standard needed for submission.

Prof. Pabitra Mitra

Dept. of Computer Science and Engineering
Indian Institute of Technology
Kharagpur 721302, INDIA

Mayank Jain

Dept. of Computer Science and Engineering
Indian Institute of Technology
Kharagpur 721302, INDIA

Acknowledgement

I take this opportunity to express my deep sense of gratitude to my guide Prof. Pabitra Mitra for his guidance, support and inspiration throughout the duration of the work. Also I would like to specially acknowledge the help and encouragement I received from the my project partners Arpit Jain and Udit Sajjanhar. Without them, it would not have been possible to complete this work.

I would like to thank all the faculty members, staffs and research scholars of the Department of Computer Science and Engineering and my friends specially Piyush, Joy deep, Sankalp, Umang, Prithvi, Lalit for providing me adequate moral support and encouragement which kept me motivated.

Finally, I would like to thank my family members for allowing me to concentrate in my work. This work would not have been possible but for their constant support and encouragement.

Mayank Jain
Dept. of Computer Science and Engineering
Indian Institute of Technology,
Kharagpur, 721302

ABSTRACT

An ***intelligent tutoring system*** *(ITS) is a system that provides direct customized instruction or feedback to students without the intervention of human beings. Most of the research in ITS has been in evaluation, restructuring of the learning objects, and personalization of the learning object navigation path. With the explosion of content on the World Wide Web(WWW), the scope of application of Data Mining to E-Learning applications has increased tremendously. In this work, we identify a set of applications which go one step ahead from ITS and use the WWW to aid the learning process of the "learning object content". Each application has a high level of coupling with the knowledge representation model, which models the resources stored in the Digital Library. This domain model must be complete and accurate otherwise the learning system will not perform effectively.*

This research presents the architecture for simplifying and automating the process of creating the domain model for an intelligent e-learning system. In this work we present an algorithm to create a ***knowledge representation*** *of educational resources using the World Wide Web. We outline the advantages and limitations of this general architecture, and describe its implementation in Intinno – a developing Intelligent Learning Management System.*

The work is implemented for intinno, an intelligent web based learning content management system. The Intinno system aims to circumvent certain drawbacks of existing learning management systems in terms of scarcity of content, lack of intelligent search and context sensitive personalization. The system is deployed at http://www.intinno.com

Table of Contents

Chapter 1: Introduction

1.1 E-Learning Introduction

1.1.1 Learning Content Management System

A Learning Management System (or LMS) is a software tool designed to manage user learning processes. 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 [1].

In addition to managing the administrative functions of online learning, some systems also provide tools to deliver and manage instructor-led synchronous and asynchronous online teaching based on learning object methodology. These systems are called Learning content management systems or LCMSs. An LCMS provides tools for authoring and re-using or re-purposing content as well as virtual spaces for learner interaction (such as discussion forums and live chat rooms). The focus of an LCMS is on learning content. It gives authors, instructional designers, and subject matter experts the means to create and re-use e-learning content more efficiently [2]. LCMSs provide instructors with the ability to perform the following tasks [3]:

- Place course materials online. Most CMSs provide pre-programmed buttons for the course syllabus, course schedule, and course materials linked to specific lessons, such as copies of readings and PowerPoint slides from lectures.
- Track student progress through assessment features, which enable instructors to give quizzes and tests online, and an online gradebook, where instructors can post student grades.
- Discussion board, where instructors and students can discuss readings and continue class discussions between formal class sessions.

- Other communications tools, which let instructors send announcements to classes and communicate individually with students.
- Lock box for students, where students can store class materials in a safe place—either a presentation to give later in class or backing up class assignments in a safe place.
- Course statistics, which provide information on the use of the course site, including who used the course site and when.

LCMSs also have proven popular in managing asynchronous academic distance courses, too, because of their ability to manage discussions. In addition, given that LCMSs were already installed and in wide use only adds to their popularity. When using a LCMS to manage a distance course, instructors post a core lessons master script, of sorts, that guides students through readings, discussions, and learning activities instead of merely posting readings and PowerPoint slides for each lesson,. Instructors then use the discussion board to manage the course discussions, which are usually more extensive than those used in classroom courses.

1.1.1.1 Problems faced by LCMS

The current course management systems have a number of drawbacks which hinder their wide acceptance among teachers and students.

- One of them being the problem of cold start. Instructors who begin to make up a course don't have the material to start up.
- Seamless content reuse is often not possible.
- Materials presented may lack coverage of the subject area and thus fail to cater information needs of all students in a class.
- Students while studying or reading a lecture have to waste a lot of their time in searching for relevant resources from the web.

1.1.1.2 Advantages of LCMS

The current course management systems have a number of advantages:

- LCMSs enable instructors to easily create a course website by following a template and uploading existing documents in PowerPoint, Word, Excel, Acrobat and other popular formats without converting them to a web format (like HTML), they require few specialized skills.
- LCMSs are easy to learn and were quickly adopted by instructors, even those who might claim to be luddites.
- LCMSs allow active participation of students in learning activities even outside the physical boundaries of a classroom.
- LCMSs help the instructors to create and archive of the course material and discussions which would be helpful to the students opting the course next year.

1.1.2 LCMS + Web 2.0 = E-Learning2.0

The changes in e-learning are being driven by two primary forces [4]. The first force is a steady increase in the pace information creation, boosted by the availability of easy to use LCMS. This has led to a shift in work, especially knowledge work, and an evolution in information needs. The second driver affecting workplace learning is the advent of Web 2.0.

In its most basic form, Web 2.0 means that anyone should be able to easily create and contribute content on the Internet. This ranges from writing a blog, to providing video on YouTube, to putting pictures on Flickr, to contributing written content on wikis such as Wikipedia, as well as developing a social network on something like MySpace. The key components to Web 2.0 are the ease of using the tools and the collaboration/social interaction that naturally results. One of the interesting results from Web 2.0 is something called collective intelligence. For example, consider how Amazon's user ratings and comments influence buyer behavior.

The term *E-Learning 2.0* was coined by Stephen Downes, a Canadian researcher, and it derives from the overall e-learning trends stated above in combination with Web 2.0. To begin to examine E-Learning 2.0, let's consider an example:

A small team of five practitioners in a corporate learning department has adopted e-learning 2.0 tools as part of their daily work. They need to define their strategy around the use of "rapid e-learning" and present it to management as part of the annual budget process. Here are some of the ways the workgroup will take advantage of E-Learning 2.0 tools:

- Search for useful web pages, then tag, add comments, and share them by using such social bookmarking tools as del.icio.us or Yahoo MyWeb. By using these tools, the team will keep a copy of each page; the page is full-text searchable; it can be accessed from any computer; and everyone on the team has access to the same links.
- Create public blog posts (using a tool like Blogger) that will outline the team's current thinking about how rapid e-learning fits into its future strategic plans. The blog also will solicit feedback from everyone on the team, as well as the larger e-learning blog community.
- Write or copy-and-paste notes into a wiki, which will become a shared resource that everyone on the team can edit.
- Use an RSS reader (for example, Bloglines) to track updates to the wiki, social bookmarking tools, and the blog. This eliminates the need for email as the reader becomes the single place each team member visits to see whatâ€™s happened recently.

E-Learning 2.0 is making an impact in formal learning settings, and they are particularly useful for collaborative formal learning. For example, wikis can be used as part of group projects; blogs can be used to submit written work and offer the opportunity for peers to provide feedback in a collaborative learning setting; and social bookmarking tools can be used as part of collaborative research. Again, the ease-of-use and collaborative nature of these tools make them a natural fit for learning.

1.1.3 Intelligent tutoring Systems (ITS)

1.1.3.1 Introduction

Imagine that each learner in a classroom has a personal training assistant who pays attention to the participant's learning needs, assesses and diagnoses problems, and provides assistance as needed. The assistant could perform many of the routine instructional interventions and alert the instructor of learning problems that are too difficult for it. By taking on basic assistance tasks, the instructor would be free to concentrate on training issues that require greater expertise.

Providing a personal training assistant for each learner is beyond the training budgets of most organizations. However, a *virtual* training assistant that captures the subject matter and teaching expertise of experienced trainers provides a captivating new option. The concept, known as intelligent tutoring systems (ITS) [5] or intelligent computer-aided instruction (ICAI), has been pursued for more than three decades by researchers in education, psychology, and artificial intelligence. Today, prototype and operational ITS systems provide practice-based instruction to support corporate training, schools and college education, and military training.

The goal of ITS is to provide the benefits of one-on-one instruction automatically and cost effectively. Like training simulations, ITS enables participants to practice their skills by carrying out tasks within highly interactive learning environments. However, ITS goes beyond training simulations by answering user questions and providing individualized guidance. Unlike other computer-based training technologies, ITS systems assess each learner's actions within these interactive environments and develop a model of their knowledge, skills, and expertise. Based on the learner model, ITSs tailor instructional strategies, in terms of both the content and style, and provide explanations, hints, examples, demonstrations, and practice problems as needed.

ITS systems typically rely on three types of knowledge, organized into separate software modules (as shown in Figure 1). The "expert model" represents subject matter expertise and provides the ITS with knowledge of what it's teaching. The "student model" represents what the user does and doesn't know, and what he or she does and doesn't have. This knowledge lets the ITS know who it's teaching. The "instructor model" enables the ITS to know how to teach, by encoding instructional strategies used via the tutoring system user interface.

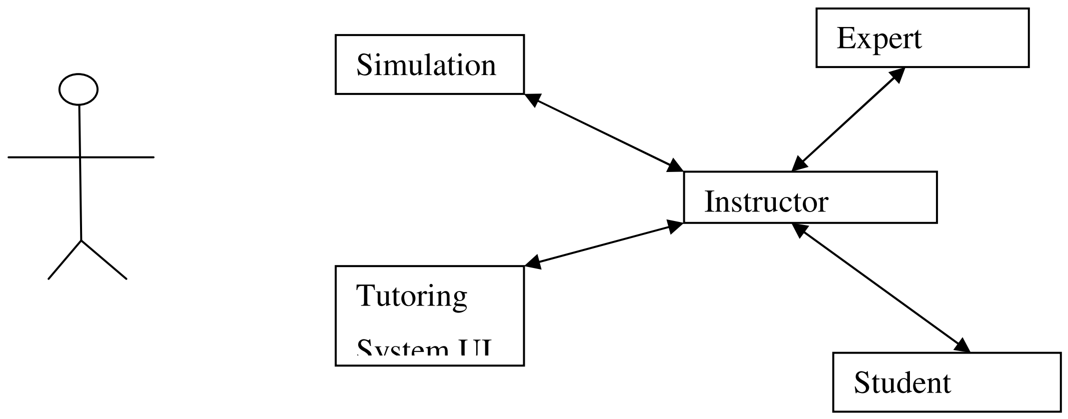


Figure 1: Components of an intelligent Tutoring System

1.1.3.2 Disadvantages of ITS

ITS needs careful preparation in terms of describing the knowledge and possible behaviors of experts, students and tutors. This description needs to be done in a formal language in order that the ITS may process the information and draw inferences in order to generate feedback or instruction. Therefore a mere description is not enough, the knowledge contained in the models should be organized and linked to an inference engine. It is through the latter's interaction with the descriptive data that tutorial feedback is generated. All this is a substantial amount of work. This means that building an ITS is an option only in situations in which they, in spite of their relatively high development costs, still reduce the overall costs through reducing need for human instructors or sufficiently boosting overall productivity. Such situations occur when large groups need to be tutored simultaneously or many replicated tutoring efforts are needed. Cases in point are technical training situations such as training of military recruits and high school mathematics.

