INLS 490-154: Information Retrieval Systems Design and Implementation. Spring 2009.

1. Introduction

Chirag Shah*
School of Information & Library Science (SILS)
UNC Chapel Hill NC 27599
chirag@unc.edu

1   Introduction

Tools for organizing and accessing information have become indispensable. It is critical, therefore, to understand their design and operational foundations. In this course students will have an opportunity to learn about search engines, web crawling, and some Web 2.0 technologies based on hands-on experience and with a focus on techniques that can be used to access, retrieve, organize, and present information. Students will work with practical developmental tools and learn relevant concepts through experimentation. For instance, students will employ an open source search engine and learn about indexing, retrieving, and ranking techniques.

2   Goals and objectives of this course

By the end of the course, students should be able to:

1. Use and/or implement various search engine services such as stemming, indexing, retrieval, and ranking.

2. Build a set of Web crawlers to collect data from the Web.

3. Mesh a set of Web 2.0 services to implement unique applications.

Various services on the Web such as search engines are integrated in our daily information seeking processes. While we may not reinvent those wheels, as a part of this course, we will be learning how such search services can be created. Due to the interest in protecting their trade

---

secrets, these search services do not reveal the details of various processes and components that they have, but we have a fairly good understanding of how they should be working. As a part of this course, we will explore several algorithms, systems, and methods that make an information retrieval service successful.

In this course we will primarily work with unstructured data. While several databases with structured data exist, a huge portion of the Web contains unstructured textual information. Due to its unstructured nature, this information is much harder to deal with. This course will introduce the students with some of the challenges in working with unstructured data and provide insights into how one could go about addressing them. We will not only build the systems that do the work in the back-end, but also create user interfaces that provide a user access to those tools and information.

It is also important to note what this course is NOT intended to do. As noticed before, we will not work with a lot of structured data. While we will be using various programming languages and tools to create and interface with different services, it is not a goal of this course to teach how to program. As noted in the following section, a student is required to have some knowledge of programming. While we will prepare the user interfaces to provide access to the information and interact with the tools, our goal is not to build highly efficient, interactive, and/or commercial systems. We will also not use any proprietary software (other than the operating system).

3   Requirements

The students are expected to have a previous exposure to some programming (C, Java, Perl, or PHP). Basic programming experience acquired in an introductory programming course such as INLS 490-153 or in some professional settings is recommended.

If the following lines make no sense to you, chances are this course is not right for you!

```
begin
    $query = <STDIN>;
    $document = "Hello world";
    if ((strcmp($query, $document) == 0)
        print "Match\n";
    else
        print "Sorry, no match found.\n";
end
```

4   Outline of the course

We will start this course from the structured data domain. In that domain, we'll see how one could store data to and access it from a MySQL database. We will also see how this back-end processing can be incorporated to a front-end, so that we could provide an information retrieval interface to the users.

With this quick introduction to structured data processing and information retrieval, we will move toward the unstructured domain. Here we will deal with text documents and

