

University of Illinois at Urbana Champaign

UIUC

People Finder

Heewon Jung
John Laipple
Ricardo Redder
Sena Lee
Seung Pyo Lee

UIUC People Finder

University of Illinois at Urbana Champaign

Instructor: Chengxiang Zhai

Class: CS511 Advanced Database Management Systems

Date: December 13, 2006

Group:

Heewon Jung (hjung20@uiuc.edu)

John Laipple (laipple@uiuc.edu)

Ricardo Redder (rredder2@uiuc.edu)

Sena Lee (senalee2@uiuc.edu)

Seung Pyo Lee (slee232@uiuc.edu)

Table of contents

Introduction.....	4
Entity retrieval problem	5
Motivation.....	5
Formal definition	6
UIUC People finder	7
Envision	7
Environment.....	7
Phonebook.....	7
UIUC Web Search	8
Google.....	9
Yahoo.....	9
Google Images	10
Method	11
Architecture.....	12
Organization.....	13
Cache Manager	13
Searchers.....	14
Ranker	15
Phonebook.....	15
Database.....	15
User interface.....	15
Implementation	16
Demonstration.....	17
Future work.....	18
References.....	18

Introduction

The UIUC People Finder is an attempt to apply entity retrieval techniques to the context of the University of Illinois at Urban Champaign. The entity retrieval problem is basically the process of finding information about a given entity, like a person, or an organization. In this project, the entities are represented by the people related to the university, and the information, is all the information on the internet.

The system is built on the top of other major systems, like UIUC Phonebook, Google, and Yahoo. The application adds a semantic layer to these systems, and integrating their outputs, it is possible to achieve a consistent interface for executing queries about the people related to the college.

The challenges from the project and the solutions are described in this report. First, a theoretical description of the entity retrieval problem and its application on the project is presented. After that, the environment is described, followed by the solution adopted. To conclude the report some future improvements are suggested.

Entity retrieval problem

Motivation

Personal web-pages have become an important way to connect people and organizations, more and more web-pages are produced everyday. The simplicity involved in the creation of a webpage encourages many people to publish more and more documents, and other types of information online. This abundance of web-pages led to another type of applications, the search engines, which are intended to help the user to easily locate the web-pages which he or she is looking for.

However, the internet is not the only source of information available; databases are largely used by organizations, in order to store information about related people, like its employees, or related organizations, like its clients. This information is generally limited to the organization interests, like name, phone, address, department, etc.

Within this scenario, two great information sources might be observed, internet, and databases. They have different structures, different formats, and even different purposes; nevertheless they overlap in the sense that part of both refers to similar entities, where the term entity refers to either a person or an organization.

One interested in a given entity, may take benefit from these differences between the two information sources, and use the best of both. For instance, one may use the database to search for name, address, phone, email and department, while using the search engines on the internet, would return web-pages, blogs, documents, and other information.

Despite the fact that both information sources are disconnected the information they provide might be used to improve the accuracy of the searches. In fact, this is unconsciously done everyday, for instance, in the previous scenario, one could use the database information to improve the accuracy of the search on the internet.

To exemplify the situation, imagine a university, which has a database about its professors. This database may include the name, department, phone and email, however it does not contain any information about professor's publications. Nevertheless, many of these publications might be found on the internet. In the case that one wants to know more about a professor, a simple search in the database would give limited information. To overcome this lack of information provided by the database, a search on the internet could be made. Searches on the internet may return thousands or millions of results, which makes them almost useless if they are not accurate enough to return the expected results among the first ones. Thus, to improve the accuracy of the internet search, besides the professor's name, the information retrieved from the database could also be used, making the search more useful.

It is easy to extend this scenario to other situations, where people have a small knowledge about an entity – like the name, then using a specific and accurate source more information is acquired – like a database, and after that using the previous acquired

information a more general and large information source is used – like the internet. The information set about the entity grows incrementally.

Besides the database or the internet other information sources could be used, for instance instead of performing a search all over the internet, the search could be restricted to only a small part of it, like a specific domain, or a set a of documents.