1.1.3.3 Advantages of ITS

An ITS system has the following advantages:

- They provide the benefits of one-on-one instruction automatically in a cost effective manner.
- ITS-taught students generally learn faster and translate the learning into improved performance better than classroom-trained participants.
- Provides direct feedback to the students without the intervention of human beings.

1.2 Motivation

LCMS and ITS systems are two ends of same rope. Both the systems have the basic goals of making the learning process efficient for the students and reducing the work required to be done by the teacher. However both the systems differ in the way they go about achieving the basic goals.

LCMS provide a platform where it is easier for the teacher to upload content, students have a central place for all their learning materials and the discussion/questions are extended out from the physical boundaries of the classroom. LCMS systems are easier to build and such systems involve active participation from the student community. However LCMS suffer from the problem that they have no intelligence built into them.

ITS systems are intelligent. They model how a teacher would teach in the class and also keep a track of the student's performance. Such systems use the record of students' performance to enhance their learning process. However these systems are expensive to build and model. They require much human expertise and are domain specific.

Thus we see that neither LCMS nor ITS system is a one stop solution to making the learning process efficient. ITS systems are difficult to build and LCMS aren't intelligent enough. An intelligent application would be the one that would combine the benefits of both the systems into one, i.e. it is as easy a CMS for the use of the teacher (eliminating the task of annotating the sources), involves high participation from the user and is also intelligent enough to make the user learning process easier and efficient.

1.3 Our Approach

We propose a LMS, *Intinno*, motivated from the above discussion. Intinno being an LMS would be easy to build and would have all the benefits of an LCMS. To make Intinno intelligent we propose the addition of intelligent applications to the system. These applications would be directly integrated in the LCMS and would use Data-mining techniques [6] to make the learning process efficient. Till date there has been no work on integrating intelligent application into an LCMS that enhance the students learning process. However there have been several attempts to develop stand alone intelligent applications.

The work in [7] focuses on helping a teacher moderate a classroom of students using e-discussion tools in which the students comprise multiple discussion groups. Generally a teacher can bring to bear his or her experience and moderation expertise to steer the discussions when problems occur and provide encouragement when discussions are productive. However when multiple e-discussions occur simultaneously, a single teacher may struggle to follow all of the discussions. To direct the teacher's attention to the 'hot spots,' this paper proposes software tools that pre-process, aggregate, and summarize the incoming flood of data. [8] Proposes Information Retrieval techniques to detect conflicts within the same exam. A Conflict exists in an exam if at least two questions within that exam are redundant in content, and/or if at least one question reveals the answer to another question within the same exam. However none of these mentioned application aim to enhance the student's learning process.

1.4 Applications

From the student's point of view, an application that abstracts the learning material and presents it in way that is easier to grasp would be highly useful. Such an application would help the student to learn the whole concept by learning small related set of concepts. Proposed set of such applications are:

- Memory Maps: This application would extract a set of keywords from the document and present a graph of these keywords connected by a set of associations. It would be similar to making an automatic memory map of the concept to be learned.

- Presentation Module: This module would abstract a given page in the form of a power point presentation. This application would help the users to grasp a few important aspects of the concepts to be learned. It would also prove as the starting for preparing presentations from a given piece of content.

With the explosion of Web2.0 the content on the web has increased manifold. This has provided the user with treasures of information. However there is a catch here. Since the number of sources that present the same content to the user is large, a large portion of the user's time is spent in searching for appropriate material on the web. The same logic applies to students. Also there exist no educational search engines that focus only on educational content. Applications that can be developed to cater to the above problems:

- Recommendation Engine: This engine would implicitly help the user in getting similar content. When a student is reading a lecture then similar content would be automatically recommended to him. The recommendation would be content diversified, i.e. if the person is currently studying Lectures then he/she will be recommended questions/quizzes/Assignments. however on the other hand if the person is bust doing Assignments or solving questions then he/she will be recommended lectures/tutorials on that particular subject from the digital library. Recommendation will be personalized i.e it will be based on the courses that the user has done and also on his/her level of understanding which can be judged from his courses list.

- Educational Search Engine: This will provide two additional capabilities in addition to keyword based search, namely (i) Content based search for similar courses, and (ii) Intelligent search for course materials (i.e. if a search is given for material on biochemistry course then materials from molecular biology course should also turn up in the results.

Until now the applications mentioned above have focused only on improving the efficiency of learning from the student's point of view. However an intelligent LCMS should minimize the efforts required from an instructor. Thus the following applications are proposed:

- Automatic Evaluator for Coding Assignments: In large number of courses the assignments given to students require a coded solution. The final answer is same for every student. This evaluator will automatically check the answers minimizing the efforts required from the instructors.

- Automatic Question Answering: One of the major advantages of LCMS is the discussion forums. However it may so happen that a question/query asked by a particular student may have been answered before in some other course. This module will seek out such answers and will present them to the users, automatically.

- Duplicate Detection in Assignments: This application would detect duplication in submitted assignments by the use of text matching algorithms. The instructor would be provided of the percentage match between any two submitted assignments.

The current LCMSs have the drawback of 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. Our system 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.

1.4 Organization

This thesis is organized as follows: - In chapter 2, we give the literature survey and describe some of the related work in Document Representation. We explain architectures used by different intelligent tutoring systems to represent data to enable intelligent applications being built over it. We also give the literature survey of the work in manual, (semi-)automatic and automatic Ontology Generation. In Chapter 3, we introduce a new architecture for document representation which has been developed as part of this work. In chapter 4, we propose a new architecture for ontology generation. In chapter 5, we give the details about our implementation and present the results. We finally conclude in chapter 6.

Chapter 2: Literature Survey

2.1 Related Work in Document Representation

2.1.1 Introduction

There are representation techniques such as frames, rules and semantic networks which have originated from theories of human information processing. Since knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to represent knowledge in a manner as to facilitate inferences (i.e. drawing conclusions) from knowledge. Problem Solving can be simplified by an appropriate choice of *knowledge representation (KR)*. Representing knowledge in some ways makes certain problems easier to solve.

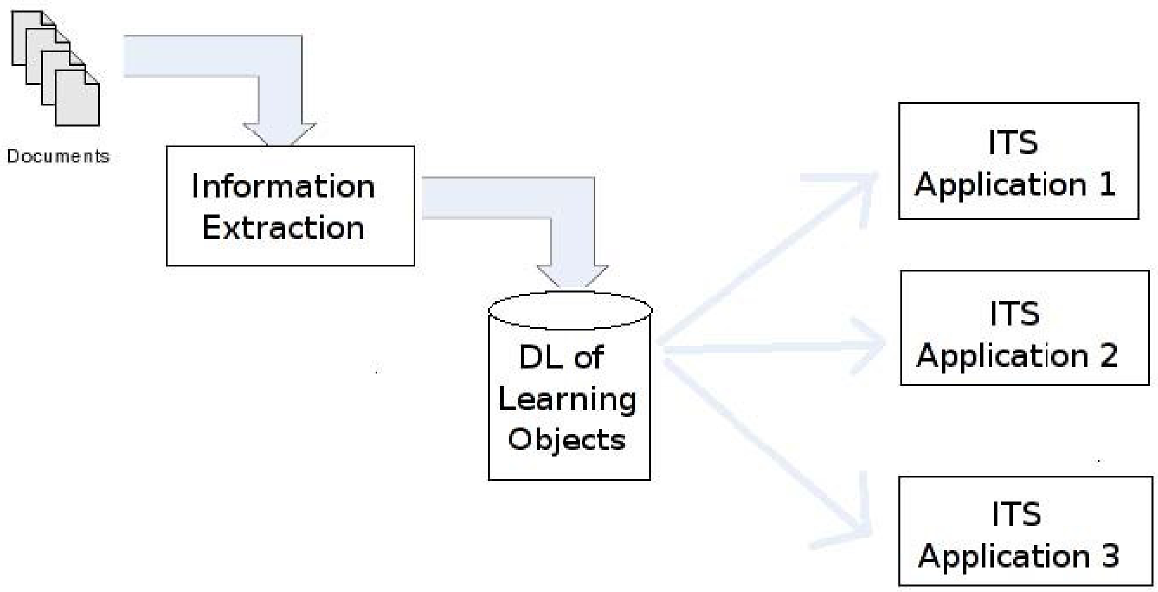


Figure 2: Applications of Knowledge Representaion

KR is most commonly used to refer to representations intended for processing by computers, and in particular, for representations consisting of explicit objects (the class of all humans, or Ram a certain individual), and of assertions or claims about them ('Ram is a human', or 'all humans have one head'). Representing knowledge in such explicit form enables computers to draw conclusions from knowledge already stored ('Ram has one head').

2.1.2 Motivation

The knowledge is crucially important in the development of an intelligent tutoring system for e-learning. For this work, we assume that we have a repository of educational documents mined from the web. The content described above can be mined from the following major resources

(i) MIT Open Courseware, NPTEL India
(ii) .edu domain
(iii) Discussion Forums -Google Groups, Yahoo Answers
(iv) YouTube, Google Video and Metacafe
(v) Wikipedia, MathWorld
(vi) Company Websites for product related info and case studies
(vii) Domain specific websites for questions, tutorials etc.

Open repositories like Wikipedia and information pages authored as blogs etc:- by casual users if used efficiently can be a very good resource for learning. All this knowledge needs to be represented efficiently for use by e-learning systems.

The goal of this work is to explore approaches for representation of knowledge for efficient use of resources for an intelligent learning system. We intend to find an approach which can help in capturing the semantics of the crawled resources and efficiently implement a set of learning applications. Hence the final goal is to have a knowledge representation technique specially designed to support intelligent tutoring applications like automatic annotation of text and construction of memory maps.

2.1.3 Existing Approaches

This section presents a literature survey of the innovative approaches for performing data mining on documents, which serves as a basis for knowledge extraction in e-learning environments.

2.1.3.1 DIG Representation

The work by Hammouda and Kamel [9] presents an innovative approach for performing data mining on documents, which serves as a basis for knowledge extraction in e-learning environments. The approach is based on a radical model of text data that considers phrasal features paramount in documents, and employs graph theory to facilitate phrase representation and efficient matching. In the process of text mining, a grouping (clustering) approach is also employed to identify groups of documents such that each group represents a different topic in the underlying document collection. Document groups are tagged with topic labels through unsupervised key phrase extraction from the document clusters.

The model presented by Hammouda and Kamel [9] for document representation is called the Document Index Graph (DIG). This model indexes the documents while maintaining the sentence structure in the original documents. This allows use of more informative phrase matching rather than individual words matching. Moreover, DIG also captures the different levels of significance of the original sentences, thus allowing us to make use of sentence significance. Suffix trees are the closest structure to the proposed model, but they suffer from huge redundancy [10].

