EMAIL AND PHONE NUMBER ENTITY SEARCH AND RANKING

by

SHUANG HAO

B.A. Northeast Agricultural University, 2003

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2008

Approved by:

Major Professor
William H. Hsu

Abstract

Entity search has been proposed as a search method for domain-specific Internet applications. It differs from the classical approaches used by search engines which give a “page-view result”: listing the URLs of web pages containing the desired keywords. Entity search returns more structured results listing the specific information that a user seeks, such as an email address or a phone number. It not only provides the URL links to targets, but also attributes of target entities (*e.g.*, email address, phone number, *etc*.). Compared to classical search methods, entity search is a more direct and user-friendly method for searching through a large volume of web documents. After the user submits a query, the extracted entities are ordered by their relevance to the query. While previous work has proposed various complex formulas for entity ranking, it has not been shown whether such complexity is needed. In this research I explore the problem of whether a simpler method can achieve reasonable results. I have designed an entity-search and ranking algorithm using a formula that simply combines a page’s *PageRank* and an entity’s distance to the query keywords to produce a metric for ranking discovered entities. My research goal is to answer the question of whether effective entity ranking can be performed by an algorithm that computes matching scores specific to the entity search domain, and what improvements are necessary to refine the result. My approach takes into account the entity’s proximity to the keywords in the query as well as the quality of the page where it is contained. I implemented a system based on the algorithm and perform experiments to show that in most cases the result is consistent with the user’s desired outcome.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Thanks to Dr. Hsu and the members of the KDD group at Kansas State University. I also appreciate Drs. Caragea and Amtoft for agreeing to serve on my committee and providing a great deal of valuable feedback on my work.

Especially, I would like to thank my husband, for his love and support throughout my Master's study. Last but not least, I would like to thank my parents, little brother and my uncle's family for their infinite love and care.

Dedication

To my husband and parents.

CHAPTER 1 - Introduction

The Internet has become important in daily life, and as a result Internet search has never played a more significant role. It is crucial for Internet users to obtain the desired information in an efficient and direct manner. Currently, the most popular way to search for information is using Google's search engine. Google, like many other search engines, adopts a "page-view" method, where the search results are simply a list of URL to web pages. Users must click through links to dig out the information they need. Obviously, this is not the best possible way to find for useful information from the entire Internet. Ideally, the search engine shall return answers directly related to a user's query, not just links to pages that may contain the answers.

Consider a daily search, for example: a student wants to find his supervisor John Smith's email. In current search engines, he will first type in the professor's name, *e.g.*, "John Smith". Second, from the given results, he needs to click to open the professor's homepage (if any). Third, find the email address.

Consider another example: suppose a resident wants to find Manhattan Fire Service's phone number. Again, he needs to type in "Manhattan Fire Service" in the searching engine first. Then he will click the most likely page link to open Manhattan Fire Service's home page. Then he must browse the whole page to find "Contact us" or "Please call" to find the phone number.

Through these real world problems, we can see that the standard page view search engine does not work well enough. We want a more efficient and reliable search method. In this work, we seek to explore more efficient methods for Internet search, through the study of *entity search* [2][3]. An entity is normally the object of interest during a web search. For instance, emails and phone numbers could be an entity in search queries like "John Smith email" and "Manhattan fire department phone number". The desired behavior of entity search is that after typing in the search keyword ("John Smith" in our previous example) and entity name ("email"), the results returned will be the email address of John Smith, rather than just hyperlinks to pages that contain the keywords "John", "Smith", and "email". In a nutshell, for entity search, the query consists of two parts: keywords such as those in typical web search, and an entity that has pre-defined special meanings for the search process. Since entity search is based on the page-view model, in the next section we briefly explain the model and give a mathematical formulation of entity search in the subsequent section.

The advantage of entity search is more significant when some information the user looks for is "hidden" for a variety of reasons. For instance, some companies do not put their customer-service phone number at conspicuous locations on their web sites. Thus, when the Internet searcher wants to find the number, it is often tedious to click through a large number of pages on the company's web site. However, this desired phone number might be contained in other pages not related to the company; for example online forums where customers share such information. Users usually do not take this as the first choice; instead, they browse web pages that do not have the desired information. Thus, users will spend lots of time and energy on "hopeless" pages -- the pages which do not have the desired information, before they finally find it or give up instead. Entity search can help in this situation, because the algorithm takes into account the entity information and as a result can return information directly related to a user query.

Overall, entity search appears to be a reasonable solution to bridge the "semantic gap" between what a user wants and what is expressible in pure keyword-based search queries. Accounting for the special meanings in certain search terms (entities), and returning the direct and apparent results the user seeks, can potentially save time improving the web-search experience. Entity search cannot replace classical web search. Moreover, in many cases, the user also needs the entire web page in addition to the entity records. In such cases it is easy to provide a link to the whole page along with the entity search result.

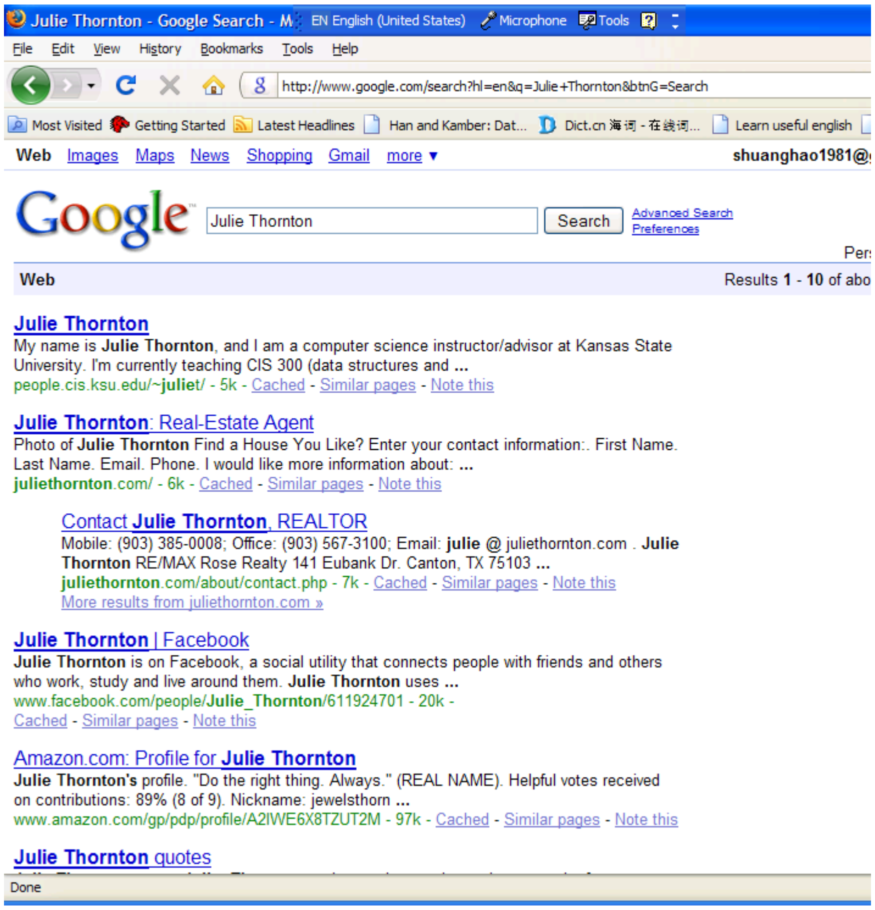
1.1 Page view model

Let S be a “page-view” model search algorithm. The input to the algorithm consists of two parts: P (for pages) and Q (for query). The Internet can be considered a collection of web pages, with link structure included in the web-page documents. Thus, we have $P = \{d_{1\,,}\ d_{2\,,}\ \ldots\,,d_m\ \}$, where d_i is a web-page document. The query is a question that describes what the user wants to find out. In the page-view model, it is just a collection of as keywords which can be interpreted the question “what are the pages that contain the information regarding these keywords?” Thus we have $Q = \left\{\,k_{1\,,}\,k_{2\,,}\ \ldots\,,\ k_p\,\right\}$, where k_i is a single keyword.

The output for a page-view search algorithm is a list of web pages that contain the query keywords ordered by the rank of the page. The rank typically represents the quality of the page related to the query. We use O to denote the output and we have $O = \left\{\,l_{1\,,}\,l_{2\,,}\ \ldots\,,\ l_n\,\right\}$, where each l_i is a link to a web page.

Figure 1.1 shows an example of the page view model in the context of the Google search engine.

Figure 1.1 Page View Example



Julie Thornton
My name is Julie Thornton, and I am a computer science instructor/advisor at Kansas State University. I'm currently teaching CIS 300 (data structures and ...
people.cis.ksu.edu/~juliet/ - 5k - Cached - Similar pages - Note this

Julie Thornton: Real-Estate Agent
Photo of Julie Thornton Find a House You Like? Enter your contact information:. First Name. Last Name. Email. Phone. I would like more information about: ...
juliethornton.com/ - 6k - Cached - Similar pages - Note this

Contact Julie Thornton, REALTOR
Mobile: (903) 385-0008; Office: (903) 567-3100; Email: julie @ juliethornton.com . Julie Thornton RE/MAX Rose Realty 141 Eubank Dr. Canton, TX 75103 ...
juliethornton.com/about/contact.php - 7k - Cached - Similar pages - Note this
More results from juliethornton.com »

Julie Thornton | Facebook
Julie Thornton is on Facebook, a social utility that connects people with friends and others who work, study and live around them. Julie Thornton uses ...
www.facebook.com/people/Julie_Thornton/611924701 - 20k -
Cached - Similar pages - Note this

Amazon.com: Profile for Julie Thornton
Julie Thornton's profile. "Do the right thing. Always." (REAL NAME). Helpful votes received on contributions: 89% (8 of 9). Nickname: jewelstorn ...
www.amazon.com/gp/pdp/profile/A2IWE6X8TZUT2M - 97k - Cached - Similar pages - Note this

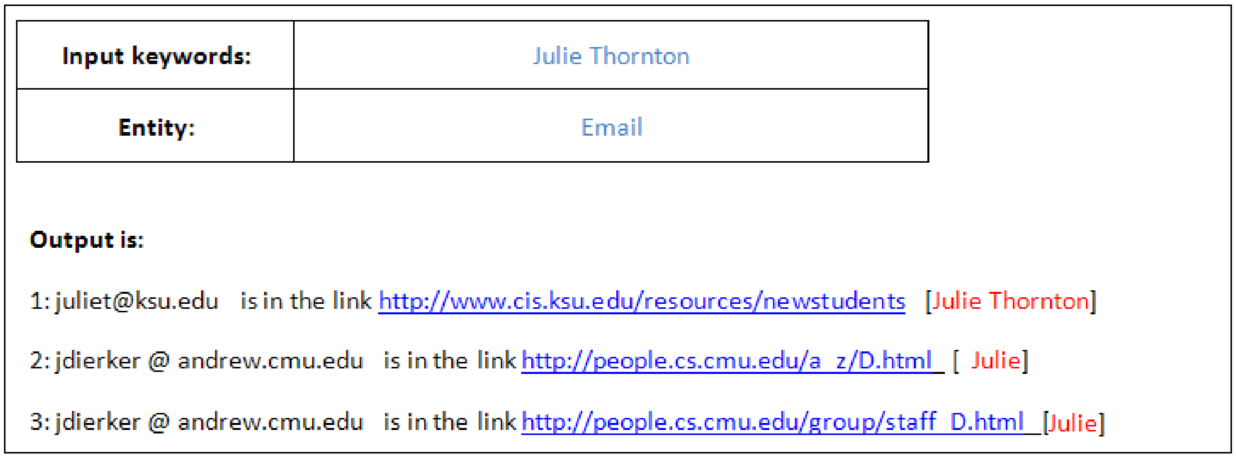
Julie Thornton quotes

1.2 Entity search model

Let ES be an entity search algorithm. As with the page-view model S, there are two parts in S's input: pages P and query Q. The query, however, consists of two parts in an entity search— the entity that the user wants to get, and the keyword that he wants the search to return. From the previous example, the keyword is "John Smith" (or "Manhattan Fire Service") and the entity name is "email" (or "phone number"). These entities are pre-defined by the search algorithm and represent common questions people want to query for in web searching. Formally, we have

Q = < E , K > , where E is a set of entities names and K is a set of keywords. The results returned by ES are the entities that satisfy the search criteria and the links to the web pages where they are contained. As with the page-view model, the results are also sorted by their quality and relevance. Thus we have the output $O = \{ < E_1 , L_1 >, < E_2 , L_2 >, \ldots , < E_n , L_n > \}$, where E_i is an entity and L_i is the link to its page. Note that every occurrence of an entity is counted as a distinct entity, even if they have the same value. Figure 1.2 shows an example of entity search.