With these examples it is easy to realize the importance of this type of task, which may be understood as an entity retrieval problem. Due to the frequency of the task, it is also reasonable to provide an automatic way to execute it. For instance, in the aforementioned example, a system could receive the professor's name, execute a query in the database, and then execute a search on the internet, providing the results under a unified user interface. As this would be executed by a computer program, and not by humans anymore, more complex algorithms could be used to improve the retrieval accuracy.

Formal definition

Informally, the entity retrieval problem consists of finding related documents of an entity based on the available information, using different information sources. For this project the definition is further simplified, in order to match the needs. A variation of the definition in [1] will be used:

Let E be a set of real world entities.

Let T be a relational table with attributes A , such that each tuple describes some aspects of the entities in E .

Let C be a set of documents.

One, who is interested in an entity e from E , would pose a query containing a basic description of e , like name.

The expected output is the description of e present in T , and a set of documents related to e present in C .

The entity retrieval problem has an important characteristic, its semantic nature. Based on the previous definition, it is noticed that the output consists of the information related to the entity e , the problem is how to define this relation. Currently, it is only possible for a human, to ultimately decide whether a given document is related to an entity, thus this check is made manually by analyzing the retrieved documents and deciding whether it is a positive result or not.

UIUC People finder

Envision

The UIUC People Finder is proposed as an application of entity retrieval techniques on the UIUC domain. It is designed to search for information about entities related to the university – professors, students, staff in general.

The idea is to use the existent database to provide structured information, and then use this information to retrieve documents from different search engines, in order to take advantage of the different services provided.

After this information is collected, a personal web-page should be generated to present the information about the person.

Environment

Currently, there two main search systems in the UIUC domain, the Phonebook and the web search. Besides that, three other web search engines were chosen to provide unstructured information: Google, Yahoo, and Google Images.

Phonebook

The Phonebook is a bridge between the web and the internal database, thus despite being presented as a web-page, the underlying information is structured. It contains basically the following information: alias, name, pretty name, first name, middle name, last name, email, phone, office phone, address, office address, title, department, www, type. Most of these attributes are self explanatory, thus only some of them are explained bellow:

- alias: the unique identifier of a user inside the UIUC domain; it is derived from the name.
- name: the combination – last name + first name + middle name
- pretty name: the direct form of the name – first name + middle name + last name
- title: the titles of the person – assistant professor, coordinator of project, professional, etc.
- www: a personal web page
- type: some keywords identifying the person

The Phonebook is presented in Figure 1.

It has a slightly different format for student entries, however these differences are not relevant.

Phonebook Gateway
University of Illinois - Urbana-Champaign - ns.uiuc.edu

Name

Chengxiang Zhai - czhai@uiuc.edu

```
alias: czhai
name: zhai chengxiang
pretty_name: chengxiang zhai
first_name: chengxiang
last_name: zhai
email: czhai@ad.uiuc.edu
phone: (217) 244-4943
office_phone: (217) 244-4943
address: computer science
: 2116 siebel center
: mc 258
: 201 n goodwin
: urbana, il 61801
office_address: computer science
: 2116 siebel center
: mc 258
: 201 n goodwin
: urbana, il 61801
title: asst prof
: asst professor, igb, institute for genomic biology
: assistant prof, library & information science
department: computer science
type: person phone staff
```

Figure 1

UIUC Web Search

The UIUC Web Search provides most of the capabilities commonly presented by other web search engines, however it has the advantage to be more restricted, thus it is likely to provide more accurate information, since the conflicts between names are less frequent. Despite these differences, it has the same format of other web search engines, it receives a text as a query, and returns the URLs that approximate more to query.

The UIUC Web Search might be seen in the Figure 2.



Figure 2

Google

Google is probably the most famous web search engine, it covers a large number of webpages, thus if a person has published a document online, like a paper, the probability that Google has indexed it is very high. Thus, the system may take advantage from this fact in order to extend the search over document in web, and not only under the UIUC domain.

Figure 3 presents a typical result from a Google's query.