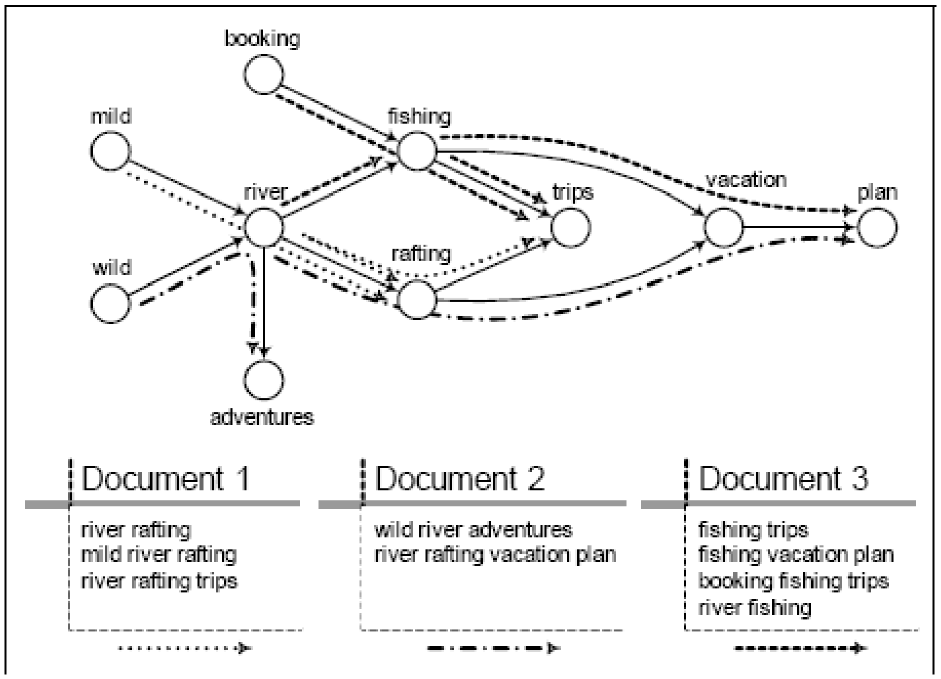


Figure 3: Example of the Document Index Graph

The DIG is built incrementally by processing one document at a time. When a new document is introduced, it is scanned in sequential fashion, and the graph is updated with the new sentence information as necessary. New words are added to the graph as necessary and connected with other nodes to reflect the sentence structure.

Upon introducing a new document, finding matching phrases from previously seen documents becomes an easy task using DIG. This is done by incremental graph building and phrase matching. The approach serves in solving some of the difficult problems in e-learning where the volume of data could be overwhelming for the learner; such as automatically organizing documents and articles based on topics, and providing summaries for documents and groups of documents.

2.1.3.2 Document maps - WebSOM Model

With the WEBSOM [11] method a textual document collection may be organized onto a graphical map display that provides an overview of the collection and facilitates interactive browsing. Interesting documents can be located on the map using a content-directed search. The documents are organized using the SOM algorithm [12] onto a document map. A graphical display of the map provides a general overview of the information contained in the document collection. Each document is encoded by locating the categories of its words on the word category map. The histogram of the hits on the word category map is updated at the location of the word, and finally the histogram is normalized. The *document map* is formed with the SOM algorithm using the histograms as fingerprints of the documents.

The WEBSOM appears to be especially suitable for exploration tasks in which the user either does not know the domain very well or has only a vague idea of the contents of the full-text database being examined. With the WEBSOM, the documents are ordered meaningfully on a document map according to their contents. It can be used as a tool especially in exploring a document collection but also in searching and filtering tasks. The texts used in experiments contained texts of 85 newsgroups containing over one million documents.

2.1.3.3 Three level Hierarchical Representation

In this work [13] [14] [15], an ontological structure has been proposed to represent the domain knowledge. The knowledge representation database or ontology is organized into a three level hierarchical structure namely as in Figure 4.

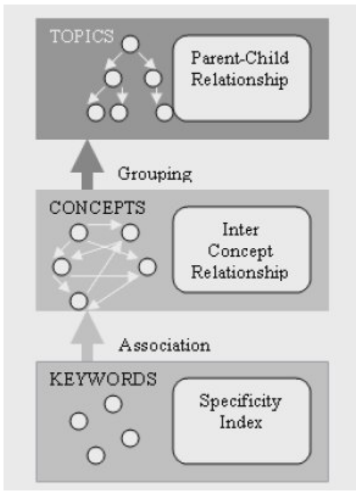


Figure 4: Three level Hierarchical Representation

The top most level contains generalized representative entities from the domain and forms a hierarchy in between them. On the topic level, the topics share a parent child relationship. This provides a way of generalization from a specific to a more general topic. The hierarchy of the topics is stored as an n-ary tree with the exception that a node may have multiple parents. The documents on a topic may contain several concepts. The next level contains more specific terms related to the domain and are connected to each other by some domain dependent relationships defined by the domain experts. The concepts are categorized into the topics by means of a grouping function. The bottom most level contains the raw terms or keywords occurring in the real world. The keywords are associated with concepts by defining some association values. This work also defines a set of 11 relations to cover most types of relations between two concepts in the domain of physics, biology and geography.

2.2 *Related Work in Ontology Generation*

2.2.1 Introduction

One of the hottest R&D topics in recent years in the AI community, as well as in the Internet community, is *the Semantic Web*. It is about making the Web more understandable by machines [16]. It is also about building an appropriate infrastructure for intelligent agents to run around the Web performing complex actions for their users [17]. In order to do that, agents must retrieve and manipulate pertinent information, which requires seamless agent integration with the Web and taking full advantage of the existing infrastructure (such as message sending, security, authentication, directory services, and application service frameworks) [18]. Furthermore, Semantic Web is about explicitly declaring the knowledge embedded in many Web-based applications, integrating information in an intelligent way, providing semantic-based access to the Internet, and extracting information from texts [19]. Ultimately, Semantic Web is about how to implement reliable, large-scale interoperation of Web services, to make such services computer interpretable to create a Web of machine-understandable and interoperable services that intelligent agents can discover, execute, and compose automatically [20].

The problem is that the Web is huge, but not smart enough to easily integrate all of those numerous pieces of information from the Web that a user really needs. Such integration at a high, User-oriented level is desirable in nearly all uses of the Web. Today, most Web information is represented in natural-language; however, our computers cannot understand and interpret its meaning. Humans themselves can process only a tiny fraction of information available on the Web, and would benefit enormously if they could turn to machines for help in processing and analyzing the Web contents [21]. Unfortunately, the Web was built for human consumption, not for machine consumption - although everything on the Web is *machine-readable*, it is not *machine-understandable* [22]. We need the Semantic Web to express information in a precise, machine-interpretable form, ready for software agents to process, share, and reuse it, as well as to understand what the terms describing the data mean. That would enable Web-based applications to interoperate both on the syntactic and semantic level. The explicit representation of the semantics of data, accompanied with domain theories (that is, ontology), will enable a Web that provides a qualitatively new level of service - for example, intelligent search engines, information brokers, and information filters [23].

There is research on important issues related to the development of the Semantic Web, and their implications for Web-based teaching and learning [24]. It describes what it means precisely to create, to find, and to use educational resources on the Semantic Web pages, as opposed to doing it on today's Web. Our work presents the background and context for activities of developing Semantic

Web-based educational systems, indicates some existing applications and tools, and introduces some applications which can enhance learning using the semantics.

2.2.2 Motivation

In recent years, web-based learning has become a new means for learners to learn without time and distance barriers. Although information technologies enable learners to access large amounts of learning materials across geographical boundaries, this proliferation has led to a serious problem, called information overload. If the learners come across scrappy and fragmentary data, they can easily experience learning disorientation and find themselves being unable to construct complete and systematic domain knowledge. Therefore, this research proposes a solution to these problems from an ontology perspective.

Ontology is a description of the concepts and relationships for the purpose of enabling knowledge sharing and reuse [25]. Many instructors use ontology, created by domain experts, to reduce information overload and learning disorientation. Besides, instructors also use ontology to provide adaptive teaching materials and design adaptive learning path to guide learners. However, construction of ontology requires consensus among domain experts. Therefore, it is a rather tough and time consuming task. The purpose of this research is to use concept maps which can be constructed automatically to represent ontology.

Also e-Learning is a newly growing field. In such a new domain, ontology would be out of date if domain experts try to construct it manually over months if not years. To cope with the changing nature of e-Learning domain, we need to develop a mining technique to construct e-Learning ontology automatically. As e-Learning domain development will mature and get stabilized, the concept maps drawn by our system could then resemble the real domain ontology.

Recent research has demonstrated the important of ontology and its applications. For example, while designing adaptive learning materials, designers need to refer to the ontology of a subject domain. Moreover, ontology can show the whole picture and core knowledge about a subject domain. Research from literature also suggested that graphical representation of ontology can reduce the problems of information overload and learning disorientation for learners. However, ontology constructions used to rely on domain experts in the past; it is a time consuming and high cost task. Ontology creation for emerging new domains like e-Learning is even more challenging. The aim of this paper is to construct e-Learning domain concept maps, an alternative form of ontology, from academic articles. We adopt some relevant journal articles and conferences papers in e-Learning domain as data sources, and apply text-mining techniques to automatically construct concept maps for e-Learning domain. The constructed concept maps can provide a useful reference for researchers, who are new to e-Leaning field, to study related issues, for teachers to design adaptive courses, and for learners to understand the whole picture of e-Learning domain knowledge.

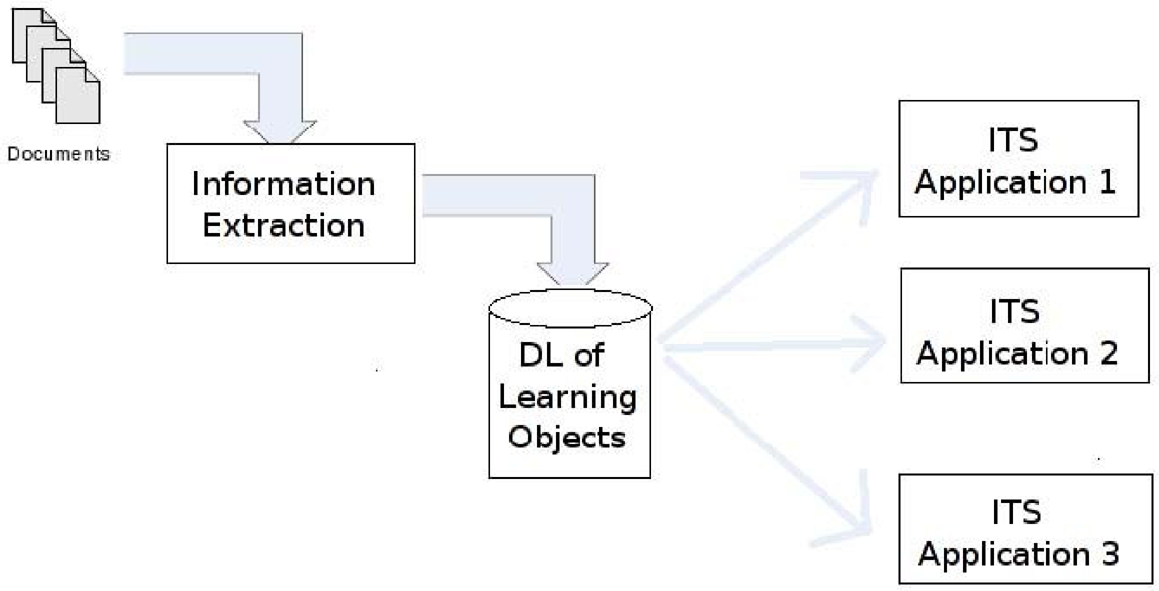


Figure 5:Applications of Knowledge Representation

2.2.3 Existing Approaches

Researchers and practitioners in the fields of databases and information integration have produced a large body of research to facilitate interoperability between different systems. This research ranges from techniques for matching database schemas to answering queries using multiple sources of data. Ontology research is another discipline that deals with semantic heterogeneity in structured data.