Figure 1.2 Entity Search Example



Input keywords:	Julie Thornton
Entity:	Email

Output is:

1: juliet@ksu.edu is in the link http://www.cis.ksu.edu/resources/newstudents [Julie Thornton]

2: jdierker @ andrew.cmu.edu is in the link http://people.cs.cmu.edu/a_z/D.html [Julie]

3: jdierker @ andrew.cmu.edu is in the link http://people.cs.cmu.edu/group/staff_D.html [Julie]

1.3 Objectives

There have been a number of other efforts in the entity search area [2][3][7]. We review those methods in Chapter 2. The goal of this research is to investigate whether simple and intuitive ranking methods can yield an efficient and effective entity search system. A review of the literature found that there has not been much work that studies whether a simple combination of entity matching algorithm and standard web ranking techniques (i.e. Google *PageRank* [8]) can yield reasonably good results on the Web. As Einstein once put it: “Everything should be made as simple as possible, but no simpler.” [11] My research strives to answer the question of how simple an entity search algorithm could be, and what limitations the simple algorithm has, which may shed light on the right direction of future research in the research on ranking measures.

Regarding convenience, as for other searching engines, we would like the most likely results to be on the top, followed by the less likely ones. I used the unmodified Google *PageRank* algorithm to calculate the rank of each page based on a random walk model [8]. This rank is a factor in my overall algorithm that ranks entities after extracting them from the various pages. How to rank entities is a core challenge in this work. I designed an algorithm that combines a number of important metrics that affect the quality of a discovered entity instance, and produces an overall metric that can be used to rank the entity instances. I introduce the algorithm in Chapter3, and the experiments on the simple algorithm are conducted with results presented and analyzed in Chapter 4.

CHAPTER 2 - Related Work

Entity search and ranking are part of a relatively new research area which has been explored in a number of research efforts. One of them is that of Vercoustre *et al.*, which focuses on entity ranking in *Wikipedia* [7]. They proposed an approach for extracting entities from *Wikipedia*'s XML documentations, along with a ranking measure. Vercoustre *et al.* focus only on retrieving information from *Wikipedia*, and the ranking measure is therefore restricted in this domain as well. Both Pechcevski's work and my research have the same goal: entity search and ranking; however, my research is to address the entity search problem over the entire web, as opposed to just *Wikipedia*.

Entity search, in some sense, is a special case of the *question answering* (QA) problem [15]. In the traditional web search, the user enters a sequence of keywords, and the results are just the pages most relevant to the query, *i.e.* the pages that contain those keywords. In the entity search, however, the user has already predefined data that he would like to know, for example, the phone number or the email address. Based on this, the system has a better understanding on what the user is looking for, and hence can potentially serve the user with more satisfactory results. Entity search is different from the general QA problem; entity search cannot answer arbitrary questions. The type of question that it can answer is just "what is the entity most relevant to the user's keyword?"

Another relevant research effort was conducted by the Database and Information Systems Laboratory (DAIS) of Computer Science Department at University of Illinois at Urbana-Champaign. Cheng *et al.* constructed a model to this entity search and ranking problem [2][3]. Compared to Vercoustre's work, their model is more widely applicable. Their model is divided into three main layers: a global access layer, a local recognition layer, and a validation layer. The ranking measure is calculated based on these three layers. The ranking measure in the global access layer is the possibility of a document being selected. The local recognition layer measure is the probability that the query can be true in a given document. The validation layer measure is the probability of the query over the whole virtual model (the whole collection of documents).

Although my research also addresses the entity ranking problem over the whole Internet, there are several differences between my work and Cheng's. The first and most important is that the ranking principle is different. Cheng's work proposes an approach for entity search and ranking with a rather complex scheme. As mentioned above, it divides the ranking process into three layers, each with several parametric formulas, and emphasized the refinement and empirical tuning of these parameters. The goal of my research is to investigate if good results in a more specific domain can be achieved using a simpler model. It is not clear from previous works whether a simpler and more straightforward method can produce reasonable entity search result. In this research, I explore a simple scheme for entity ranking over the whole Internet and assess the effectiveness and potential limitations of the approach.

CHAPTER 3 - Methodology

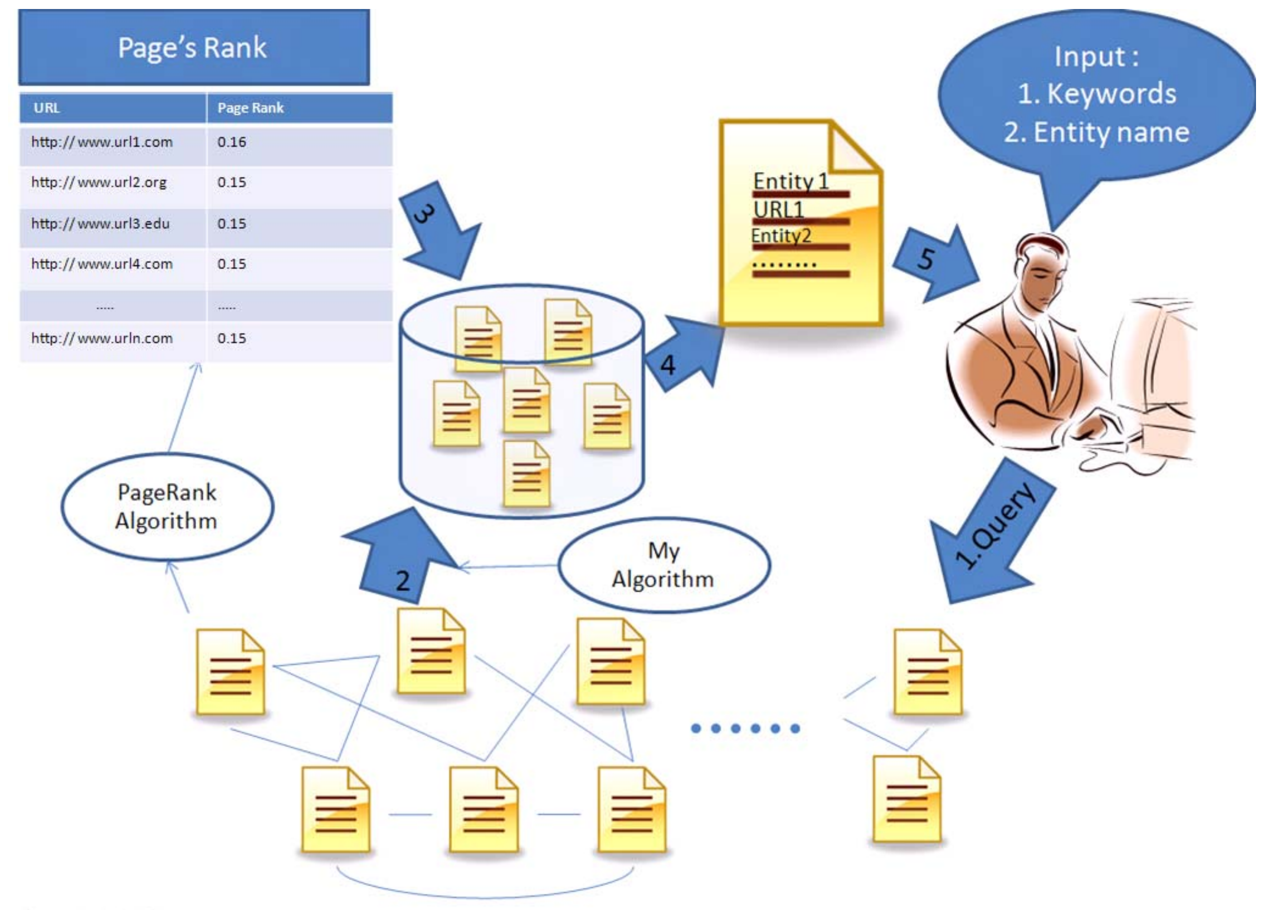
3.1 Overview

The Internet can be viewed as a huge corpus of information organized in pages, each of which has its own unique URL (Uniform Resource Locator). A URL is a special type of URI (Uniform Resource Identifier), a compact string of characters used to identify or name a resource on the Internet. [5] URLs also specify where the identified resource is available and the protocol for retrieving it. [4] As discussed in Chapter 1, each page contains keywords and entities of interest. Keywords are just strings that match user search queries. Entities are contents of a page with pre-defined semantics in the query, such as email address and phone numbers. Links exist between pages. Using an algorithm such as Google's *PageRank* [8], each page can be assigned a rank representing its importance (discussed in Section 3.3). The user enters a string that may contain the names of entities (*e.g.* phone or email address) and keywords that he indicates as his interest. The system will apply the query (step 1 in the Figure 3.1 below) across the entire Internet and return a listing of all the pages that satisfy the query (step 2 in the Figure 3.1). Pages' information retrieved from this search, including the query and some of the frequent and highly ranked hits is stored in a database for future use.

We combine three metrics to calculate the final rank of an entity hit during the search: Google *PageRank* [8] (step 3 in the Figure 3.1) of the page containing the entity, a line rank that indicates the relevance of a line with respect to the search keywords, and a distance between a hit entity and a hit keyword (step 2 in the Figure 3.1). These metrics will be addressed in section 3.3. The overall rank of the entity is returned along with the entity name and its URL (step 4 in the

Figure 3.1). The list of entities is then ordered according to the entity ranking measure and shown to the user (step 5 in the Figure 3.1).

Figure 3.1 System Architecture



URL	Page Rank
http:// www.url1.com	0.16
http:// www.url2.org	0.15
http:// www.url3.edu	0.15
http:// www.url4.com	0.15
.....	.....
http:// www.urln.com	0.15

3.2 Data preparation

3.2.1 Heritrix

We use the *Heritrix* [6] web crawler to collect web pages. *Heritrix* is an open-source, Java-based web archiving crawler developed by Internet Archive and Nordic National Libraries beginning in 2003. The current version is 2.0.1. My project used version 1.14.1, which was the newest when I started my project and is currently the second newest. *Heritrix*'s main user interface is accessed from a web browser. Once the user launches *Heritrix* from the command line, he can access it through a local port via the HTTP protocol. After login the console page will display a user interface providing access to monitoring and management capabilities. One can also configure a new job's various setting through the web interface. I will explain my settings in the following sections.

3.2.1.1 Specification

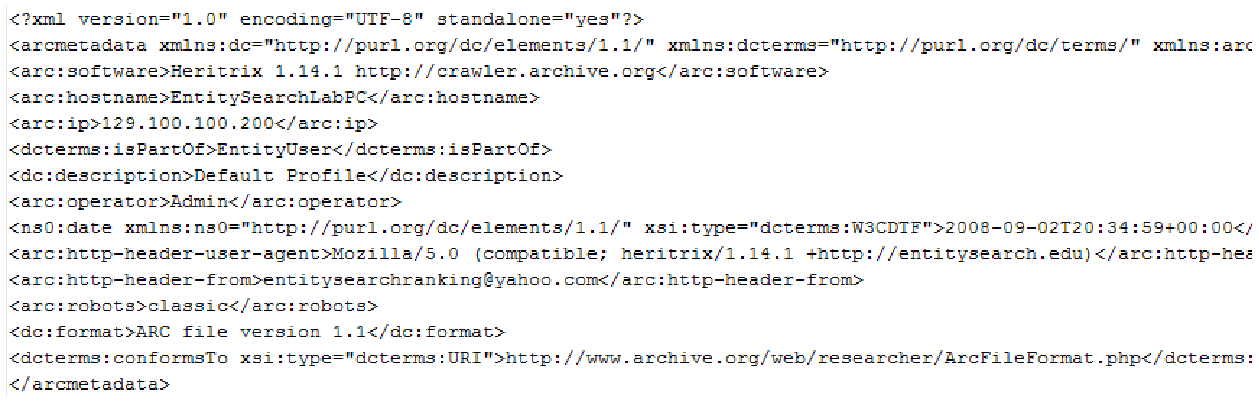
Starting from a pre-defined set of URLs (crawling seeds), *Heritrix* crawls the World Wide Web automatically and methodically. I configured *Heritrix* to restrict the crawled files to only HTML.