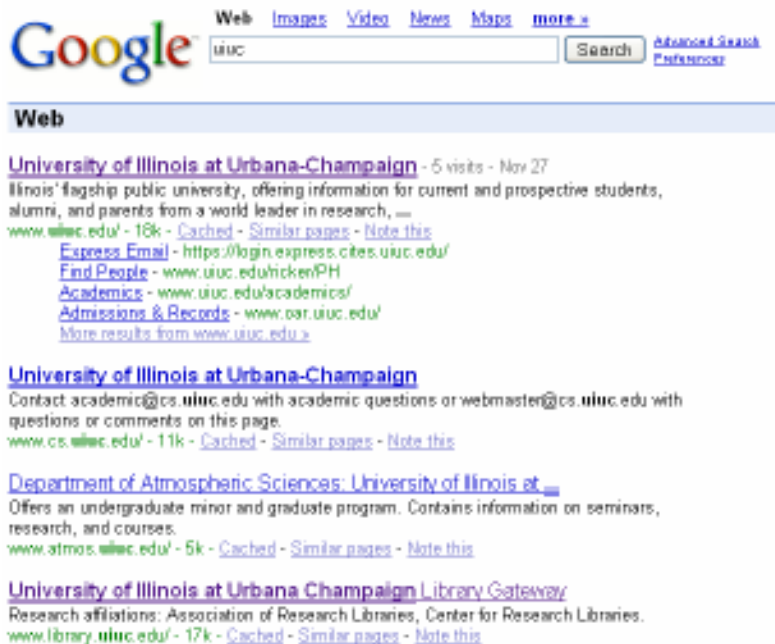


Figure 3

Yahoo

Another famous web search engine is provided by Yahoo, it is very similar to Google, in fact, as only the basic characteristics from both are explored, the results will also be very similar. The main reason to add a similar search engine, is the proof of concept, as the idea of the project is to apply the concepts from the entity retrieval field, it is important to show that system can interact with different information sources.

Figure 4 shows a picture from Yahoo website.

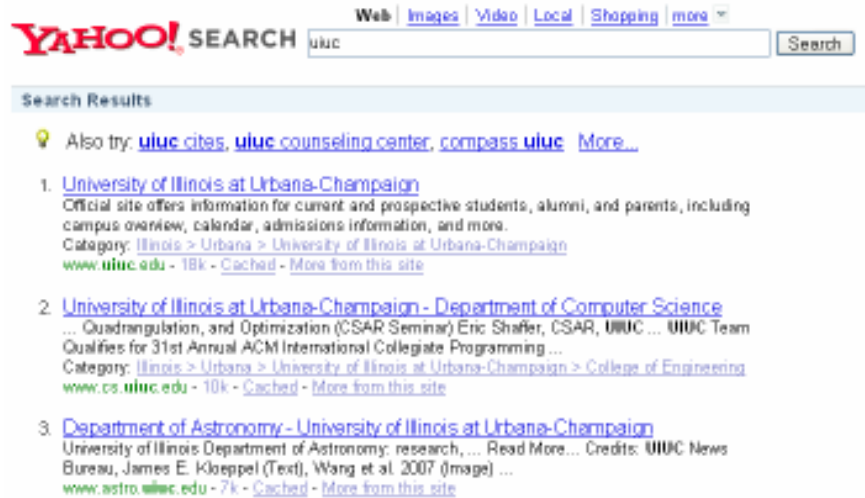


Figure 4

Google Images

Currently, the web search engines are starting to provide more than only a search over the webpages, there is a trend to offer more specialized content, like images or videos. Besides the format specialization, there is also the content specialization, like Google Book Search, or Google Blog Search. These types of services may be very helpful to the entity retrieval problem, since they already have a layer of meaning, increasing the semantic of the results.

For instance, if a search for a person is executed in the context of a book search, it is expected to return the books the person has written or collaborated, as the search is specialized in books only. It is also possible to assume that the results will contain only information about books, there is already a semantic embedded in the service.

This project tries to take advantage of this existent semantic layer, in order to acquire on of the most important information about a person, its picture. This is achieved by using the Google Image Search service.

Figure 5 shows a example of a search using Google Image.



Figure 5

Method

Within the given scenario, it is possible to specify how exactly the entity retrieval problem will be addressed.

The set of real world entities E corresponds to the people related to the college, professors, students, etc.

The relational table T is provided by the Phonebook.

The set of documents C is provided by the search engines: UIUC Web Search, Google, Yahoo and Google Images.