The survey paper by Natalya [26] discusses the major thrusts of approaches to semantic integration produced by various projects in the ontology community and to provide readers with pointers to sources for additional information. We will focus on the approaches that highlight the use of ontology, their emphasis on knowledge sharing, and their use in reasoning for the applications to be developed for Intinno System. There are many definitions of what ontology are [27], the common thread in these definitions is that ontology is some formal description of a domain of discourse, intended for sharing among different applications, and expressed in a language that can be used for reasoning. These features of ontologies underscore the main trends that distinguish semantic-integration research in the ontology community: First, since the underlying goal of ontology development is to create artifacts that different applications can share, there is an emphasis on creating common ontologies that can then be extended for more specific domains and applications.

If these extensions refer to the same top-level ontology, the problem of integrating them can be greatly alleviated. Second, since ontologies are developed for use with reasoning engines and semantics of ontology languages are specified with reasoning in mind, inference and reasoning takes center stage in ontology-integration approaches. Ontologies have gained popularity in the AI community as a means for establishing explicit formal vocabulary to share between applications.

The main approaches used to build ontologies are:

(i) Manual

(ii) Automatic

(iii)Semi-automatic

In the past, most concept maps were built by domain experts, and scholars used them in related researches [28,29]. The process of building concept maps has been largely manual. Even at present, although various computer-based tools are available that can help the concept map building process by providing various visual editing functions, the actual data input still relies solely on human experts' contribution.

Recently lot of research has been published to make this process of ontology creation automatic. This work [30] presents a process for constructing the concept map automatically for journal articles. The are four main steps in the procedure presented are: article information retrieval, concept item extraction, research keyword indexing and calculation of "relation strength". Finally, users can use query parameters to obtain concept maps through the friendly user interface. Figure 6 illustrates the procedure for constructing the concept map.

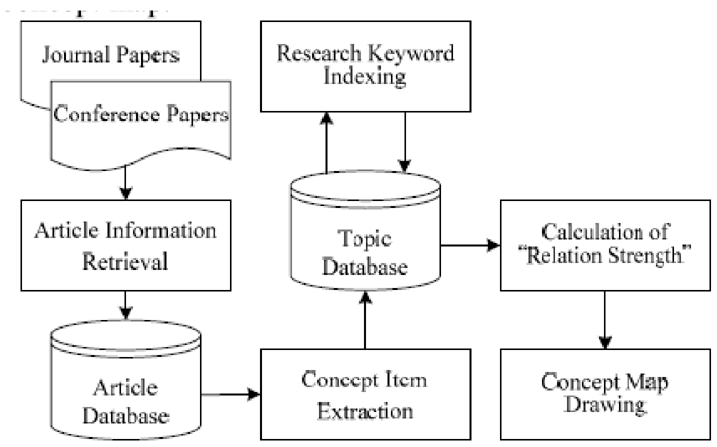


Figure6: procedure for constructing the concept map

2.2.4 Existing Ontologies

The Web Ontology Language (OWL) [31] is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium. This family of languages is based on two (largely, but not entirely, compatible) semantics: OWL-DL and OWL-Lite semantics are based on Description Logics, which have attractive and well-understood computational properties, while OWL-Full uses a novel semantic model intended to provide compatibility with RDF Schema. OWL ontologies are most commonly serialized using RDF/XML syntax. OWL is considered one of the fundamental technologies underpinning the Semantic Web, and has attracted both academic and commercial interest.

There is a library of ontologies which were developed either here at Stanford or by its user community which can be browsed at [32]. The Protégé-OWL editor developed at Stanford enables users to:

- Load and save OWL and RDF ontologies.
- Edit and visualize classes, properties, and SWRL rules.
- Define logical class characteristics as OWL expressions.
- Execute reasoners such as description logic classifiers.
- Edit OWL individuals for Semantic Web markup.

Chapter 3: Document Representation Architecture

3.1 Introduction

The archived content crawled by the focused crawler is used to develop concept maps that capture the various semantic relations among data. The semantic content from the digital library is then used to develop intelligent learning applications. These applications focus on making the learning process for a student both efficient and effective.

The knowledge representation is crucially important in the development of an intelligent learning system for e-learning. In order to use the document corpus effectively and efficiently, not only the contents but also the representation of contained knowledge is important. The effectiveness of the learning applications like memory maps will depend significantly on the knowledge representation architecture. The content mined from the web can be divided into two categories in terms of its usability for an e-learning system.

(i) Structured content (courses)

(ii) Non-Structured educational content.

Structured course content crawled in section 3 is annotated before it is archived in the digital library. We identify a set of tags which are required to represent the course in SCORM [33] standard. We extract information from the structured courses to convert them into (semi-) SCORM format. In addition to the above tags we also store some entity specific meta-tags that are important from the point of view of indexing and parameterized search. For both the above category of tags hand crafted wrappers are used for information extraction [34]. Adding the search keywords in the meta-tags ensures that information about related course/course material is added in the tags of the entity. This will ensure that if the search is made in the name of the course then related material also turns up in the results.

Non-Structured content is available in abundance on the web. Open repositories like Wikipedia and information pages authored as blogs etc:- by casual users if used efficiently can be a very good resource for learning. All this knowledge needs to be represented efficiently for use by e-learning systems. In our work, we report the existing approaches. We also propose a new approach for automatic construction of concept maps from the content mined from the web. Our knowledge representation technique is specially designed to support intelligent tutoring applications like automatic annotation of text and construction of memory maps. We have designed and implemented a heuristic based algorithm to extract the headings from web documents.

3.2 System Design and Methodology

The document representation module consists of the three main modules namely:

- Tree Building Phase 1
- Tree Building Phase 2
- Key-phrase Resolution

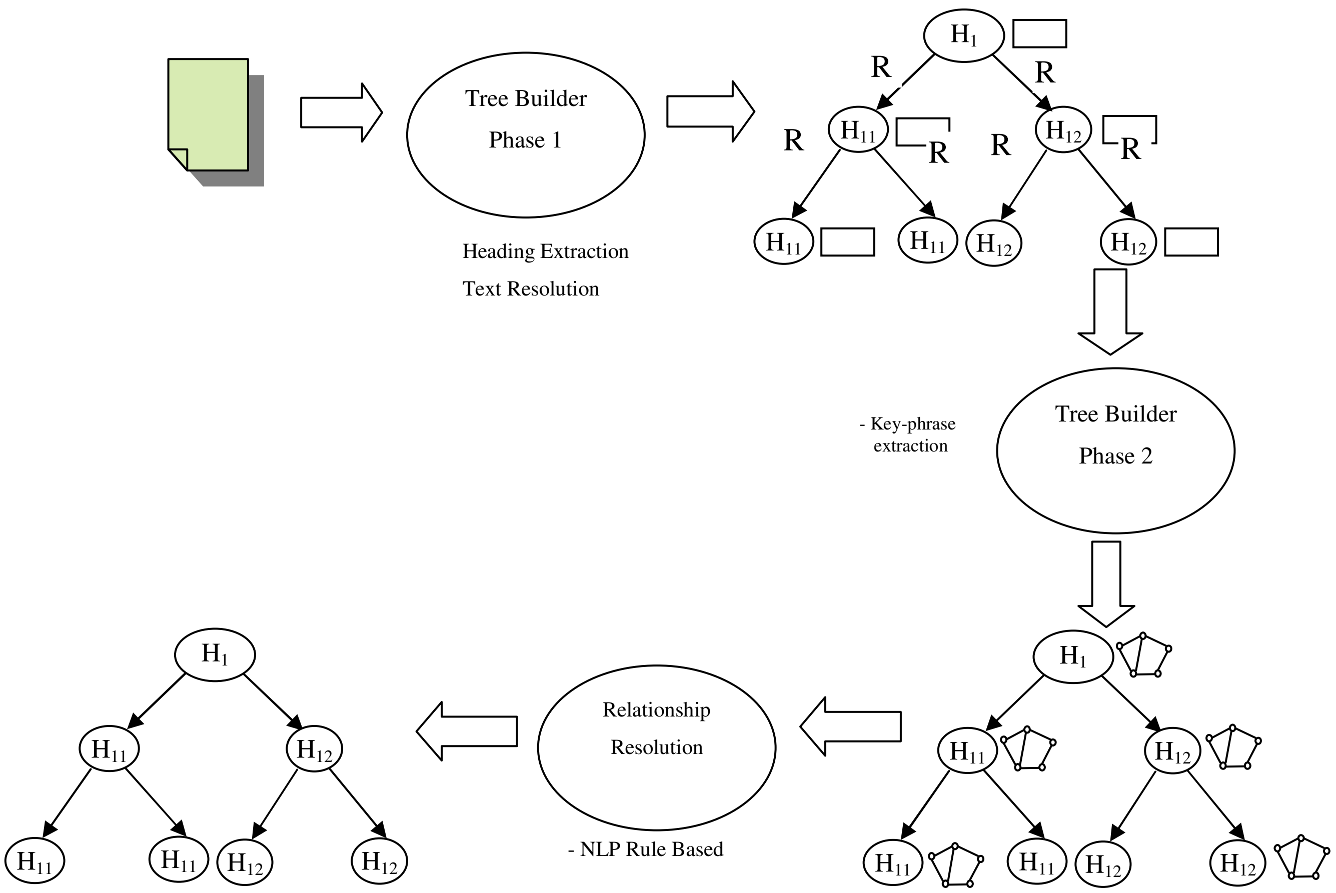
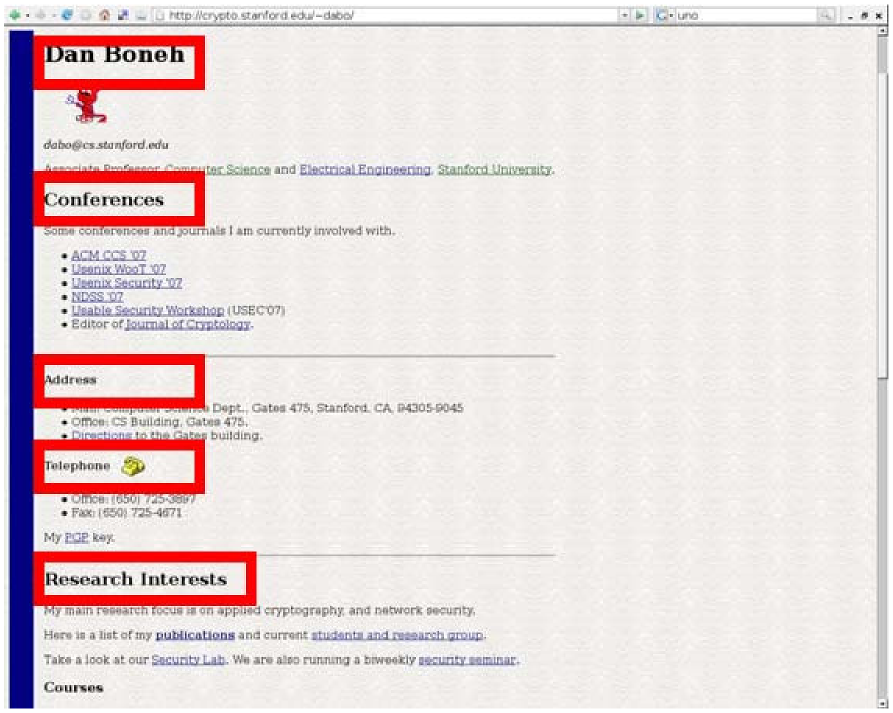


Figure 7: Document representation process

3.2.1 Tree Builder Phase 1

We breakup the problem of heading extraction into three parts:

1. Extract headings
2. Resolve text under each heading.



http://crypto.stanford.edu/~dabo/

Dan Boneh

dabo@cs.stanford.edu

Associate Professor, Computer Science and Electrical Engineering, Stanford University.