3.2.1.2 ARC files

The files that *Heritrix* crawls are saved as internet archive ARC files. Then each ARC file is compressed using `gzip`. Each compressed ARC file corresponds to a volume containing many HTML files, but the entire crawl is divided among many ARC files. This process provides some convenience for seeking some records randomly, because it takes less space and time to unzip a divided small ARC file than a single huge ARC file.

At the beginning of each ARC file is an XML metadata record. Figure 3.2 shows an example metadata record for an ARC file.

Figure 3.2 An Example of ARC File Metadata Record Body



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<arcmetadata xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/" xmlns:arc
<arc:software>Heritrix 1.14.1 http://crawler.archive.org</arc:software>
<arc:hostname>EntitySearchLabPC</arc:hostname>
<arc:ip>129.100.100.200</arc:ip>
<dcterms:isPartOf>EntityUser</dcterms:isPartOf>
<dc:description>Default Profile</dc:description>
<arc:operator>Admin</arc:operator>
<ns0:date xmlns:ns0="http://purl.org/dc/elements/1.1/" xsi:type="dcterms:W3CDTF">2008-09-02T20:34:59+00:00</
<arc:http-header-user-agent>Mozilla/5.0 (compatible; heritrix/1.14.1 +http://entitysearch.edu)</arc:http-hea
<arc:http-header-from>entitysearchranking@yahoo.com</arc:http-header-from>
<arc:robots>classic</arc:robots>
<dc:format>ARC file version 1.1</dc:format>
<dcterms:conformsTo xsi:type="dcterms:URI">http://www.archive.org/web/researcher/ArcFileFormat.php</dcterms:
</arcmetadata>
```

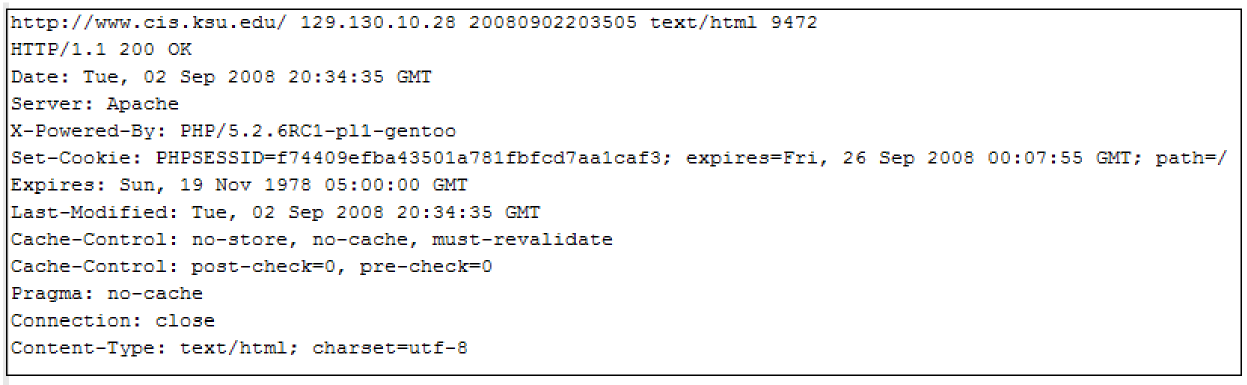
3.2.2 Crawling seeds

Crawling seeds are a set of initial URLs from which a crawl proceeds by following hyperlinks found in each crawled page. This process proceeds recursively, terminating only when all reachable pages have been crawled. Crawlers often allow the user to specify limits on the depth and number of pages returned by a crawl. The crawls I performed in this research had no such specified limits; however, for my research, I chose to focus on the information about faculty in computer science departments, and the crawling seeds are mostly URLs of computer science departments at U.S. universities. Besides, in order to show that my research is suitable for any area, some other kinds of pages' URLs are also includes. Appendix A includes the whole list.

3.3 Keyword and entity recognition

Figure 3.3 shows a portion of an ARC file after being unzipped. A single ARC file may contain a large number of HTML pages. We use the pattern "HTTP/1.1" as the delimiter of HTML pages.

Figure 3.3 Example of the beginning of a New HTML Record in an ARC file



```
http://www.cis.ksu.edu/ 129.130.10.28 20080902203505 text/html 9472
HTTP/1.1 200 OK
Date: Tue, 02 Sep 2008 20:34:35 GMT
Server: Apache
X-Powered-By: PHP/5.2.6RC1-pl1-gentoo
Set-Cookie: PHPSESSID=f74409efba43501a781fbfcd7aa1caf3; expires=Fri, 26 Sep 2008 00:07:55 GMT; path=/
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Last-Modified: Tue, 02 Sep 2008 20:34:35 GMT
Cache-Control: no-store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=utf-8
```

3.3.1 Keyword search

The keyword search is implemented using the standard API provided by Java. Keyword search considers the various possible user typing manners. For example, case of letters does not affect the result of search. I also implemented string matching to make sure that the word hit in the searched line is an independent word and not a part of other words.

3.3.2 Email entity search

Retrieving email entities from web pages can be tricky. Due to the escalating spam problem, more and more people are starting to obscure their email addresses online. For example, instead of bhsu@cis.ksu.edu, the user may put bhsu [AT] cis.ksu.edu on his web page. Some users even go further and put an image of the email address instead of textual email. Since in some sense a search engine is acting similarly to a robot used by spammer, any mechanism that prevents the spammer from collecting email information will also work against email entity search. As a result, I do not address this problem in my research. To improve the accuracy of my system; however, a number of common patterns are compared against to find lightly obscured email entities such as the one above.

3.3.3 Phone-number entity search

There are a number of patterns in which people write phone numbers. For example: (123)456-7890, (123)-456-7890, (123) 456-7890, 123 456 7890, (123) 456 7890, 123-456-7890, (123)4567890, 123-4567890, 123.456.7890, 123 4567890, or (123) 4567890 and so on. I used a regular expression to specify all such formats so that the program can extract phone numbers written therein. When search results are returned back to the users, those phone numbers are converted to a standard format [1]:

“+” + international prefix + country code + area code + phone number

3.4 Algorithms

Once the data have been downloaded, a strategy for efficiently retrieving the entities is needed. In this section, I document the ranking algorithm I developed for this purpose. It consists of four components: *PageRank* [8], LineRank, Entity-PageRank and finally, EntityRank.

3.4.1 PageRank

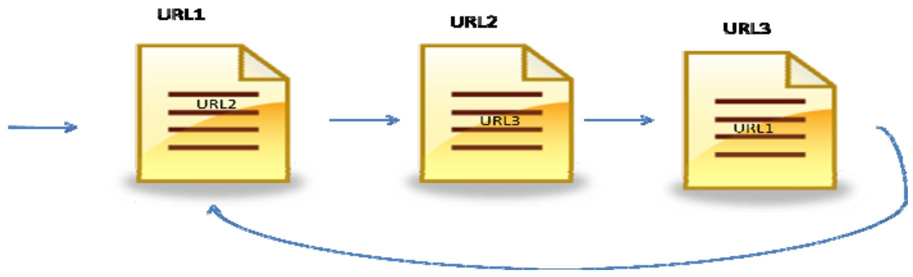
Web pages have various aspects of "quality", each contributing to the overall quality of an entity named in a web page. I use Google's *PageRank* [8] algorithm to provide the metric for a page's quality. The definition of *PageRank* is provided below.

A page may have some links to other pages and may be linked from other pages. Let all the pages that a page p links to be the page's *out-links*, denoted F_p, and all the pages that link to it be the page's *in-links*, denoted B_p. The total number of out-links is $|F_p|$, and the total number of in-links is $|B_p|$. Let $N_p = |F_p|$. The rank of a page p is defined as the solution to the following linear system.

$$R(p) = c\sum_{u\in B_p}\frac{R(u)}{N_u} + (1-c)E(p) \qquad [8]$$

c is a "damping factor" used to prevent entrapment in page sinks, i.e. cycles of links. In my work, I set $c = 0.85$ which is the default value for Google *PageRank* [8]. The purpose of having the damping factor is to alleviate the rank aggregation effect of sink nodes and cycles, which may skew the ranks. The interpretation of the damping factor is that a "random surfer" may "get bored", stop surfing, and jump to a random page based on the distribution specified by *E(p)*, which is sometimes called the "initial vector" or "personalization vector". In my work, as in the classical Google *PageRank* [8], all pages' initial value is set to 1, *i.e.*, *E* is a vector of all ones. I normalize the total ranks of all nodes to be N, the number of total pages in the corpus. Figure 3.4 illustrates the "sink node" or "cycle" problem in *PageRank* [8].

Figure 3.4 Page Sink



Such cycles will disproportionately accumulate ranks in computation and may make the Jacobi method of computing the solution to the above equation fail to converge. It can be proved that when the damping factor is less than one, a solution to the above linear system always exists and the Jacobi method always converges to the solution [10]. I implemented the Jacobi method for the *PageRank* [8] algorithm. The convergence condition is determined when the difference between two contiguous rank vectors fall within a pre-set small value.

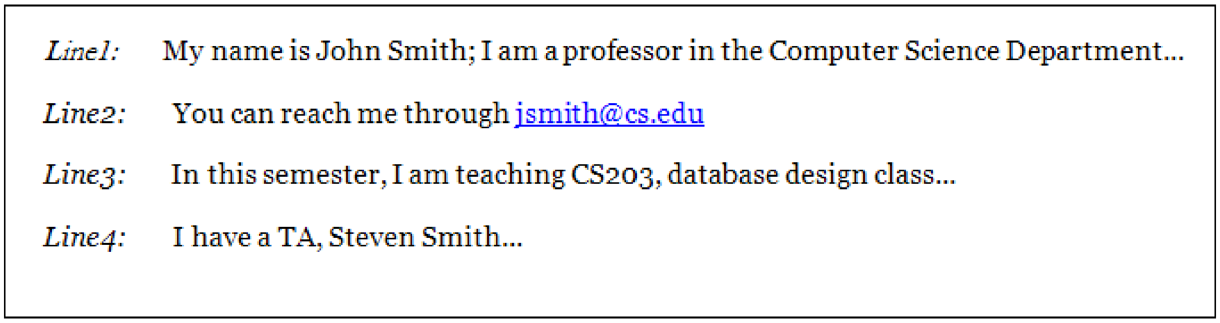
3.4.2 LineRank

A web page consists of multiple lines of text. Each line may contain zero or more keywords supplied by the user. Because the user wants each returned result to contain as many keywords of the query as possible, we introduce LineRank as a measure of the relevance of a line to the user query with respect to the input keywords. Let *n* be the number of distinct keywords that are "hit" by the line L, and k be the total number of distinct keywords the user entered. The line rank (with respect to the query) is defined as follows.

$$R_l(L) = \frac{n}{k}$$

For example, if the user wanted to search "John Smith's email", "John" and "Smith" are keywords; hence we have k =2. Suppose page p contains the following information:

Figure 3.5 Example Page



Line1: My name is John Smith; I am a professor in the Computer Science Department...

Line2: You can reach me through jsmith@cs.edu

Line3: In this semester, I am teaching CS203, database design class...

Line4: I have a TA, Steven Smith...

Based on our definition, we have

$$R_l(Line1) = \frac{2}{2} = 1$$

$$R_l(Line2) = \frac{0}{2} = 0$$

$$R_l(Line3) = \frac{0}{2} = 0$$

$$R_l(Line4) = \frac{1}{2} = 0.5$$

The higher the line rank, the more potential relevance the line has to the user's query. *Line 1* is more relevant to the query than other lines.

3.4.3 DistanceRank

We assume the closer an entity is to the query keywords, the more relevance the entity is to the query. Thus, we introduce DistanceRank to represent the distance between keywords and entities. It is an inverse function of the difference in the number of lines registering a hit for the keyword (L) versus for the entity (E). Intuitively, the closer an entity is to a keyword, the bigger the chance that the entity is "correct" within the search. The formula is:

$$R_d(L, E) = \frac{1}{|L - Line(E)| + 1}$$

where *L* is a line number and *Line(E)* is the line number of an entity. If the keyword and entity are in the same line, the distance rank for the entity and the line is 1. Consequently the distance rank will be the maximized in this case, indicating that the entity might be strongly connected with the keyword. In the previous example, the entity is an email address and occurs in *Line 2*. The keywords are in *Line 1* and *Line 4* respectively. Thus, the distance ranks of email, the following entity, occurring in two lines are (e is the only email entity jsmith@cs.edu in the example page):

$$R_d(Line1,\ \ e) = \frac{1}{|1-2|+1} = 0.5$$

$$R_d(Line2,\ \ e) = \frac{1}{|2-2|+1} = 1$$

$$R_d(Line3,\ \ e) = \frac{1}{|3-2|+1} = 0.5$$

$$R_d(Line4,\ \ e) = \frac{1}{|4-2|+1} = 0.333333$$

Note that *Line 2* has the highest score because it is the line where the entity is. *Line 1* and *Line 3* have the same DistanceRank because they have same distances between *Line 2*, where the entity is. *Line 4* has the lowest ranking. The DistanceRank and the LineRank are combined to measure an entity's overall relevance to a user query. This is the Entity-PageRank algorithm introduced below.

3.4.4 Entity-PageRank

The purpose of the Entity-PageRank algorithm is to calculate a numerical metric that captures an entity's relevance to a user's query within the context of the page where it is contained. It combines the DistanceRank of the entity with every line in the page that contains some keywords, and the LineRank of the corresponding page. The formula and algorithm for computing Entity-PageRank are illustrated in below.

$$R_{ep}(E) = \sum_{L \in P(E)} R_l(L) R_d(L, E)$$

Figure 3.6 Entity-PageRank Algorithm