The system will first query the Phonebook with the words given by the user, and then it will use this information to query the search engines. After the information is gathered, it will be analyzed and ranked, according to the similarity to the entity. Finally, a personal webpage is built and presented to the user.

In order to rank a webpage, the system will access the webpage, read it, and then search for specific terms within it, like the full name, email, phone, etc. The webpage starts with a zero score, and for each found term in the webpage the score will be incremented. Each of these terms will have different weights, which reflect the importance of the term to identify the webpage as belonging to the person.

For instance, the following weights might be assigned to the terms:

full name = 10

email = 100

If a document contains only the full name of the person and not the email, it will receive only 10 points. If the webpage contains both, it will receive 110 points.

This ranking schema was created in order to overcome some limitations of the search engines. If the query is excessively restrictive (e.g. full name + email + phone + ...) the search engines are likely to return a null set, to avoid this behavior, a less restrictive query is posed (e.g. full name + email) and then use all the information available to rank the results (name, email, phone, address, etc.)

Besides that, it is important to mention that different queries are built for the different search engines, since their behavior are different. For instance, in UIUC Web Search a less restrictive query (e.g. full name) may be used, since the name conflicts are less frequent, whereas in Google, a more restrictive query will be used (e.g. full name + email) since, name conflicts are likely to occur.

The ranking schema is considerably slow, which turns it into a limitation factor. In order to avoid this slowness, two procedures are adopted. First the number of results retrieved from the search engines is limited. Second, the results of the queries are saved to a local database, dramatically increasing the speed for repeated queries.

Architecture

Within the given scenario, the system architecture is created in order to integrate all the search engines, and correctly route the user query.

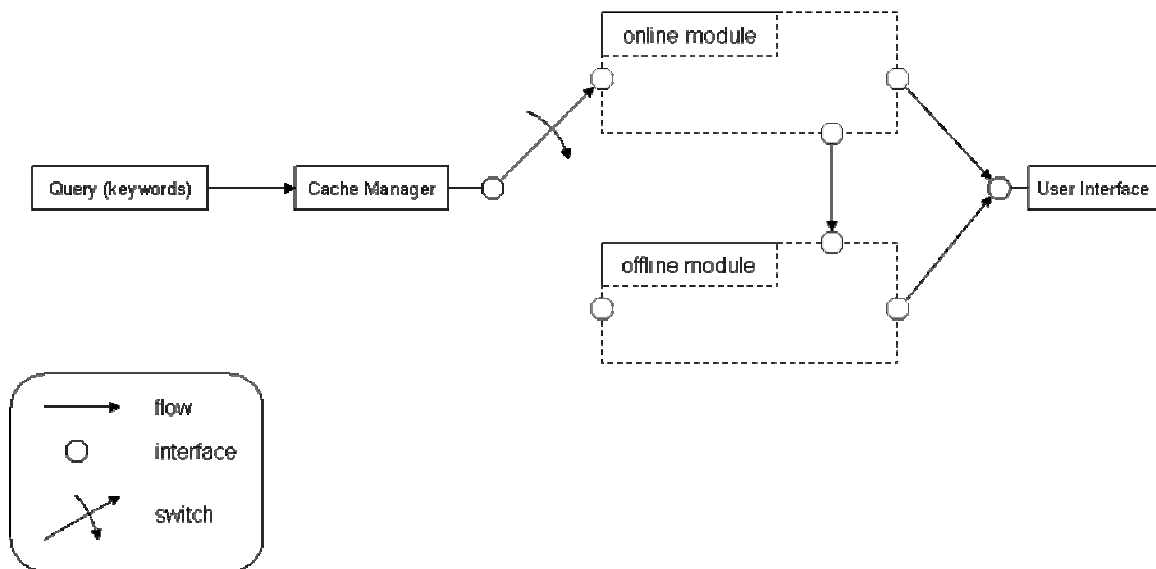


Figure 6

Figure 6 presents the data flow architecture of the UIUC People Finder. It is possible no notice that there is a switch which will define which module is used to retrieve the results. Switching to the online module will cause the application to fetch all the results from the internet, while switching to the offline module, the application will only query the database for the results. When the online module is used, it sends its results to the offline module, which in turns persist to the database, creating a cache.