Conferences

Some conferences and journals I am currently involved with.

- ACM CCS '07
- Usenix WooT '07
- Usenix Security '07
- NDSS '07
- Usable Security Workshop (USEC'07)
- Editor of Journal of Cryptology.

Address

- Mail: Computer Science Dept., Gates 475, Stanford, CA, 94305-9045
- Office: CS Building, Gates 475.
- Directions to the Gates building.

Telephone

- Office: (650) 725-3897
- Fax: (650) 725-4671

My PGP key.

Research Interests

My main research focus is on applied cryptography, and network security.

Here is a list of my **publications** and current students and research group.

Take a look at our Security Lab. We are also running a biweekly security seminar.

Courses

Figure 8: Extracting Headings

We have developed a heuristic based algorithm to extract headings from html documents. The details of implementation of the algorithm are given in section ***. The above algorithm was successful in extracting information from only 65% of the pages classified as faculty pages.

The main problems encountered were

(i) Ill-formatted HTML

(ii) No Heading in the page

(iii) Homepages constituting of multiple pages.

3.2.2 Tree Builder Phase 2

The main task involved in this phase is the extraction of ***Key phrases*** from each text bundle. Lot of research has been published attacking this problem. The following are the some of the possible approaches that we found most appropriate for our purpose:

1) Simple Heuristic Based Approach – The approach by Gutwin and Nevill-Manning [35] shows a simple procedure for key phrase extraction based on the Naive Bayes learning scheme performs comparably to the state of the art. It goes on to explain how this procedure's performance can be boosted by automatically tailoring the extraction process to the particular document collection at hand.
2) Document Index Graph - There has been lot of research on keyphrase extraction from documents. The model presented by Hammouda and Kamel [6] for document representation is called the Document Index Graph (DIG). This model indexes the documents while maintaining the sentence structure in the original documents. This allows us to make use of more informative phrase matching rather than individual words matching. A list of matching phrases between two documents is computed by intersecting the sub graphs of both documents.

3.2.3 Relationship Resolution

The main task involved in this phase is:

1. Define NLP rules
2. Find relation between key-phrases extracted in the previous phase.

Chapter 4: Ontology

4.1 Motivation

Recent research has demonstrated the important of ontology and its applications. For example, while designing adaptive learning materials, designers need to refer to the ontology of a subject domain. Moreover, ontology can show the whole picture and core knowledge about a subject domain. Research from literature also suggested that graphical representation of ontology can reduce the problems of information overload and learning disorientation for learners. However, ontology constructions used to rely on domain experts in the past; it is a time consuming and high cost task. Ontology creation for emerging new domains like e-Learning is even more challenging. The aim of this work is to construct e-Learning domain concept maps, an alternative form of ontology, from documents from the web. We crawled content from the web using a focused crawler, and apply text-mining techniques to automatically construct concept maps for e-Learning domain.

4.2 System Design

We present a (semi) automatic framework that aims to produce a domain concept map (DCM) from text and to derive ontology of the world from this concept map. This methodology is particularly aimed at the educational field because of the need of such structures (Ontology and CM) within the e-Learning communities to sustain the production of e-Learning resources tailored to learner's needs.

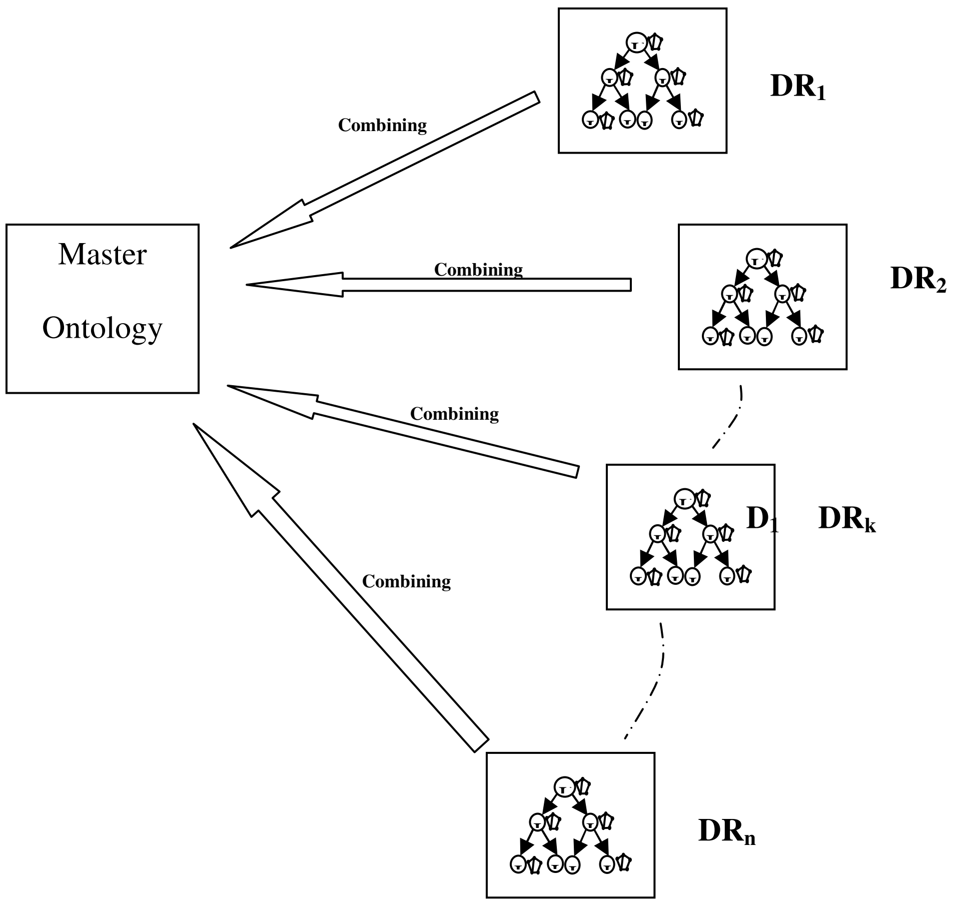


Figure 9: Ontology Generation Architecture

The algorithm demonstrates the detailed steps that transform textual resources (and particularly textual learning objects) into a domain concept map and how the more abstract document representation is transformed into more formal ontology.

4.2.1 Data Structure of Ontology

```
Node {
    Name (Type: String)
    AssociatedText (Type: String)
    Keyphrase (Type: String)
    ParentList (Type: Array of (Node *, Relationship))
    ChildList (Type: Array of (Node *, Relationship))
}

Graph {
}
```

4.2.2 Algorithm for Combining Documents

```
Input: Document Representation DR(DR), Master Ontology(MO)
Output: Expanded Master Ontology
Function combineDocument(DR, MO)
    For Each Node in DR
        //Search for the current node in Master Ontology
        //Returns a ranking with a Confidence Factor
        RankedMatchingNodeList = SearchandRankNodeInMO(Node, MO);
        If CF(RankedMatchingNodes[0]) > $CF_{threshold}$
            //Adding a node to the master ontology
            AddNode(RankedMatchingNodes[0], MO);
        End
    End
End
```

4.2.3 Algorithm for Ontology Creation

```
Input: DocumentList ($DR_1$ - $DR_n$), OntologyList($O_1$, $O_2$, … $O_m$)
Output: Master Ontology (MO)

Function OntologyCreate(DocumentList, OntologyList)
    //Initialize the Master Ontology with existing ontologies.
    MO = Initialize (OntologyList);
    //Iteratively Combine DR to MO
    For Each Document in ($DR_1$ - $DR_n$)
        //Combine Document with MO – Section 5.7.2.2
        MO = CombineDocument(DR, MO);
    End
End
```

4.2.4 Algorithm for Searching and Ranking Nodes

```
Input: Master Ontology (MO), Node to be searched (QueryNode)
Output: RankedMatchingNodeList

Function SearchandRankNodeInMO (QueryNode, MO);
    //Initialize  to empty
    MatchingNodeList = {};
    RankedMatchingNodeList = {};
    //Search Level 1: Search for Node.name in the Master Ontology
    MatchingNodeList = Search (Node.name, MO);
    //Search Level 2: Find confidence factor of each match for ranking
    For Each Node in MatchedNodeList
       CF = CalculateConfidence(QueryNode, Node);
```

```
            //AddNodeToList adds node at its correct position (sorted)
            RankedMatchingNodeList = AddNodeToList(RankedMatchingNodeList, CF);
        End
        Return RankedMatchingNodeList;
End
```

4.2.5 Algorithm for Calculating confidence in matching nodes

```
Input: Nodes to be matched (Node1, Node2)
Output: ConfidenceFactor (CF)

Function CalculateConfidence (QueryNode, MO);
        CF = Metric();
        Return ConfidenceFactor;
End
```

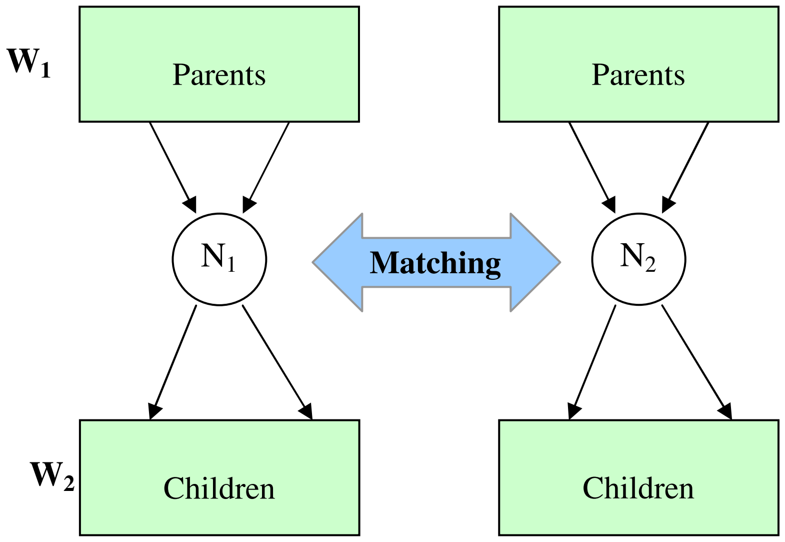


Figure 10:Node Matching and ConfidenceFactor calculation

There can be various approaches used for node matching. The simplest approach would be statistical matching of the text associated with the two nodes. The node matching algorithm starts from the root of the document representation. The criteria that we propose to use are:

1) Parent Nodes Match – W_1* (NumberOfMatches)
    a. Parents upto 3 levels matched
2) Child Nodes Match – W_2 * (NumberOfMatches)
    a. Children upto 3 levels matched
3) Node Text Matching – W_3 * (NumberOfMatches)

Chapter 5: Experiments and Results

5.1 Data Set Details

We first crawled Yahoo Directory of Universities. This gave a list of 16000 university websites. Of those we crawled 600 websites using a focused crawler. After classification as faculty pages we had a 1034 pages identified as faculty pages. The heading extraction algorithm was tested on these pages.