see how we can (1) represent textual information in terms of an index, (2) parse a query for retrieval purpose, (3) match document and query representations, (4) retrieve a set of documents matching to a query, and (5) order/rank or organize the retrieved results for the user. This model of information retrieval is shown in Figure 1.
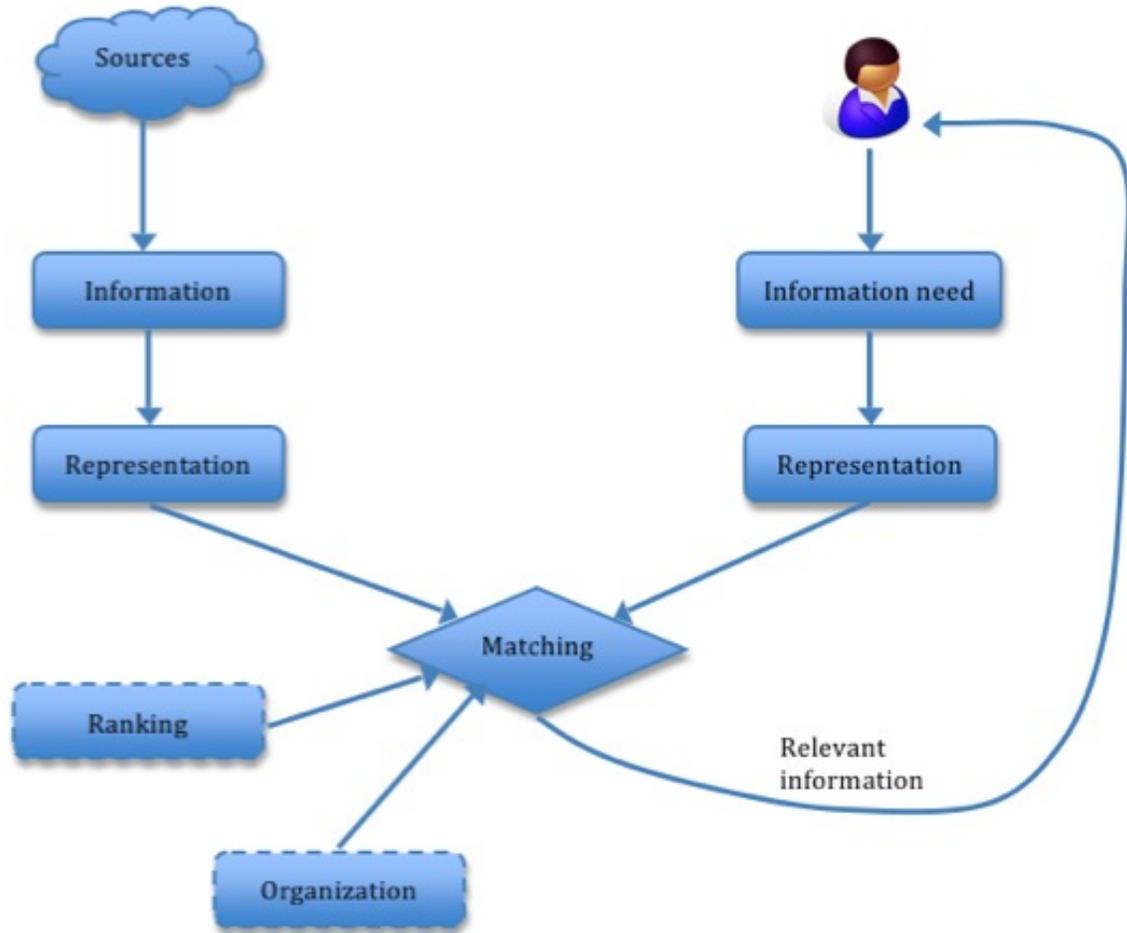

Figure 1: A model of Information Retrieval (IR)

Keeping in line with the spirit of this course, we will learn about each of the components shown in the above figure by experimentations. To be specific, we will write and/or use off-the-shelf programs to process unstructured textual information (both documents and queries) and perform the matching and retrieval. After trying all these things "manually", we will move toward automatically accomplishing the same using Lemur Toolkit.

With the use of Lemur Toolkit, we will also implement and study various retrieval models such as vector space, language models, and probabilistic models.

Next, we will look at evaluating the effectiveness of an IR system. Once again, we will accomplish this by using the IR systems that we built so far in the course. Before we move on, we will work on creating user interfaces for these systems focusing on enhancing user interaction to an IR system.

The next three topics will round-up our exploration of designing and implementing IR

systems. They are (1) crawling the information from the Web, (2) meshing various Web 2.0 services for IR purpose, and (3) organizing collected or retrieved information.

Depending on the time left in the course and the need, we may talk about some advance topics and review some of the topics already covered. It is important to note here that due to the dynamic nature of this course, the syllabus may keep changing throughout the course. Please keep checking the course website for any updates.

5   Getting started with MySQL

Following are some of the most frequent operations on a MySQL database. In these examples, we are working with the 'world' database available from MySQL website.[1]

1. Create a database.
   ```
   create database 'world';
   ```

2. Create a table.

   ```
    CREATE TABLE 'City' (
     'ID' int(11) NOT NULL auto_increment,
     'Name' char(35) NOT NULL default '',
     'CountryCode' char(3) NOT NULL default '',
     'District' char(20) NOT NULL default '',
     'Population' int(11) NOT NULL default '0',
     PRIMARY KEY  ('ID')
   );
   ```

3. See the structure of a table.
   ```
   DESCRIBE City;
   ```

4. Insert a record in a table.
   ```
   INSERT INTO City VALUES('','New York','USA','New York','8008278');
   ```

5. Get all the records from a table.
   ```
   SELECT * FROM City;
   ```

6. Count all the records of a table.
   ```
   SELECT count(*) FROM City;
   ```

7. Get a set of records matching some criteria.
   ```
   SELECT * FROM City WHERE population>7000000;
   SELECT Name,Population FROM Country WHERE Region='Caribbean'
                                   ORDER BY Population;
   ```

---

[1]http://dev.mysql.com/doc/

6 Summary

- This is a unique course that introduces several of the core aspects of an IR system design and implementation by the means of experimentation.

- We will learn various concepts by first trying them in practice.

- The majority of the course is designed to deal with unstructured textual information.

- We will try to manually do as many things as possible. Then we will move toward using some off-the-shelf tools.

- Regular assignments will be given to let the students extend their work and understanding that they developed during the class.

- The course expects one to know a little bit of programming, but that is not the focus. As long as you could code a few basic things in almost any programming language, you will be fine.

- While expertise in programming is not required, an open mind and a sense of adventure is mandatory!