The sequence of the tasks is as follows:

1. User type the query (e.g. name)
2. The cache manager checks the database, trying to find the entity.

- a. If the entity is found

All the information is retrieved from the offline module

Go to step 3.

- b. If the entity is not found

The query is passed to the Phonebook, and the structured information is retrieved.

For each search engine:

The information acquired in the previous step is used to build a query.

The built query is executed and the results are retrieved

The links (webpages) are ranked

The entries are persisted to the database

Go to step 3.

3. The results are sent to the user interface.

Organization

Cache Manager

The cache manager is the entry point of the system, it receives the query, then checks whether the query may be answered using the database or not. If it is possible, then it switches to the offline module. If it is not possible to answer the query with the information available in the database, it switches to the online module, which will access the internet and retrieve the necessary information.

The internal organization of the online and offline modules, are presented in Figure 7.

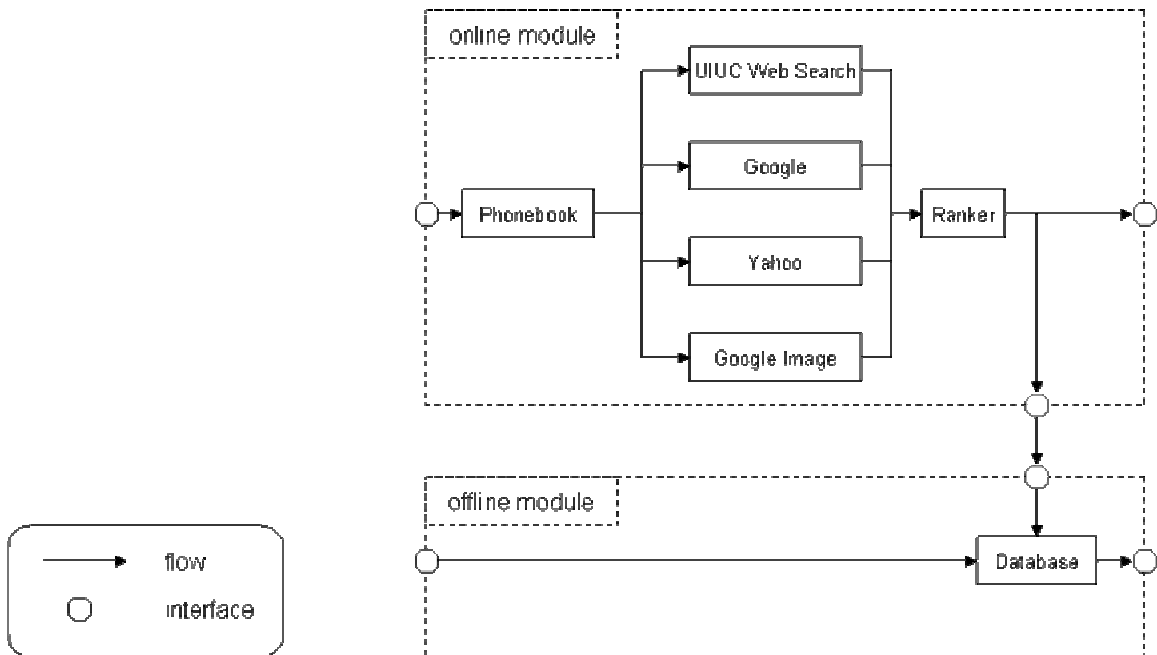


Figure 7

The online module consists of basically three important components: Searchers, Ranker and the Phonebook.

Searchers

They encapsulate the logic necessary to access a given web search engine and parse its results, there are four searchers: UIUC, Google, Yahoo, Google Images. The first three are common web search engines, they receive a text as a query, and return the links found. The last is an image search engine, it receives a text as query, and return the links from the images found.

As the web search engines do not expose a well defined interface to execute the queries, it is necessary to wrap the web search engines, in order to provide an interface to the application to interact with them. This is done, by converting the text queries to GET methods from HTTP protocol, by analyzing the URL used to execute a manual query, it is possible to infer the parameters, and consequently, dynamically build the necessary URL. After receiving the response from the URL accessed, the results come in form of HTML, which also does not have a well defined interface, thus it is necessary to parse the results, in order to extract the relevant links, by searching in HTML for specific links, colors, fonts, etc.