5.2 Results for Heading Extraction Algorithm

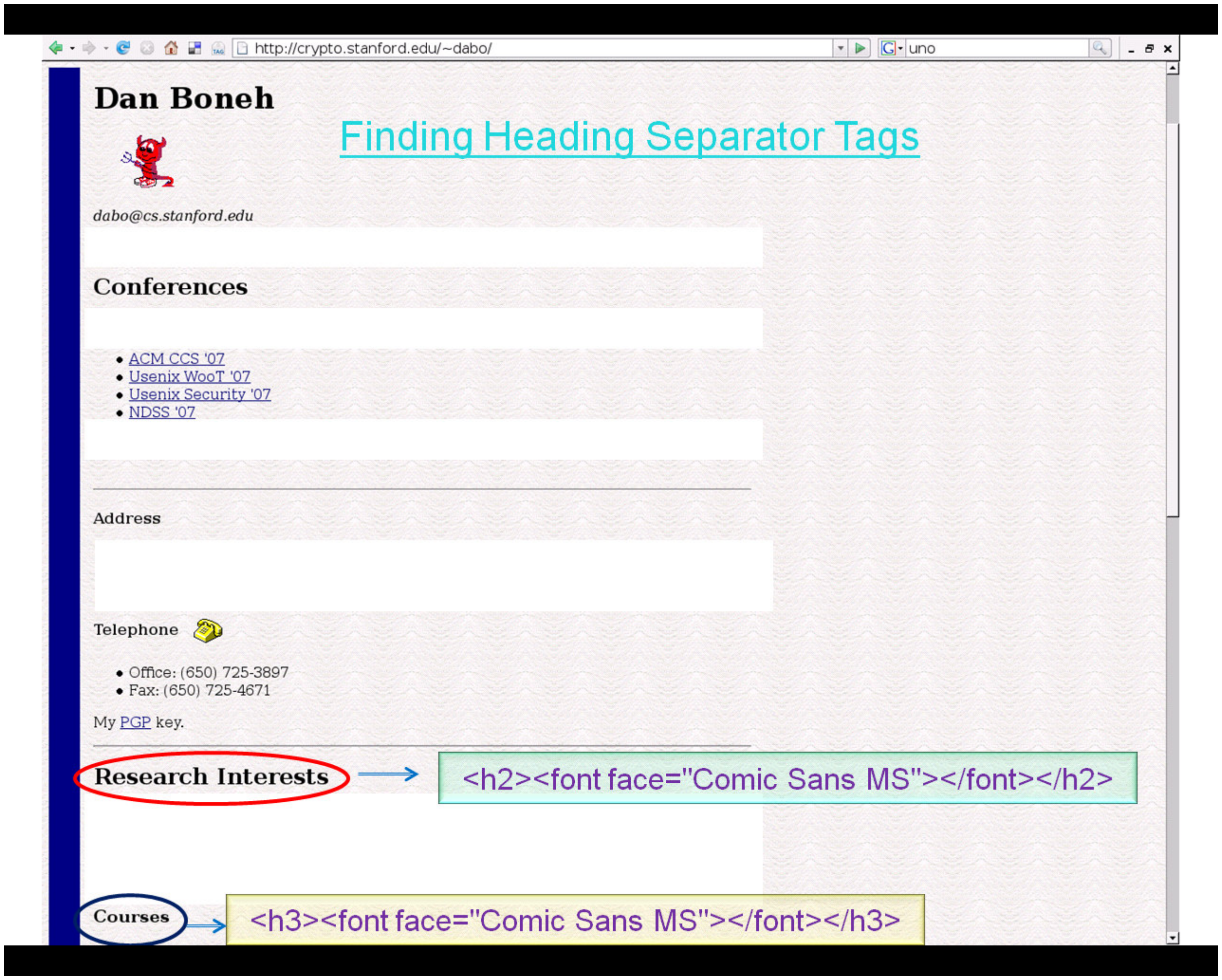
For tagging our system for intelligent search, we need to parse the pages and find out relevant portions. These portions then must be tagged in order to facilitate tagged based searching. The dataset we used to test this algorithm was faculty pages from university websites.

In Universities, most of the faculty pages are database generated, so they have a predefined structure. Even if they are not database generated, most of them have same structure in terms of - like they have some sections like - About Me, Research Interests, Publications and Courses on their main page. We analyzed lot of faculty pages and based on the general structure, we decided few set of keywords which are most likely to appear as Heading on the faculty's main page. Those keywords are -

- Publications
- Papers
- Research
- Course
- University Activities
- Honors
- Professional Activities
- Awards
- Lectures
- Contact
- About Me
- Teaching
- Office Hours
- Projects

Our main aim here was to find the headings on a faculty page, and then tag the data within that heading with the Heading. Page is then looked for the presence of above keywords. After removing paragraphs from the page, we look for the presence of above keywords in the remaining text. In database generated faculty pages, it is most likely that all the headings have same HTML tags around them. We will call them as Separator Tags for Headings. Even in the pages which are generated by faculty themselves, they are most likely to put all the headings inside the same Separator Tags.

So, for the headings which are not included in the above list we can identify them by their separator tags. For Example,



```
<h2><font face="Comic Sans MS"></font></h2>
<h3><font face="Comic Sans MS"></font></h3>
```

Figure 11: Heading Demostration 1

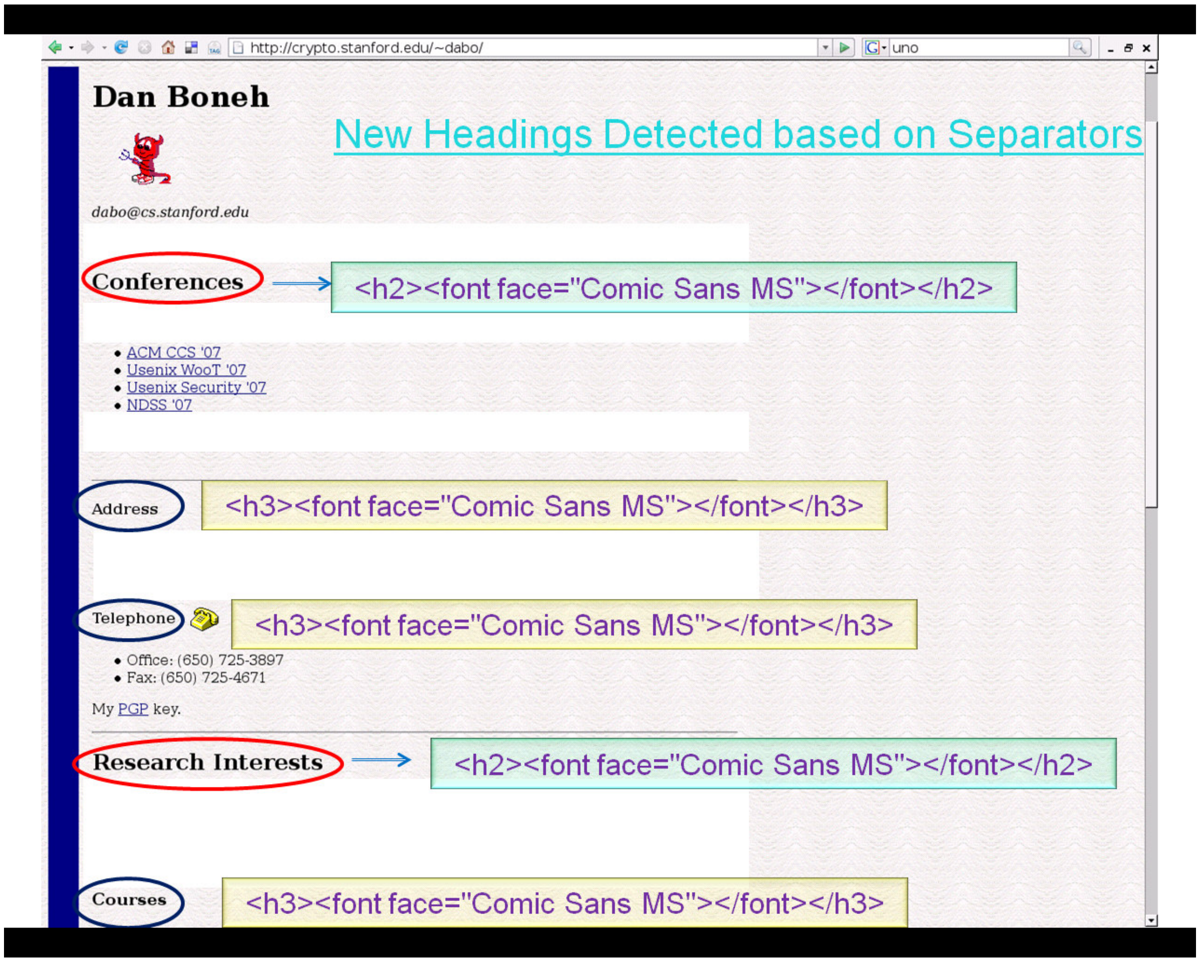
We downloaded the page of Prof. Dan Boneh from Stanford University and ran our algorithm on that page for illustration. First we will have to remove the paragraphs in above page so as to remove all those parts where we can find the keyword but they are not in heading. We use the algorithm mentioned above for paragraph removal to do that. We now find out the heading sections based on the keywords. Here the headings found are:

- Research Interests
- Courses

Separator Tags are:

- Research Interests => <h2><font face="Comic Sans MS"></font></h2>
- Course => <h3><font face="Comic Sans MS"></font></h2>

After finding the Separator Tags for headings from above list, we find the presence of those "Separator Tags" in rest of the page and collect the text between them and identify those also as Heading. These keywords were previously remained undetected because of their absence in the keywords list.


http://crypto.stanford.edu/~dabo/
uno
New Headings Detected based on Separators

Dan Boneh

dabo@cs.stanford.edu

Conferences

```
<h2><font face="Comic Sans MS"></font></h2>
```

- ACM CCS '07
- Usenix WooT '07
- Usenix Security '07
- NDSS '07

Address

```
<h3><font face="Comic Sans MS"></font></h3>
```

Telephone

```
<h3><font face="Comic Sans MS"></font></h3>
```

- Office: (650) 725-3897
- Fax: (650) 725-4671

My PGP key.