```
Input: Entity E, Page P
Output: $R_{ep}$

For each entity E in page P
        $R_{ep} \leftarrow 0$
          For each line L in P
                $R_l \leftarrow \frac{\#keywords\ in\ L}{\#total\ keywords}$
                $R_d \leftarrow \frac{1}{|L - Line(E)| + 1}$
                $R_{ep} \leftarrow R_{ep} + R_l R_d$
          End
End
```

3.4.5 EntityRank

Entity-PageRank indicates an entity’s potential relevance to a user query within a single page’s context. Since the pages themselves have various levels of quality, an entity’s quality will also be affected by the quality of the page where it is contained. The EntityRank is the final rank computed by my algorithm, which is the product of the two ranking measures mentioned above: Entity-PageRank and *PageRank* [8].

$$R(e) = R_{ep}\big(E, page(e)\big)R(page(e))$$

This metric reflects the intuition that the global rank of an entity/page pair should be directly proportional to both the entity's rank on the page and the page's rank in the entire web.

3.5 Result Post-processing

While a database server (*e.g.*, *MySQL* or *Oracle*) could be used to store the ranking result, I chose to use *MS Access 2007* to store and sort the rankings. Since this research focuses on the ranking algorithm rather than constructing a production quality system, the database is only an assistant tool. Furthermore, my implementation uses Open Database Connectivity (ODBC) [13] interface to communicate with the database. Thus any database server that supports ODBC could be used in this project.

CHAPTER 4 - Experiment and Setup

In order to see how well my EntityRank algorithm works, I conducted several experiments to study whether the ranking output by EntityRank corresponds to the subjective measure of quality and ordinal rank a human user would assign to a discovered entity.

The experiments were conducted in the following manner. I used *Heritrix*[6] to download a corpus of web pages of about 20GB. I ran the EntityRank algorithm on this corpus with a number of queries. The resulting entities and EntityRank scores were then put into an MS Access database along with each page's URL and other relevant information. I then sorted all the entities from each query based on the EntityRank score. The results are analyzed to confirm whether it is consistent with the quality of the identified entities.

4.1 Search scope

This research only focuses on searching professors' emails and phone numbers in the computer science departments of U.S. universities. However, to show that this ranking measure can be applied to a wider setting, I also used a couple of crawling seeds that are government or businesses web sites. Some example organizations used in the crawler's seeds are: Kansas State University, Stanford University, Princeton University, Carnegie Mellon University, *etc.*

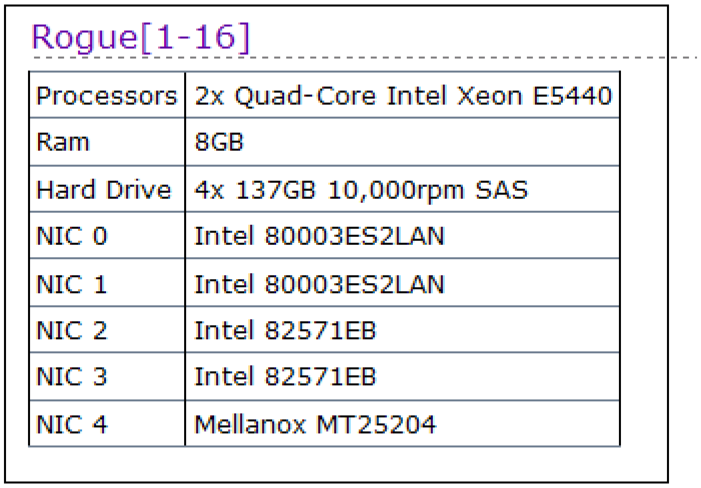
4.2 Crawler configuration

There are several configuration options that are set for the web crawler *Heritrix*[6]. As mentioned before, in order to eliminate unnecessary files such as GIF files, PDF files, *etc.*, I used *Heritrix*'s `MatchesFilePatternDecideRule` in the DecidingScope for the crawler's scope. This rule will filter out most but not all binary files that are not useful for this research. I also set “max-size-bytes” 50000000 Bytes, so opening such downloaded files will not be too time-consuming.

4.3 Machine configuration

The crawler and EntityRank program were both run on the *Beocat* [12] cluster in the Computing and Information Science (CIS) Department at Kansas State University. *Beocat* consists of 35 computing nodes containing 210 processors in total. The total amount of RAM is 600GB. The cluster systems run Gentoo Linux with a 2.6 kernel. My program was assigned to a node, whose specifications are shown in Figure 4.1.

Figure 4.1 Specifications of Node Rogue on Beocat System



Rogue[1-16]

Processors	2x Quad-Core Intel Xeon E5440
Ram	8GB
Hard Drive	4x 137GB 10,000rpm SAS
NIC 0	Intel 80003ES2LAN
NIC 1	Intel 80003ES2LAN
NIC 2	Intel 82571EB
NIC 3	Intel 82571EB
NIC 4	Mellanox MT25204

It took about one hour to crawl just above 20 gigabytes of data. However, the crawled compressed data by *Heritrix* [6] could not be read. I recompressed all of them. Therefore, the total data is slightly less than 20 gigabytes. A query specifies keywords (*e.g.*, “John Smith”) and an entity name (*e.g.*, “email”). Each query took approximately 10 minutes to retrieve the entities along with the ranking results. Note that most of the time was spent on searching for the entity, which will be greatly sped up if one indexes the entities that appear in a page during crawling. Since my research focuses on entity ranking, I did not implement the indexing of entities, which can be readily achieved using open-source tools such as *Nutch* [14].

The entities and ranking results along with keywords, URLs associated with them are stored in an *MS Access* database through the ODBC. The database that stores the ranking result and conducts the sorting was loaded on a Windows XP machine with Intel Core 2 CPU with 1.50 GHz, and 2GB memory.

CHAPTER 5 - Results and Analysis

Several queries were issued to test if my ranking algorithm worked as expected.

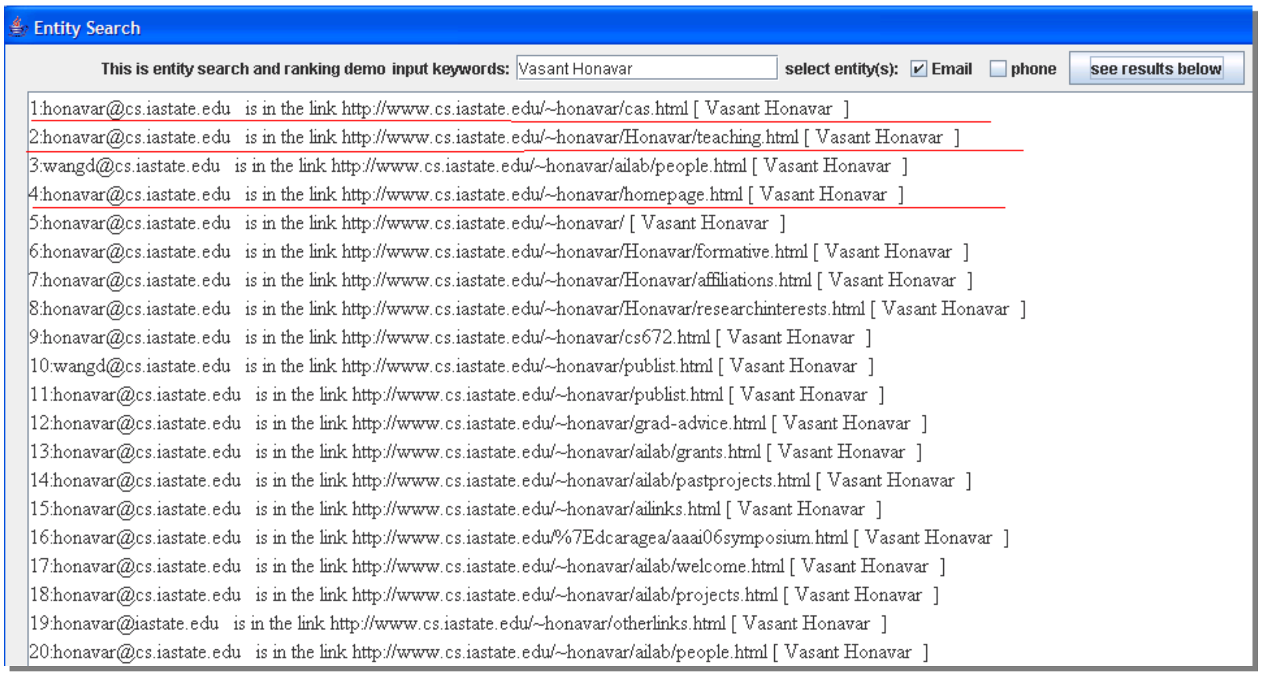
5.1 Email entity results

For a query using key phrase "Vasant Honavar" and entity name "email" to search for the address of a professor of computer science at Iowa State Univesity, the first 10 results of returned by my system are shown in Table 5.1. Figure 5.1 is the screenshot of the system's front-end which displays the top 20 results.

Table 5.1 Email Entity Results with Keyword "Vasant Honavar"

Rank	Entity	URL	Keywords
1	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/cas.html	Vasant Honavar
2	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/Honavar/teaching.html	Vasant Honavar
3	wangd@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/ailab/people.html	Vasant Honavar
4	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/homepage.html	Vasant Honavar
5	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/	Vasant Honavar
6	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/Honavar/formative.html	Vasant Honavar
7	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/Honavar/affiliations.html	Vasant Honavar
8	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/Honavar/researchinterests.html	Vasant Honavar
9	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/cs672.html	Vasant Honavar
10	wangd@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/publist.html	Vasant Honavar

Figure 5.1 Top 20 Email Entity Results with Keyword "Vasant Honavar"

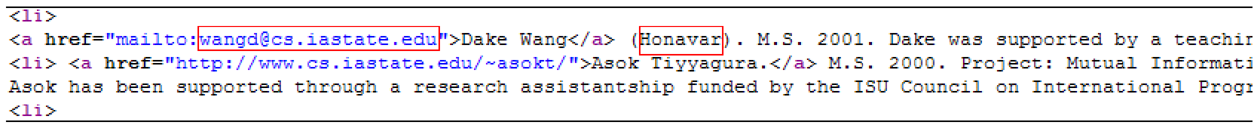


This is entity search and ranking demo input keywords: Vasant Honavar select entity(s): [✓ Email] phone