With the previous two tasks accomplished the searchers expose a well defined interface, they receive a query as string, and return a list links. With this interface exposed, it is possible to smoothly integrate them with the rest of the application.

Ranker

It is the component that access and rank the webpages retrieved by the searchers. Its main responsibility is to analyze the retrieved webpages, and using the algorithm described in the section Method. After ranking the webpages, they are persisted to the database, using the offline module.

Phonebook

It accesses the Phonebook, and parses its results, which identifies an entity. After receiving a query from the user, the Phonebook sub-module, will send the query to the Phonebook webpage. After receiving the answer, in the form of a HTML page, the information about the entity is extracted, and a model of the entity is created. The Phonebook sub-module, added to the Phonebook webpage act as a bridge between the UIUC database and the application. Both components, together with the database form the structured information source, used by the application.

The offline module has only one important component, which is the Database sub-module.

Database

This sub-module, act as the bridge between the application and the database. The database behind the application has one main purpose, provide cache. As stated before, the online module is excessively slow, thus, a cache is created to store the queries results, the option made, is to store them in a database. Besides this immediate use, the offline module has an additional advantage, it increases the system availability. The online module, depends on five online services, despite the high availability of them individually, a fail in one of them may produce inaccurate results as final answers, thus having an additional path, will allow the system to switch to the offline module, when a fail in the online module is detected.

Finally, there is the user interface sub-module.

User interface

The results produced by the previous modules are generated and saved in a specific file format, thus each query will produce a new file. This file is then read by the user interface sub-module, which parses the file, extracting the information acquired. With this information, a personal webpage for the entity is dynamically created, and sent to the user. The generated webpage consists of three parts, the entity information, pictures and links. The entity information is merely a copy of the content in the Phonebook webpage. The pictures are the ones retrieved by Google Image. Finally the links, are the ones retrieved from UIUC Web Search, Google and Yahoo, they are ordered by the ranking.

Implementation

The main application was developed in Java, using MySQL as a database. The system was developed using an object-oriented design. The front end of the application was created by combining some CSS style sheet programming and Perl scripting language programming.

The Perl scripting generates a command line call to execute the java program in the background based on the search criteria entered in the search field. The results are stored locally into an array data structure. The output generated is a text file which has delaminated tags that identifies the data blocks (entity information, pictures and links). This output interface allows for integration into numerous different sources.

The user interface script will fetch the pictures based on the image URL's generated by the program, and place them on the page for viewing. Also, the links associated with the queries are generated as active, live links in order of highest rank, which is determined by the program as well. The information retrieved from the original UIUC People Finder (Phonebook webpage) is also outputted to the user.

The limitation to this approach is the length of time it takes for the program to access all of the web pages retrieved from the search engines and run the ranking algorithm on them. When a query is executed it is stored in the database, thus if the query has already been executed it is retrieved from the cache, however with a real large number of users this may not be helpful to the system performance, which would cause the pages to load slowly.

The high coupling with the UIUC domain allows for eventual possible integration as an application to be used by the university.

Demonstration

Figure 8 shows the initial webpage, where the system expects for a query.



Figure 8

Figure 9 presents a sample output from the system, in the upper part, a picture of the entity shown, on the left, the information from the Phonebook, on the right, links related to the person.



Figure 9

Future work

An improvement in the usability could be achieved using dynamic webpage technologies, like AJAX, to periodically re-query the program based on who the current search term was for and re-output the portion of data related to the links and their rankings on the web page. This portion of the project was worked on but not fully completed.

Another enhancement would be to expand the scope of the Searchers, allowing the application to execute search queries across numerous search engines, specifically Google scholars, Google Blog, MySpace, etc. The wider array of search results, which are gathered, the more likely it becomes that the results are going to have pages with higher rankings. The full functionality of a larger set of search engines was not fully implemented.

The ranking system may also be improved, in order to reflect better the importance of the terms (name, phone, email, etc.). For instance, if a term that occurs frequently in the set, it should have a lower weight. Thus the weight would change dynamically and as opposed to the fixed weighting system.

References

- [1] M. Sayyadian, A. Shakery, A. Doan and C. Zhai. Toward Entity Retrieval over Structured and Text Data. In *WIRD'04*, 2004.