Research Interests

```
<h2><font face="Comic Sans MS"></font></h2>
```

Courses

```
<h3><font face="Comic Sans MS"></font></h3>
```

Figure 12: Heading Demostration 2

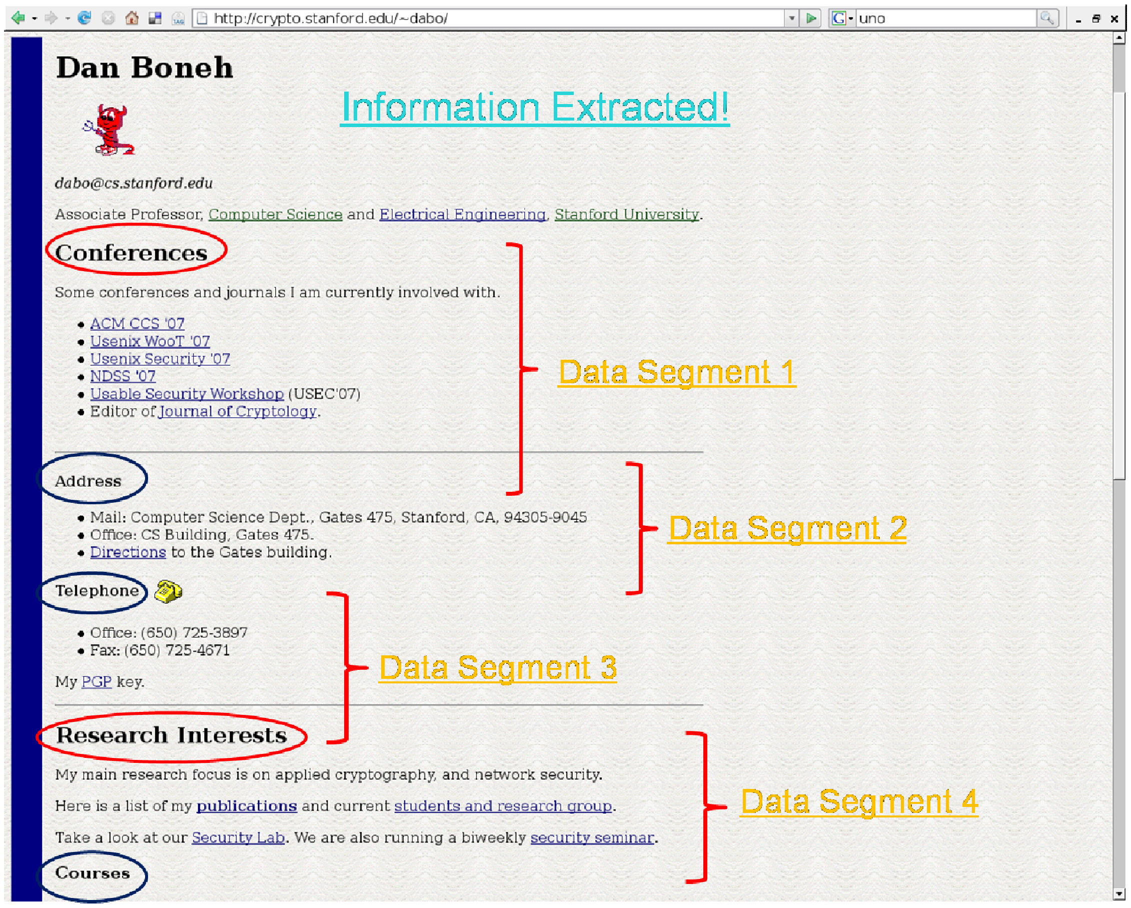
Now, in the page following set of keywords had same separator tags:

- Research Interests
- Conferences

AND

- Address
- Telephone
- Courses

Based on the above algorithm, we detected 3 more Headings in this page – Conferences, Address, Telephone. We can now analyze the text under each heading by separating the text between the two headings and putting it in the category of Heading it falls inside. That makes us separate the information from Prof. Dan Boneh's page into these categories – His Research Interests, Conferences, Address, Telephone and Courses he takes.



Dan Boneh

dabo@cs.stanford.edu

Associate Professor, Computer Science and Electrical Engineering, Stanford University.

Conferences

Some conferences and journals I am currently involved with.

- ACM CCS '07
- Usenix WooT '07
- Usenix Security '07
- NDSS '07
- Usable Security Workshop (USEC'07)
- Editor of Journal of Cryptology.

Address

- Mail: Computer Science Dept., Gates 475, Stanford, CA, 94305-9045
- Office: CS Building, Gates 475.
- Directions to the Gates building.

Telephone

- Office: (650) 725-3897
- Fax: (650) 725-4671

My PGP key.

Research Interests

My main research focus is on applied cryptography, and network security.

Here is a list of my publications and current students and research group.

Take a look at our Security Lab. We are also running a biweekly security seminar.

Courses

Figure 13: Heading Demostration 3

6.3 *Implementation of Document Representation Algorithm*

6.3.1 Proof of concept

The Wikipedia dataset was chosen to develop a proof of concept for the proposed algorithm. The algorithm needs to be testing on very large datasets to verify its applicability. We are in process of downloading the data currently. The following steps were followed to develop the proof of concept.

Step 1: Topic =Page Name

Step 2: Headings Extraction = using Table of Contents

Step 3: TextResolution = Text from one internal link to another

Step 4: Graph After phase1 formed using above.

Step 5: Keyphrases = Linked Words are taken as keyphrases.

Step 6: Graph After phase2 formed using above.

Step 7: NER from NLTK applied and Manual simple NLP rules applied for relation recognition.

6.3.2 Process Demonstration

The steps mentioned above are visually shown on a wiki page below:

Sorting algorithm

From Wikipedia, the free encyclopedia

[illegible]

Classification

[illegible]

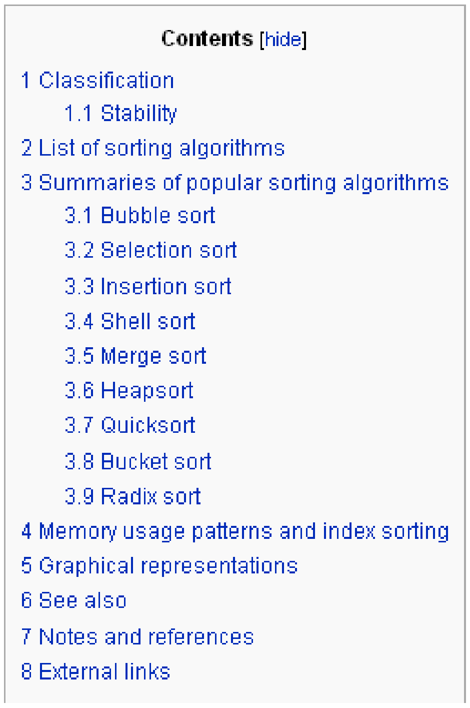
Stability

[illegible]

List of sorting algorithms

[illegible]

Figure 14: Step 1



Contents [hide]

Figure 15: Step2

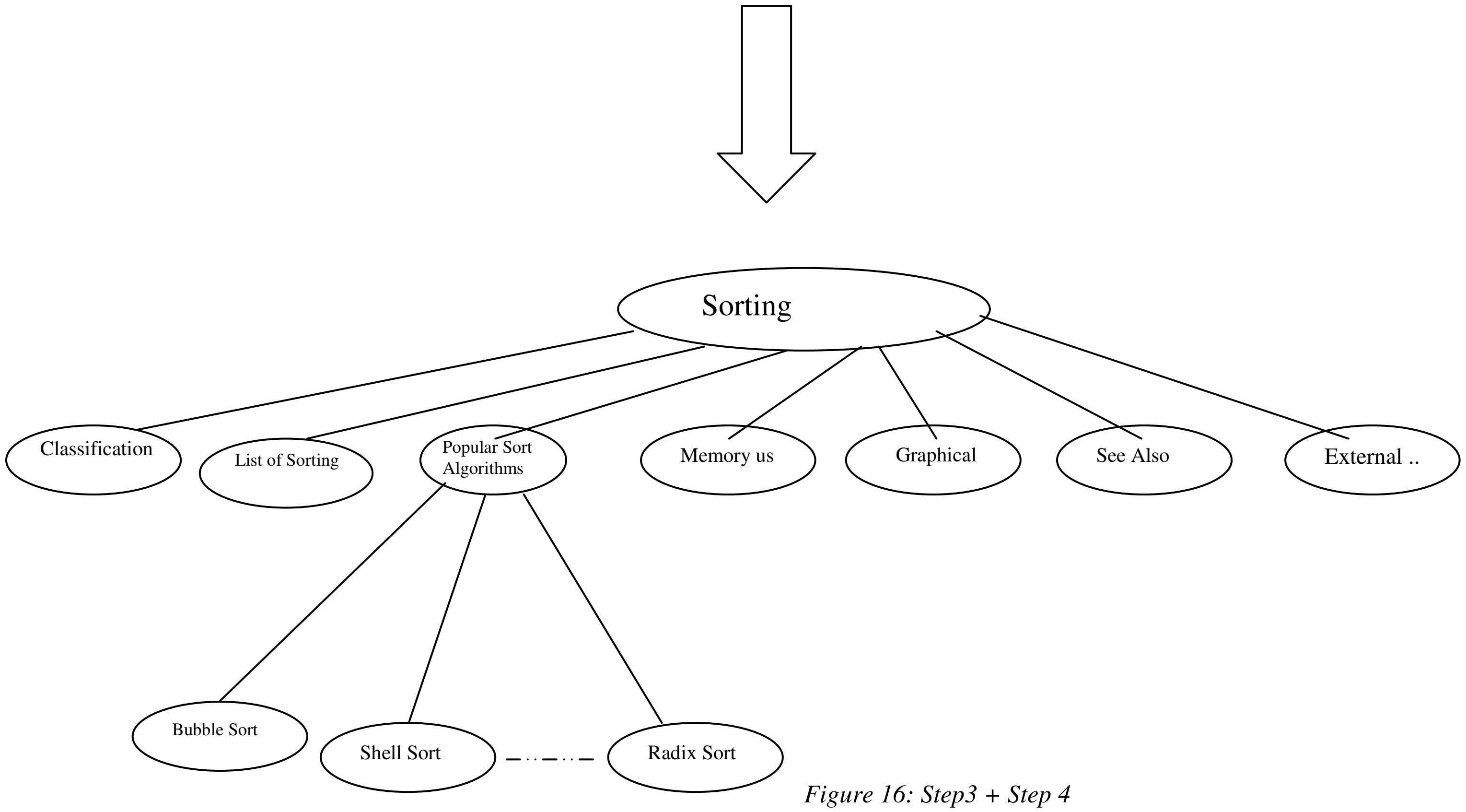
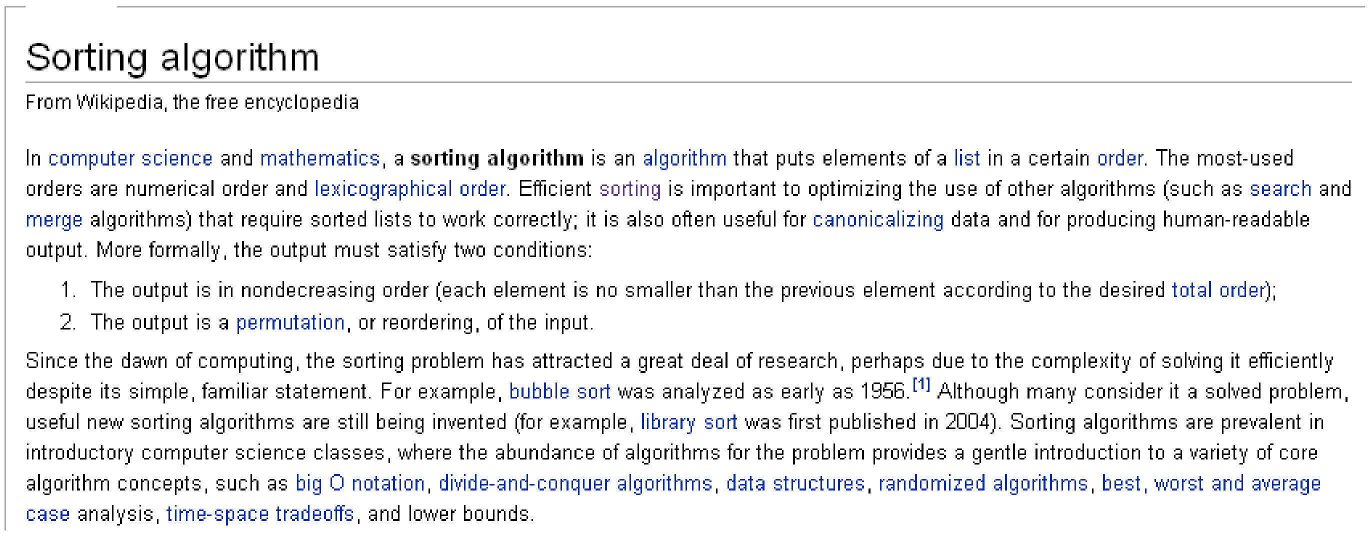