```
1:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/cas.html [ Vasant Honavar  ]
2:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/Honavar/teaching.html [ Vasant Honavar  ]
3:wangd@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailab/people.html [ Vasant Honavar  ]
4:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/homepage.html [ Vasant Honavar  ]
5:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ [ Vasant Honavar  ]
6:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/Honavar/formative.html [ Vasant Honavar  ]
7:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/Honavar/affiliations.html [ Vasant Honavar  ]
8:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/Honavar/researchinterests.html [ Vasant Honavar  ]
9:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/cs672.html [ Vasant Honavar  ]
10:wangd@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/publist.html [ Vasant Honavar  ]
11:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/publist.html [ Vasant Honavar  ]
12:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/grad-advice.html [ Vasant Honavar  ]
13:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailab/grants.html [ Vasant Honavar  ]
14:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailab/pastprojects.html [ Vasant Honavar  ]
15:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailinks.html [ Vasant Honavar  ]
16:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/%7Edcaragea/aaai06symposium.html [ Vasant Honavar  ]
17:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailab/welcome.html [ Vasant Honavar  ]
18:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailab/projects.html [ Vasant Honavar  ]
19:honavar@iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/otherlinks.html [ Vasant Honavar  ]
20:honavar@cs.iastate.edu   is in the link http://www.cs.iastate.edu/~honavar/ailab/people.html [ Vasant Honavar  ]
```

From the results, we can see that Dr. Vasant Honavar's email address "honavar@cs.iastate.edu" is ranked in the first place along with other top scores. "http://www.cs.iastate.edu/~honavar/cas.html" is not a personal page, but a group page. In these returned results, there is only one out of the top nine entities which is not Dr. Honavar's email address. It is an email address of Dr. Honavar's former M.S. student Dake Wang. The reason this hit ranks highly is that in the web page, Wang's email and "Honavar" occur in the same line, giving it the maximum DistanceRank. Figure 5.2 shows the details. In some sense, this can be viewed as a "false connection": an entity is physically close to the keyword but does not have a strong connection with the keyword. Since my ranking algorithm only considers physical proximity, such "false connections" cannot easily be identified. This identifies one potential weakness of the simple approach to entity ranking.

Figure 5.2 Page Source of http://www.cs.iastate.edu/~honavar/ailab/people.html



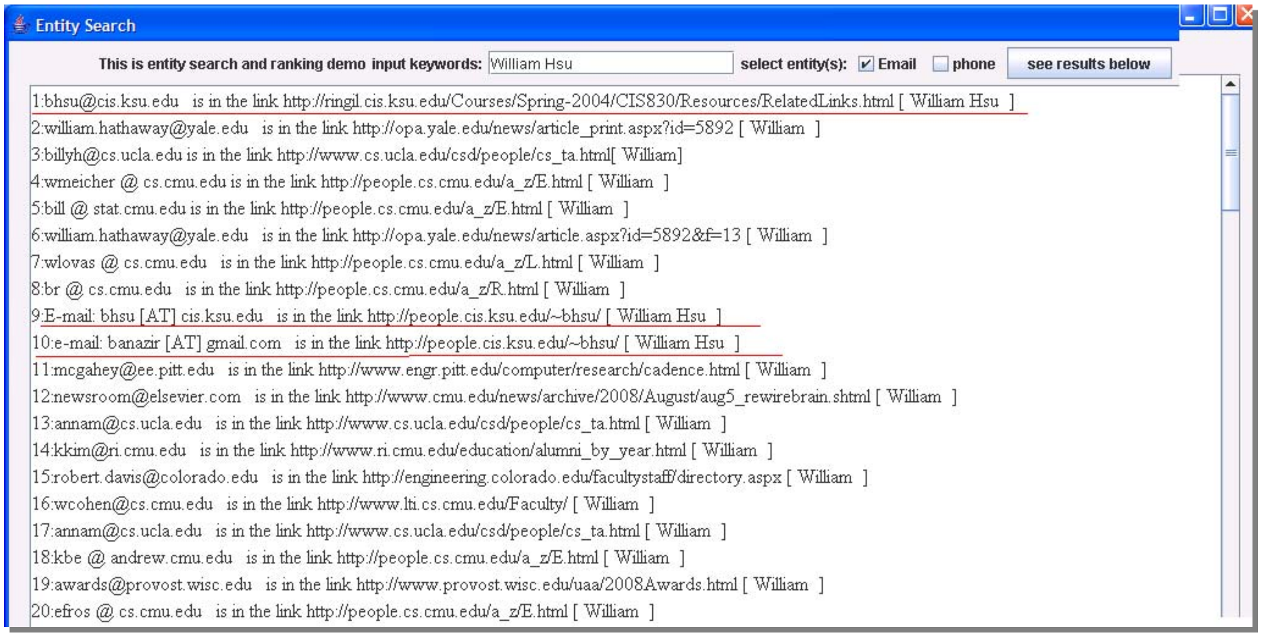
```
<li>
<a href="mailto:wangd@cs.iastate.edu">Dake Wang</a> (Honavar). M.S. 2001. Dake was supported by a teachi
<li> <a href="http://www.cs.iastate.edu/~asokt/">Asok Tiyyagura.</a> M.S. 2000. Project: Mutual Informati
Asok has been supported through a research assistantship funded by the ISU Council on International Prog
<li>
```

Another example is a search for Dr. William Hsu's email address. The returned results are as follows:

Table 5.2 Email Entity Results with Keyword "William Hsu"

Rank	Entity	URL	Keywords
1	bhsu@cis.ksu.edu	http://ringil.cis.ksu.edu/Courses/Spring-2004/CIS830/Resources/RelatedLinks.html	William Hsu
2	william.hathaway@yale.edu	http://opa.yale.edu/news/article_print.aspx?id=5892	William
3	billyh@cs.ucla.edu	http://www.cs.ucla.edu/csd/people/cs_ta.html	William
4	wmeicher @ cs.cmu.edu	http://people.cs.cmu.edu/a_z/E.html	William
5	bill @ stat.cmu.edu	http://people.cs.cmu.edu/a_z/E.html	William
6	william.hathaway@yale.edu	http://opa.yale.edu/news/article.aspx?id=5892&f=13	William
7	wlovas @ cs.cmu.edu	http://people.cs.cmu.edu/a_z/L.html	William
8	br @ cs.cmu.edu	http://people.cs.cmu.edu/a_z/R.html	William
9	E-mail: bhsu [AT] cis.ksu.edu	http://people.cis.ksu.edu/~bhsu/	William Hsu
10	e-mail: banazir [AT] gmail.com	http://people.cis.ksu.edu/~bhsu/	William Hsu

Figure 5.3 Top 20 Email Entity Results with Keyword “William Hsu”



```
1:bhsu@cis.ksu.edu   is in the link http://ringil.cis.ksu.edu/Courses/Spring-2004/CIS830/Resources/RelatedLinks.html [ William Hsu  ]
2:william.hathaway@yale.edu   is in the link http://opa.yale.edu/news/article_print.aspx?id=5892 [ William  ]
3:billyh@cs.ucla.edu is in the link http://www.cs.ucla.edu/csd/people/cs_ta.html[ William]
4:wmeicher @ cs.cmu.edu is in the link http://people.cs.cmu.edu/a_z/E.html [ William  ]
5:bill @ stat.cmu.edu is in the link http://people.cs.cmu.edu/a_z/E.html [ William  ]
6:william.hathaway@yale.edu   is in the link http://opa.yale.edu/news/article.aspx?id=5892&f=13 [ William  ]
7:wlovas @ cs.cmu.edu   is in the link http://people.cs.cmu.edu/a_z/L.html [ William  ]
8:br @ cs.cmu.edu   is in the link http://people.cs.cmu.edu/a_z/R.html [ William  ]
9:E-mail: bhsu [AT] cis.ksu.edu   is in the link http://people.cis.ksu.edu/~bhsu/ [ William Hsu  ]
10:e-mail: banazir [AT] gmail.com   is in the link http://people.cis.ksu.edu/~bhsu/ [ William Hsu  ]
11:mcgahey@ee.pitt.edu   is in the link http://www.engr.pitt.edu/computer/research/cadence.html [ William  ]
12:newsroom@elsevier.com   is in the link http://www.cmu.edu/news/archive/2008/August/aug5_rewirebrain.shtml [ William  ]
13:annam@cs.ucla.edu   is in the link http://www.cs.ucla.edu/csd/people/cs_ta.html [ William  ]
14:kkim@ri.cmu.edu   is in the link http://www.ri.cmu.edu/education/alumni_by_year.html [ William  ]
15:robert.davis@colorado.edu   is in the link http://engineering.colorado.edu/facultystaff/directory.aspx [ William  ]
16:wcohen@cs.cmu.edu   is in the link http://www.lti.cs.cmu.edu/Faculty/ [ William  ]
17:annam@cs.ucla.edu   is in the link http://www.cs.ucla.edu/csd/people/cs_ta.html [ William  ]
18:kbe @ andrew.cmu.edu   is in the link http://people.cs.cmu.edu/a_z/E.html [ William  ]
19:awards@provost.wisc.edu   is in the link http://www.provost.wisc.edu/uaa/2008Awards.html [ William  ]
20:efros @ cs.cmu.edu   is in the link http://people.cs.cmu.edu/a_z/E.html [ William  ]
```

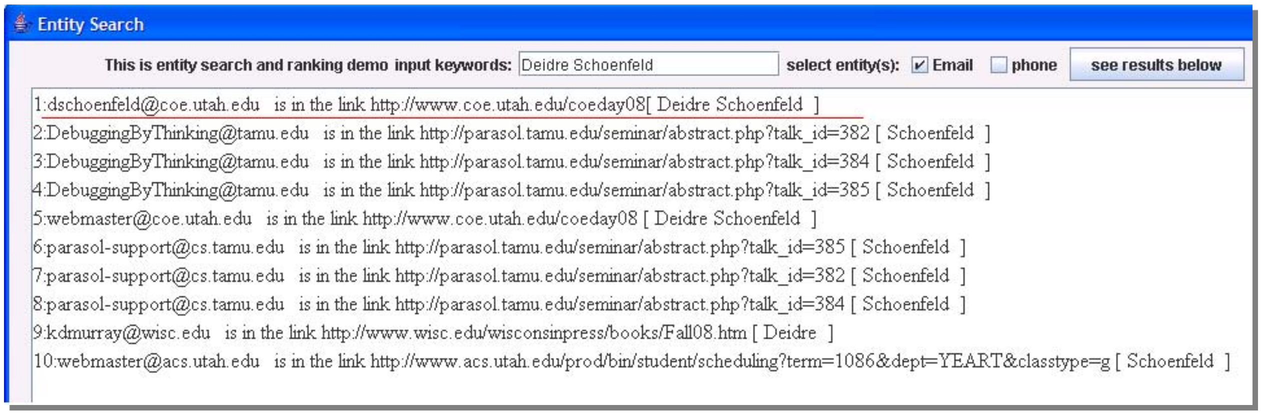
From Table 5.2 we can see that Dr. Hsu’s emails are returned in different ways that he wrote: “bhsu@cis.ksu.edu”, “bhsu [AT] cis.ksu.edu” and “banazir [AT] gmail.com”.

Consider another example; a student wants to find out Deidre Schoenfeld’s email address. Mrs. Deidre Schoenfeld is the organizer of “College of Engineering Day” at University of Utah. Therefore, with entity name “email” and keyword “Deidre Schoenfeld”, the results are shown as in table 5.3.

Table 5.3 Email Entity Results with Keyword “Deidre Schoenfeld”

Rank	Entity	URL	keywords
1	dschoenfeld@coe.utah.edu	http://www.coe.utah.edu/coeday08	Deidre Schoenfeld
2	DebuggingByThinking@tamu.edu	http://parasol.tamu.edu/seminar/abstract.php?talk_id=382	Schoenfeld
3	DebuggingByThinking@tamu.edu	http://parasol.tamu.edu/seminar/abstract.php?talk_id=384	Schoenfeld
4	DebuggingByThinking@tamu.edu	http://parasol.tamu.edu/seminar/abstract.php?talk_id=385	Schoenfeld
5	webmaster@coe.utah.edu	http://www.coe.utah.edu/coeday08	Deidre Schoenfeld

Figure 5.4 Top 10 Email Entity Results with Keyword “Deidre Schoenfeld”



This is entity search and ranking demo input keywords: Deidre Schoenfeld select entity(s): [✓ Email] phone see results below

```
1:dschoenfeld@coe.utah.edu   is in the link http://www.coe.utah.edu/coeday08[ Deidre Schoenfeld  ]
2:DebuggingByThinking@tamu.edu   is in the link http://parasol.tamu.edu/seminar/abstract.php?talk_id=382 [ Schoenfeld  ]
3:DebuggingByThinking@tamu.edu   is in the link http://parasol.tamu.edu/seminar/abstract.php?talk_id=384 [ Schoenfeld  ]
4:DebuggingByThinking@tamu.edu   is in the link http://parasol.tamu.edu/seminar/abstract.php?talk_id=385 [ Schoenfeld  ]
5:webmaster@coe.utah.edu   is in the link http://www.coe.utah.edu/coeday08 [ Deidre Schoenfeld  ]
6:parasol-support@cs.tamu.edu   is in the link http://parasol.tamu.edu/seminar/abstract.php?talk_id=385 [ Schoenfeld  ]
7:parasol-support@cs.tamu.edu   is in the link http://parasol.tamu.edu/seminar/abstract.php?talk_id=382 [ Schoenfeld  ]
8:parasol-support@cs.tamu.edu   is in the link http://parasol.tamu.edu/seminar/abstract.php?talk_id=384 [ Schoenfeld  ]
9:kdmurray@wisc.edu   is in the link http://www.wisc.edu/wisconsinpress/books/Fall08.htm [ Deidre  ]
10:webmaster@acs.utah.edu   is in the link http://www.acs.utah.edu/prod/bin/student/scheduling?term=1086&dept=YEART&classtype=g [ Schoenfeld  ]
```

The first ranked result is exactly what the student wants. This may save him some time by not looking up the official web site of College of Engineering Day” to search for contact information.

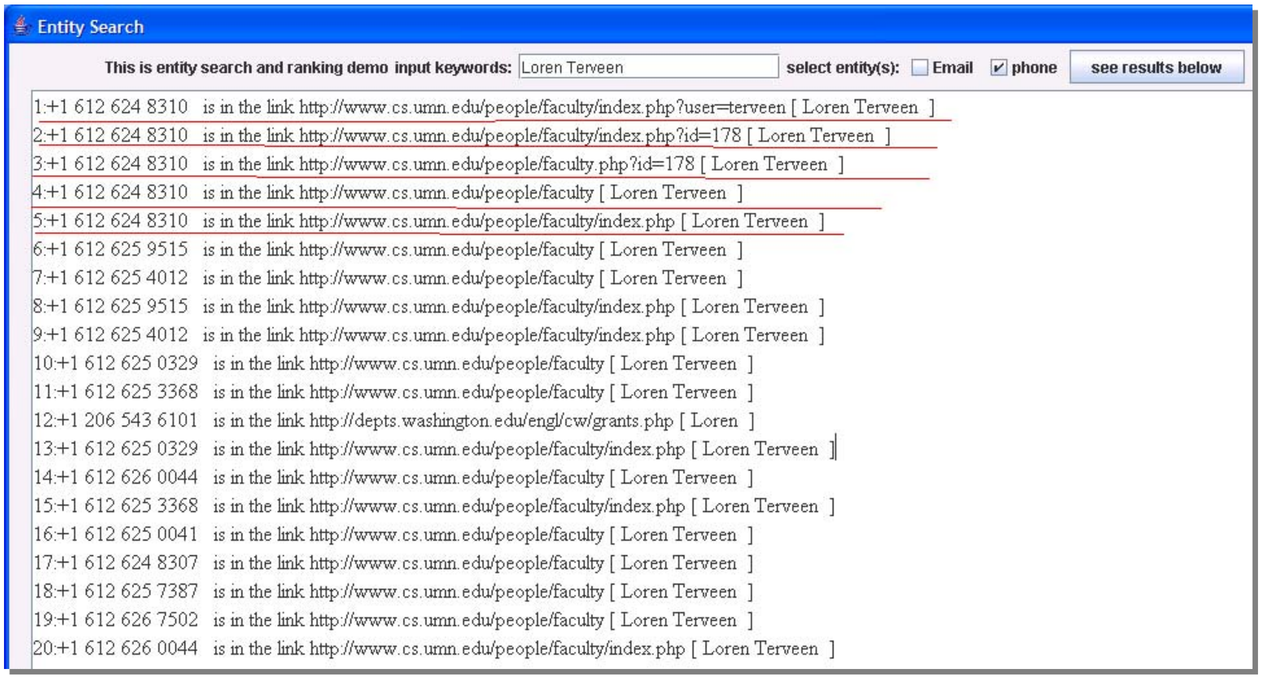
5.2 Phone entity results

Dr. Loren Terveen is an associate professor in Computer Science and Engineering Department in University of Minnesota. Suppose a student of his wants to find his phone number, by simply post the keyword “Loren Terveen”, Table 5.4 shows the results.

Table 5.4 Phone Entity Results with Keyword "Loren Terveen"

Rank	Entity	URL	Keywords
1	+1 612 624 8310	http://www.cs.umn.edu/people/faculty/index.php?user=terveen	Loren Terveen
2	+1 612 624 8310	http://www.cs.umn.edu/people/faculty/index.php?id=178	Loren Terveen
3	+1 612 624 8310	http://www.cs.umn.edu/people/faculty.php?id=178	Loren Terveen
4	+1 612 624 8310	link http://www.cs.umn.edu/people/faculty	Loren Terveen
5	+1 612 624 8310	http://www.cs.umn.edu/people/faculty/index.php	Loren Terveen
6	+1 612 625 9515	http://www.cs.umn.edu/people/faculty	Loren Terveen
7	+1 612 625 4012	http://www.cs.umn.edu/people/faculty	Loren Terveen
8	+1 612 625 9515	http://www.cs.umn.edu/people/faculty/index.php	Loren Terveen
9	+1 612 625 4012	http://www.cs.umn.edu/people/faculty/index.php	Loren Terveen
10	+1 612 625 0329	http://www.cs.umn.edu/people/faculty	Loren Terveen

Figure 5.5 Top 20 Phone Entity Results with Keyword "Loren Terveen"



This is entity search and ranking demo input keywords: Loren Terveen select entity(s): Email [✓ phone]

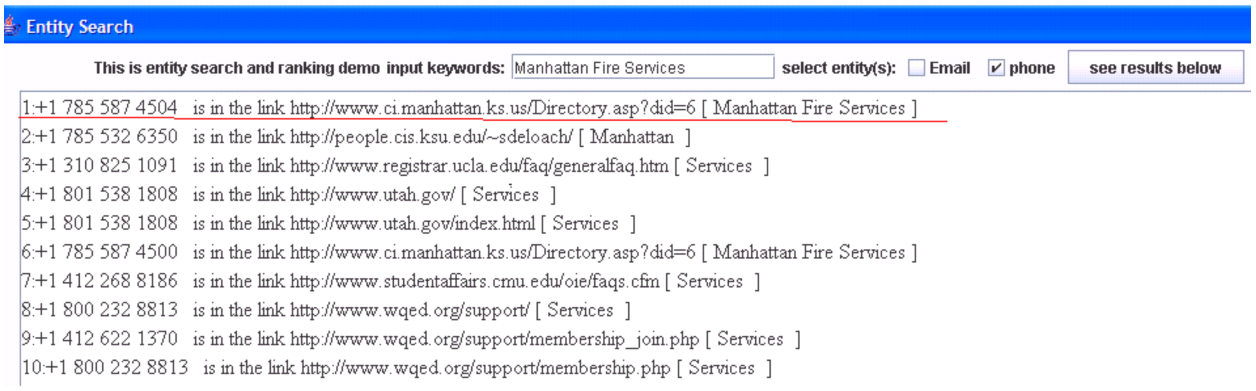
```
1:+1 612 624 8310   is in the link http://www.cs.umn.edu/people/faculty/index.php?user=terveen [ Loren Terveen  ]
2:+1 612 624 8310   is in the link http://www.cs.umn.edu/people/faculty/index.php?id=178 [ Loren Terveen  ]
3:+1 612 624 8310   is in the link http://www.cs.umn.edu/people/faculty.php?id=178 [ Loren Terveen  ]
4:+1 612 624 8310   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
5:+1 612 624 8310   is in the link http://www.cs.umn.edu/people/faculty/index.php [ Loren Terveen  ]
6:+1 612 625 9515   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
7:+1 612 625 4012   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
8:+1 612 625 9515   is in the link http://www.cs.umn.edu/people/faculty/index.php [ Loren Terveen  ]
9:+1 612 625 4012   is in the link http://www.cs.umn.edu/people/faculty/index.php [ Loren Terveen  ]
10:+1 612 625 0329   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
11:+1 612 625 3368   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
12:+1 206 543 6101   is in the link http://depts.washington.edu/engl/cw/grants.php [ Loren  ]
13:+1 612 625 0329   is in the link http://www.cs.umn.edu/people/faculty/index.php [ Loren Terveen  ]
14:+1 612 626 0044   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
15:+1 612 625 3368   is in the link http://www.cs.umn.edu/people/faculty/index.php [ Loren Terveen  ]
16:+1 612 625 0041   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
17:+1 612 624 8307   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
18:+1 612 625 7387   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
19:+1 612 626 7502   is in the link http://www.cs.umn.edu/people/faculty [ Loren Terveen  ]
20:+1 612 626 0044   is in the link http://www.cs.umn.edu/people/faculty/index.php [ Loren Terveen  ]
```

As in the example in Chapter 1, suppose a resident wants to find the phone number of "Manhattan Fire Services". Table 4.5 shows the results from my system.

Table 5.5 Phone Entity Results with Keyword "Manhattan Fire Services"

Rank	Entity	URL	Keywords
1	+1 785 587 4504	http://www.ci.manhattan.ks.us/Directory.asp?did=6	Manhattan Fire Services
2	+1 785 532 6350	http://people.cis.ksu.edu/~sdeloach/	Manhattan
3	+1 310 825 1091	http://www.registrar.ucla.edu/faq/generalfaq.htm	Services
4	+1 801 538 1808	http://www.utah.gov/	Services
5	+1 801 538 1808	http://www.utah.gov/index.html	Services

Figure 5.6 Top 10 Phone Entity Results with Keyword "Manhattan Fire Services"



Manhattan Fire Services