Figure 16: Step3 + Step 4



Sorting algorithm

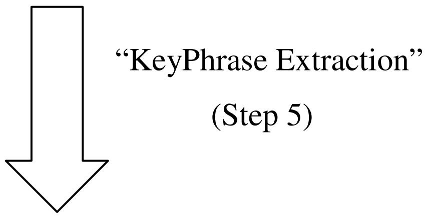
From Wikipedia, the free encyclopedia

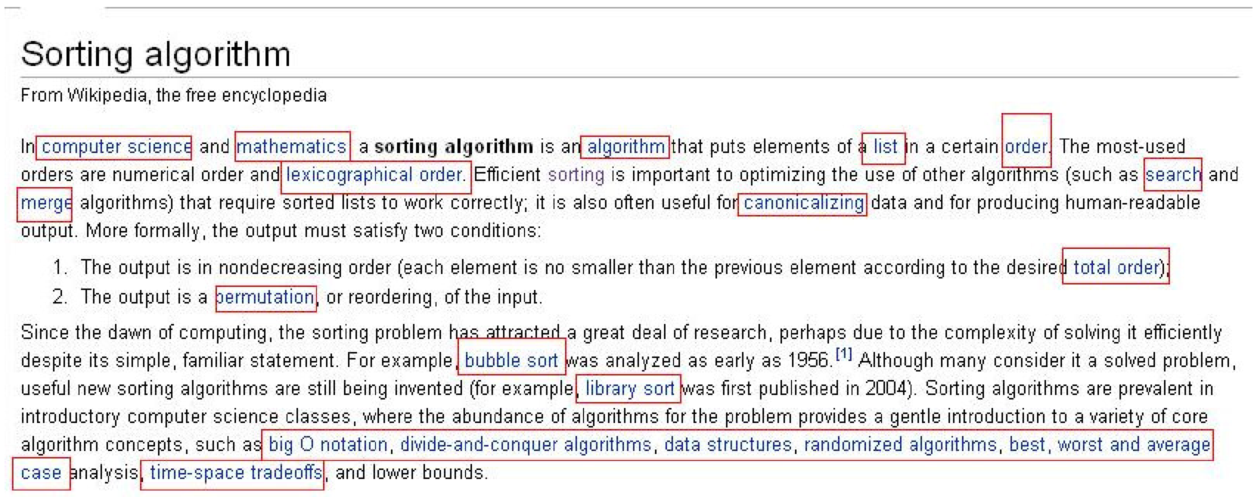
In computer science and mathematics, a **sorting algorithm** is an algorithm that puts elements of a list in a certain order. The most-used orders are numerical order and lexicographical order. Efficient sorting is important to optimizing the use of other algorithms (such as search and merge algorithms) that require sorted lists to work correctly; it is also often useful for canonicalizing data and for producing human-readable output. More formally, the output must satisfy two conditions:

1. The output is in nondecreasing order (each element is no smaller than the previous element according to the desired total order);
2. The output is a permutation, or reordering, of the input.

Since the dawn of computing, the sorting problem has attracted a great deal of research, perhaps due to the complexity of solving it efficiently despite its simple, familiar statement. For example, bubble sort was analyzed as early as 1956.[1] Although many consider it a solved problem, useful new sorting algorithms are still being invented (for example, library sort was first published in 2004). Sorting algorithms are prevalent in introductory computer science classes, where the abundance of algorithms for the problem provides a gentle introduction to a variety of core algorithm concepts, such as big O notation, divide-and-conquer algorithms, data structures, randomized algorithms, best, worst and average case analysis, time-space tradeoffs, and lower bounds.

Figure 17: Heading Demostration 1





Sorting algorithm

From Wikipedia, the free encyclopedia

In computer science and mathematics, a **sorting algorithm** is an algorithm that puts elements of a list in a certain order. The most-used orders are numerical order and lexicographical order. Efficient sorting is important to optimizing the use of other algorithms (such as search and merge algorithms) that require sorted lists to work correctly; it is also often useful for canonicalizing data and for producing human-readable output. More formally, the output must satisfy two conditions:

1. The output is in nondecreasing order (each element is no smaller than the previous element according to the desired total order);
2. The output is a permutation, or reordering, of the input.

Since the dawn of computing, the sorting problem has attracted a great deal of research, perhaps due to the complexity of solving it efficiently despite its simple, familiar statement. For example, bubble sort was analyzed as early as 1956.[1] Although many consider it a solved problem, useful new sorting algorithms are still being invented (for example, library sort was first published in 2004). Sorting algorithms are prevalent in introductory computer science classes, where the abundance of algorithms for the problem provides a gentle introduction to a variety of core algorithm concepts, such as big O notation, divide-and-conquer algorithms, data structures, randomized algorithms, best, worst and average case analysis, time-space tradeoffs, and lower bounds.

Figure 18: Heading Demostration 1

Chapter 6: Conclusion and Future Work

6.1 Conclusion

In this work, we have presented a novel approach for use of data mining for e-learning. With the explosion of the internet particularly because of the web2.0 model, a content repository has accumulated over the World Wide Web. This abundance of information if tapped efficiently can act as a very good learning resource for an intelligent tutoring system. This work is a step in the direction of bridging the information gap between an e-learning and an intelligent tutoring system.

The work presented in this thesis is done keeping in mind a novel set of applications which will be developed for the Intinno e-learning and intelligent tutoring system. We have surveyed current trends and techniques used to represent knowledge. We have then presented architecture with a set of methodologies by aggregating and customizing the state of the art research in e-learning domain.

We have implemented a proof of the concept of the proposed system architecture on Wikipedia dataset. We have also deployed the Intinno e-learning system for 25 courses with around 1000 users in IIT Kharagpur for about 8 months.

6.2 Future Work

Although we have done a lot of research on the tools developed and used in the industry as well as academia, our survey is not complete. A more thorough survey of current trends in the applicability of Data Mining to E-learning needs to be done. A large collection of data is to be collected to test the proposed architecture and algorithms.

Bibliography

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

[2] V. B. Devedzic, Key Issues in Next Generation Web Based Education, *IEEE Trans. System Man Cybernetic: Part C*, Vol. 33, pp. 339 (2003).

[3] LCMS: http://www.learningcircuits.org/2005/nov2005/carliner.htm

[4] Understanding E-Learning 2.0: http://www.learningcircuits.org/2007/0707karrer.html

[5] Joseph Psotka, Sharon A. Mutter (1988), "Intelligent Tutoring Systems: Lessons Learned", *Lawrence Erlbaum Associates.*

[6] Hammouda, K. and Kamel, M., "Data Mining in e-Learning", in Samuel Pierre (ed.) "E-Learning Networked Environments and Architectures: A Knowledge Processing perspective", series: *Advanced Information and Knowledge Processing, Springer Book Series*, 2007.

[7] Scheuer, O. & McLaren, B.M. (2008). Helping Teachers Handle the Flood of Data in Online Student Discussions. In the *Proceedings of the 9th International Conference on Intelligent Tutoring Systems.*

[8] H. Hage, E. Aimeru, "ICE: A System for Identification of Conflicts in Exams," *aiccsa*, pp. 980-987, *IEEE International Conference on Computer Systems and Applications*, 2006, 2006

[9] Hammouda, K., & Kamel, M. (2006), "Data mining in e-learning". In Samuel Pierre (Ed.), E-learning networked environments and architectures: A knowledge processing perspective, *Springer Book Series: Advanced information and knowledge processing* (pp. 1 – 28).

[10] K., S., “Reducing the space requirement of suffix trees.” *Software — Practice and Experience* 29 (1999) 1149–1171

[11] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen, “Self-organizing maps of document collections: A new approach to interactive exploration,” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, E. Simoudis, J. Han, and U. Fayyad, Ed. Menlo Park, CA: AAAI, 1996, pp. 238-243.

[12] T. Kohonen, “The self-organizing map (SOM), Neurocomputing”, 21 (1998)

[13] Plaban Kumar Bhowmick , Samiran Sarkar , Sudeshna Sarkar and Anupam Basu, “Acquisition of User Profile for Domain Specific Personalized Access”, *7th International Conference on Information Technology* December 20-23, 2004, Hyderabad.

[14] S Sarkar, P K Bhowmick, D Roy, S Sarkar, A Basu and S Ghose, “A System for Personalized Information Retrieval based on Domain Knowledge”, *National Trends in Computational Mathematics*, March 18-19, 2003.

[15] Natalya F. Noy, “Semantic Integration: A Survey Of Ontology-Based Approaches”, *SIGMOD* 2004

[16] Heflin, J., & Hendler, J. (2001). “A Portrait of The Semantic Web in Action”. *IEEE Intelligent Systems* 16(2), 54-59.

[17] Hendler, J. (2001), “Agents and the Semantic Web”, *IEEE Intelligent Systems 16(2)*, 30-37.

[18] Scott Cost, R., Finin, T., Joshi, A., Peng, Y., Nicholas, C., Soboroff, I., et al. (2002), “ITtalks: A Case Study in the Semantic Web and DAML+OIL”, *IEEE Intelligent Systems* 17(1), 40-47.

[19] Gómez-Pérez, A., & Corcho, O. (2002). Ontology Languages for the Semantic Web. *IEEE Intelligent Systems* 17(1), 54-60.

[20] McIlraith, S.A., Son, T.C., & Zeng, H. (2001). Semantic Web Services. *IEEE Intelligent Systems* 16(2), 46-53.

[21] Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R.W., & Musen, M.A. (2001). Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems* 16(2), 60-71.

[22] Lassila, O. (1998). Web Metadata: A Matter of Semantics. *IEEE Internet Computing* 2(4), 30-37.

[23] Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., & Horrocks, I. (2000, September/October). The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing* 4, 63-74.

[24] Vladan Devedzic, Education and the Semantic Web. *International Journal of Artificial Intelligence in Education* 14 (2004) 39-65

[25] Gruber, T. R., "A translation approach to portable ontology", *Knowledge Acquisition*, Vol. 5, No. 2, 1993, pp. 199-220.

[26] Natalya F. Noy, "Semantic Integration: A Survey Of Ontology-Based Approaches", *SIGMOD*, Record, Vol. 33, No. 4, December 2004.

[27] C. Welty. Ontology research. *AI Magazine*, 24(3), 2003.

[28] Khalifa, M., and R. C. Kwok, "Remote learning technologies: effectiveness of hypertext and GSS", *Decision Support Systems*, Vol. 26, No. 3, 1999, pp. 195-207.

[29] Chularut, P., and T. K. DeBacker, "The influence of concept mapping on achievement, self-regulation, and selfefficacy in students of English as a second language", *Contemporary Educational Psychology*, Vol. 29, No. 3, 2004, pp. 248-263.

[30] Chen, N., Kinshuk, P., Wei, C., and Chen, H. 2006. Mining e-Learning Domain Concept Map from Academic Articles. In *Proceedings of the Sixth IEEE international Conference on Advanced Learning Technologies* (July 05 - 07, 2006). ICALT. IEEE Computer Society, Washington, DC, 694-698.

[31] OWL Guide http://www.w3.org/TR/owl-guide/

[32] http://protege.stanford.edu/download/ontologies.html

[33] SCORM: http://en.wikipedia.org/wiki/SCORM

[34] N. Kushmerick, Gleaning the Web, *IEEE Intelligent Systems,* Vol. 14, 20 (1999).

[35] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. 1999, " Domain-specific keyphrase extraction", In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*.