```
1:+1 785 587 4504   is in the link http://www.ci.manhattan.ks.us/Directory.asp?did=6 [ Manhattan Fire Services ]
2:+1 785 532 6350   is in the link http://people.cis.ksu.edu/~sdeloach/ [ Manhattan  ]
3:+1 310 825 1091   is in the link http://www.registrar.ucla.edu/faq/generalfaq.htm [ Services  ]
4:+1 801 538 1808   is in the link http://www.utah.gov/ [ Services  ]
5:+1 801 538 1808   is in the link http://www.utah.gov/index.html [ Services  ]
6:+1 785 587 4500   is in the link http://www.ci.manhattan.ks.us/Directory.asp?did=6 [ Manhattan Fire Services ]
7:+1 412 268 8186   is in the link http://www.studentaffairs.cmu.edu/oie/faqs.cfm [ Services  ]
8:+1 800 232 8813   is in the link http://www.wqed.org/support/ [ Services  ]
9:+1 412 622 1370   is in the link http://www.wqed.org/support/membership_join.php [ Services  ]
10:+1 800 232 8813   is in the link http://www.wqed.org/support/membership.php [ Services  ]
```

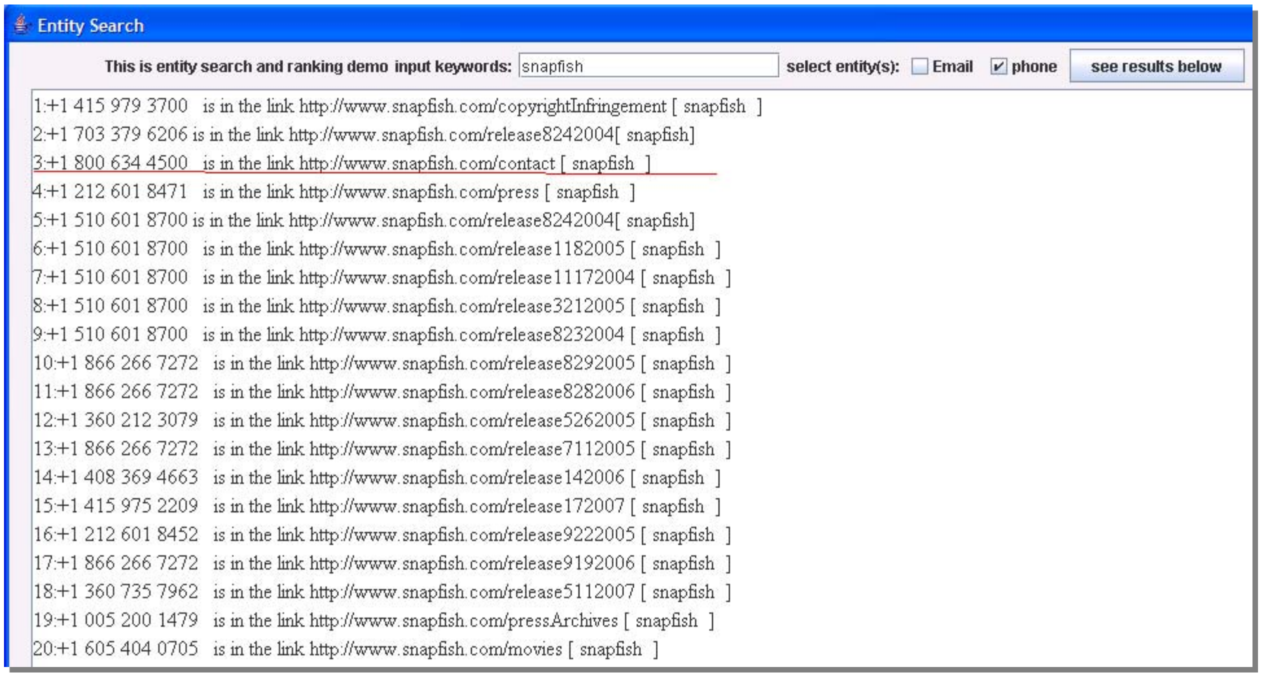
The result is more direct for the user because he can see what he wanted organized in a convenient way without spending lots of time to click through links**.**

The following example is to find the customer service phone number of the online digital photo printing company "Snapfish".

Table 5.6 Phone Entity Results with Keyword “snapfish”

Rank	Entity	URL	Keywords
1	+1 415 979 3700	http://www.snapfish.com/copyrightInfringement	Snapfish
2	+1 703 379 6206	http://www.snapfish.com/release8242004	Snapfish
3	+1 800 634 4500	http://www.snapfish.com/contact	Snapfish
4	+1 212 601 8471	http://www.snapfish.com/press	Snapfish
5	+1 510 601 8700	http://www.snapfish.com/release8242004	Snapfish

Figure 5.7 Phone Entity Results with Keyword “snapfish”



snapfish

[✓ phone]

```
1:+1 415 979 3700   is in the link http://www.snapfish.com/copyrightInfringement [ snapfish  ]
2:+1 703 379 6206 is in the link http://www.snapfish.com/release8242004[ snapfish]
3:+1 800 634 4500   is in the link http://www.snapfish.com/contact [ snapfish  ]
4:+1 212 601 8471   is in the link http://www.snapfish.com/press [ snapfish  ]
5:+1 510 601 8700 is in the link http://www.snapfish.com/release8242004[ snapfish]
6:+1 510 601 8700   is in the link http://www.snapfish.com/release1182005 [ snapfish  ]
7:+1 510 601 8700   is in the link http://www.snapfish.com/release11172004 [ snapfish  ]
8:+1 510 601 8700   is in the link http://www.snapfish.com/release3212005 [ snapfish  ]
9:+1 510 601 8700   is in the link http://www.snapfish.com/release8232004 [ snapfish  ]
10:+1 866 266 7272   is in the link http://www.snapfish.com/release8292005 [ snapfish  ]
11:+1 866 266 7272   is in the link http://www.snapfish.com/release8282006 [ snapfish  ]
12:+1 360 212 3079   is in the link http://www.snapfish.com/release5262005 [ snapfish  ]
13:+1 866 266 7272   is in the link http://www.snapfish.com/release7112005 [ snapfish  ]
14:+1 408 369 4663   is in the link http://www.snapfish.com/release142006 [ snapfish  ]
15:+1 415 975 2209   is in the link http://www.snapfish.com/release172007 [ snapfish  ]
16:+1 212 601 8452   is in the link http://www.snapfish.com/release9222005 [ snapfish  ]
17:+1 866 266 7272   is in the link http://www.snapfish.com/release9192006 [ snapfish  ]
18:+1 360 735 7962   is in the link http://www.snapfish.com/release5112007 [ snapfish  ]
19:+1 005 200 1479   is in the link http://www.snapfish.com/pressArchives [ snapfish  ]
20:+1 605 404 0705   is in the link http://www.snapfish.com/movies [ snapfish  ]
```

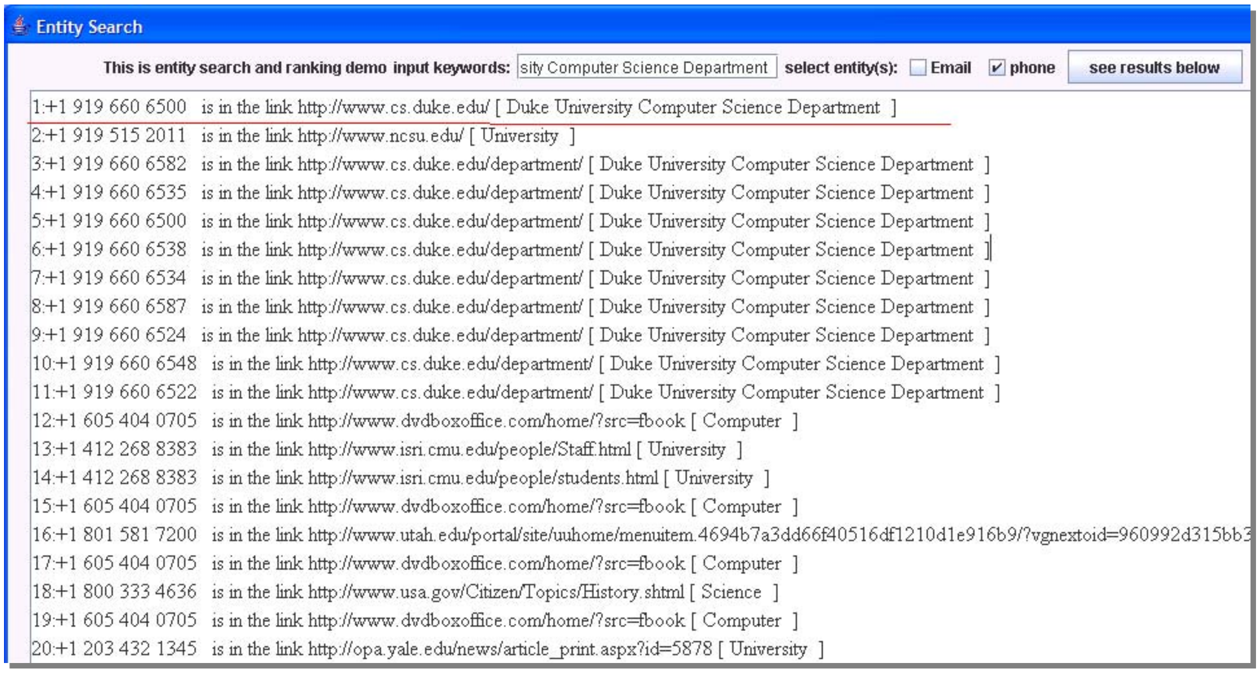
The phone number “+1 800 634 4500” is the customer contact number; it is ranked in the third place. Therefore, with associated URLs, the user can get what he wants in a very short time. Additionally, in the future work, some sounding contexts are shown for user’s convenience. Thus, user can have a clear idea if this is the exact result that he wants.

Another example is to search “Duke University Computer Science Department Phone” the results of the first five returned by my system are as follows:

Table 5.7 Phone Entity Results with Keyword "Duke University Computer Science Department"

Rank	Entity	URL	Keywords
1	+1 919 660 6500	http://www.cs.duke.edu/	Duke University Computer Science Department
2	+1 919 515 2011	http://www.ncsu.edu/	University
3	+1 919 660 6582	http://www.cs.duke.edu/department/	Duke University Computer Science Department
4	+1 919 660 6535	http://www.cs.duke.edu/department/	Duke University Computer Science Department
5	+1 919 660 6500	http://www.cs.duke.edu/department/	Duke University Computer Science Department

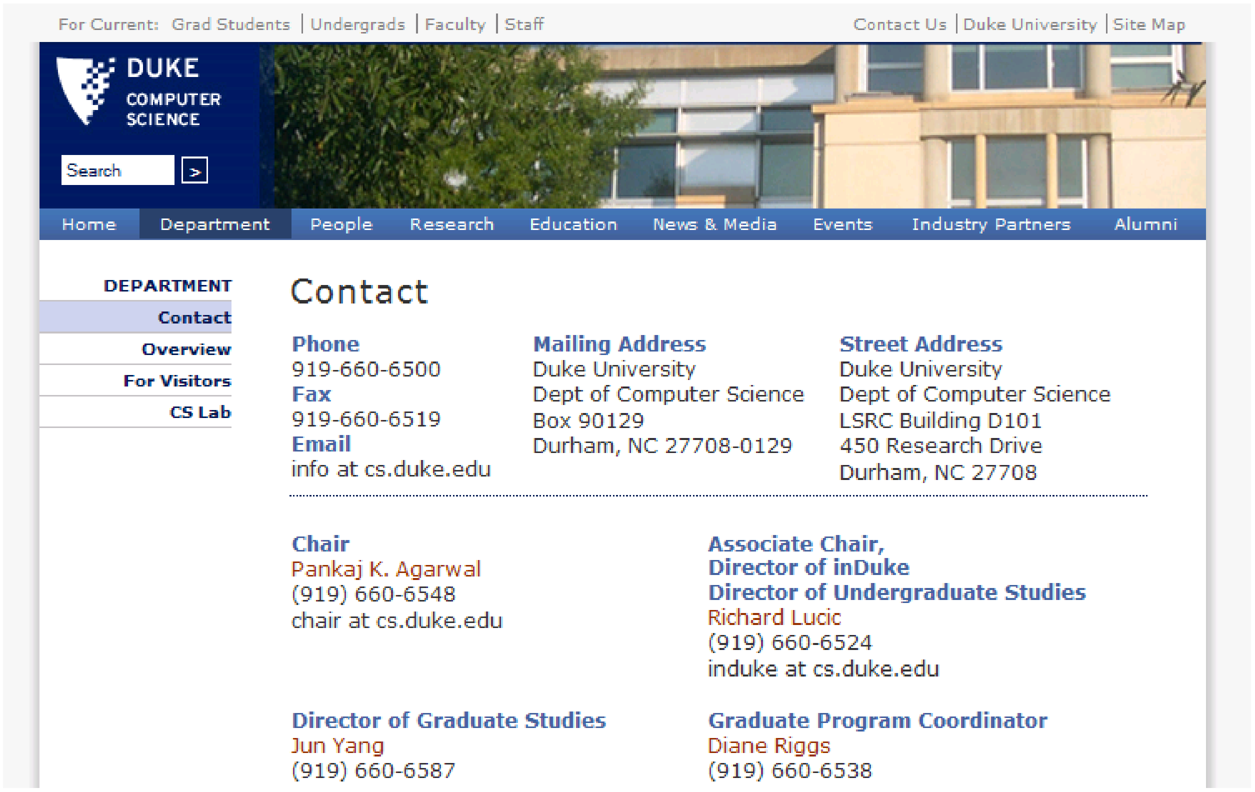
Figure 5.8 Phone Entity Results with Keyword "Duke University Computer Science Department"


Entity Search
This is entity search and ranking demo input keywords:
sity Computer Science Department
select entity(s):
Email
phone
see results below

```
1:+1 919 660 6500   is in the link http://www.cs.duke.edu/ [ Duke University Computer Science Department  ]
2:+1 919 515 2011   is in the link http://www.ncsu.edu/ [ University  ]
3:+1 919 660 6582   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
4:+1 919 660 6535   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
5:+1 919 660 6500   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
6:+1 919 660 6538   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
7:+1 919 660 6534   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
8:+1 919 660 6587   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
9:+1 919 660 6524   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
10:+1 919 660 6548   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
11:+1 919 660 6522   is in the link http://www.cs.duke.edu/department/ [ Duke University Computer Science Department  ]
12:+1 605 404 0705   is in the link http://www.dvdboxoffice.com/home/?src=fbook [ Computer  ]
13:+1 412 268 8383   is in the link http://www.isri.cmu.edu/people/Staff.html [ University  ]
14:+1 412 268 8383   is in the link http://www.isri.cmu.edu/people/students.html [ University  ]
15:+1 605 404 0705   is in the link http://www.dvdboxoffice.com/home/?src=fbook [ Computer  ]
16:+1 801 581 7200   is in the link http://www.utah.edu/portal/site/uuhome/menuitem.4694b7a3dd66f40516df1210d1e916b9/?vgnextoid=960992d315bb3
17:+1 605 404 0705   is in the link http://www.dvdboxoffice.com/home/?src=fbook [ Computer  ]
18:+1 800 333 4636   is in the link http://www.usa.gov/Citizen/Topics/History.shtml [ Science  ]
19:+1 605 404 0705   is in the link http://www.dvdboxoffice.com/home/?src=fbook [ Computer  ]
20:+1 203 432 1345   is in the link http://opa.yale.edu/news/article_print.aspx?id=5878 [ University  ]
```

From the results above, we can see that my system does not only return the first occurrence of an entity on a page; it returns all the entities associated with the keyword. However, my system ranks them in descending order of assessed quality according to my algorithm. Thus, the top-ranked hit is the putative best. Figure 5.9 shows the actual web page of http://www.cs.duke.edu/department/. There are several phone numbers in this page, but "+1 919 660 6500" is the exact one that user wants, and system returns it first.

Figure 5.9 Part of the web page of http://www.cs.duke.edu/department/



Contact

Phone
919-660-6500
Fax
919-660-6519
Email
info at cs.duke.edu

Mailing Address
Duke University
Dept of Computer Science
Box 90129
Durham, NC 27708-0129

Street Address
Duke University
Dept of Computer Science
LSRC Building D101
450 Research Drive
Durham, NC 27708

Chair
Pankaj K. Agarwal
(919) 660-6548
chair at cs.duke.edu

Associate Chair,
Director of inDuke
Director of Undergraduate Studies
Richard Lucic
(919) 660-6524
induke at cs.duke.edu

Director of Graduate Studies
Jun Yang
(919) 660-6587

Graduate Program Coordinator
Diane Riggs
(919) 660-6538

5.3 Results analysis

Table 5.8 below is the breakdown of Entity Rank into Entity-PageRank and the URL's *PageRank* [8]. Note that only the ordinal rank is shown, not the ranking measure itself. From the table, it is clear that the order could be quite different among EntityRank, an entity's page URL's *PageRank*, and an entity's Entity-PageRank. This justifies the combination of the two ranks into a single metric that better reflect an entity's relevance to a user query and its quality. For example, for the query on the phone number of Manhattan Fire Service, the *PageRank* of the URL that contains the correct result is only 125th among all the pages that contain any of the keywords. However, the Entity-PageRank of the phone number instance is ranked highest, and the total EntityRank is also ranked the highest. For the query of Vasant Honavar’e email, the highest EntityRank is the combination of the 14th highest Entity-PageRank and the highest PageRank value for the URL. However, some results are not as good, such as the entity “E-mail: bhsu [AT] cis.ksu.edu” and “e-mail: banazir [AT] gmail.com” and the result for the query “snapfish”. Their EntityRanks are low even though they shall be the most relevant result. For the first, the reason is that the results that ranked before them only contain “William” rather than “William Hsu”. If our ranking algorithm gives substantial higher weight to pages that contain all the keywords instead of just a subset of them, we can possible improve the ranking for these cases. Currently our metric does not consider this and thus it gave a less desirable result. The reason for the less than desirable result for “snapfish phone number” is because the user did not specify what phone numbers he wanted. As a result, the highest ranked phone number happens to be a local contact number, instead of the toll-free contact number. This result shows that there is also sometimes a semantic gap between what the user really wanted and what he entered as keywords.

Table 5.8 Result Analysis

Keywords	Entity Name	Extracted Entity	URL	Entity-Page Rank	URL's Page Rank	Entity Rank
Vasant Honavar	Email	honavar@cs.iastate.edu	http://www.cs.iastate.edu/~honavar/cas.html	15	1	1
William Hsu	Email	bhsu@cis.ksu.edu	http://ringil.cis.ksu.edu/Courses/Spring-2004/CIS830/Resources/RelatedLinks.html	1	8	1
William Hsu	Email	E-mail: bhsu [AT] cis.ksu.edu	http://people.cis.ksu.edu/~bhsu/	11	2	9
William Hsu	Email	e-mail: banazir [AT] gmail.com	http://people.cis.ksu.edu/~bhsu/	12	2	10
Deidre Schoenfeld	Email	dschoenfeld@coe.utah.edu	http://www.coe.utah.edu/coeday08	1	1	1
Loren Terveen	Phone	+1 612 624 8310	http://www.cs.umn.edu/people/faculty/index.php?user=terveen	1	101	1
Manhattan Fire Services	Phone	+1 785 587 4504	http://www.ci.manhattan.ks.us/Directory.asp?did=6	1	271	1
Snapfish	phone	+1 800 634 4500	http://www.snapfish.com/contact	2	5	3
Duke University Computer Science Department	Phone	+1 919 660 6500	http://www.cs.duke.edu/	1	77	1
Bruce Lin	Phone	+1 301 552 9701	http://www.umiacs.umd.edu/~cllin/	14	3	3

CHAPTER 6 - Conclusions and Future Work

6.1 Conclusion

In this work, I demonstrated that a simple entity search algorithm can provide meaningful rankings for retrieved entities on the web. From the experiments in the previous chapter, we can see that in most situations, the simple EntityRank algorithm ranks hits in an intuitive order, with the desired and relevant hits placed highly. This shows the power of a seeming simplistic algorithm for finding useful information on the web. From my limited experimentation, it appears that the algorithm works well most of the time. The research also reveals when such simple algorithms may not provide the best result, and points out possible next steps in this research area.

6.2 Future Work

One potential weakness revealed by the experiment is the “false connection” problem, where an entity is physically close to the search keywords, but is not logically connected to the keyword per the intention of the user. This is a hard problem to address in general, since it is always possible that the logical connection the user has in mind when issuing the query does not correspond to the physical connection between an entity and the keywords. It is unlikely that an algorithm that ranks an entity “independently” can yield better results in such cases. One possible approach is to identify features that depend on the relative proximity of entity names to keywords and are relevant to these logical connections. These features can then be used to further refine the ranking. The tasks of constructing, selecting, and learning similarity from these features are beyond the scope of this thesis. However, the work in this thesis indicates that this could be a potential direction for further work in entity search. In the future work, some queries will be applied to a larger dataset to compare results.

References

[1] How to write a phone number. Retrieved from URL: http://paulm.com/how_to/write_a_phone_number.html

[2] T. Cheng, X. Yan, and K. C.-C. Chang. EntityRank: Searching Entities Directly and Holistically. The 33rd international conference on Very large databases, 2007.

[3] T. Cheng, X. Yan, and K. C.-C. Chang. Supporting entity search: A large-scale prototype search system. SIGMOD, 2007.

[4] Definition of URL. Retrieved from URL: http://www.faqs.org/rfcs/rfc1738.html

[5] Definition of URI. Retrieved from URL: http://en.wikipedia.org/wiki/URI

[6] Heritrix Crawler. Retrieved from URL: http://crawler.archive.org/

[7] A-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity Ranking in Wikipedia. ACM Symposium on Applied Computing (SAC 2008). Fortaleza, Ceará, Brazil, March 16 - 20, 2008.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web, Technical Report Stanford Digital Library Technologies Project.

[9] ARC files. Retrieved from URL: http://crawler.archive.org/articles/developer_manual/arcs.html

[10] M. Bianchini, M. Gori, and F. Scarselli. Inside PageRank. ACM Trans. Inter. Tech., 5(1), 92–128, 2005.

[11] http://en.wikiquote.org/wiki/Albert_Einstein

[12] Beocat Cluster. Retrieved from URL: http://beocat.cis.ksu.edu/

[13] Open Database Connectivity (ODBC). Retrieved from URL: http://en.wikipedia.org/wiki/Open_Database_Connectivity

[14] Nutch. Retrieved from URL: http://lucene.apache.org/nutch/

[15] C.C.T. Kwok, O.Etzioni, and D.S. Weld. Scaling question answering to the web. *WWW*, pages 150-161, 2001.

Appendix A - Crawling seeds

http://www.cis.ksu.edu

http://www.cs.stanford.edu

http://dblife.cs.wisc.edu

http://www.cs.cmu.edu

http://www.cs.princeton.edu

http://www.cs.uiuc.edu

http://www.cs.cornell.edu

http://www.cs.columbia.edu

http://www.cs.washington.edu

http://www.cs.umd.edu

http://www.cs.umass.edu

http://www.cs.ucla.edu

http://www.cs.nyu.edu

http://www.cs.sunysb.edu

http://www.cs.umd.edu

http://www.cs.brown.edu

http://www.cs.usc.edu

http://www.cs.jhu.edu

http://www.cs.rpi.edu

http://www.cs.rice.edu

http://www.cs.yale.edu

http://www.cs.virginia.edu

http://www.cs.bu.edu

http://www.cs.colorado.edu

http://www.cs.rochester.edu

http://www.cs.duke.edu

http://www.cs.colostate.edu

http://www.cs.utah.edu

http://www.cs.arizona.edu

http://www.cs.pitt.edu

http://www.cs.caltech.edu

http://www.csd.cs.cmu.edu

http://www.csc.ncsu.edu

http://ww.cs.utk.edu

http://www.cs.wisc.edu

http://www.cs.dartmouth.edu

http://www.cs.byu.edu

http://www.dmoz.org/

http://www.snapfish.com

http://www.amazon.com

http://www.esteelauder.com/home.tmpl

http://www.dvdtalk.com/csr.html

http://www.ci.manhattan.ks.us/Directory.asp?did=6

http://www.ci.manhattan.ks.us/directory.asp

http://www.ci.manhattan.ks.us

http://kansasflinthills.travel/

http://www.yahoo.com

http://www.msn.com/

http://www.cnn.com

http://blogsofnote.blogspot.com/

http://www.kansas.gov

http://gov.ca.gov/