GROUPING SEARCH-ENGINE RETURNED CITATIONS FOR PERSON-NAME QUERIES

by

Reema Al-Kamha

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

June 2004

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Reema Al-Kamha

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	David W. Embley, Chair
Date	Stephen W. Liddle
Date	Parris Egbert

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the dissertation of Reema Al-Kamha in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscripts is satisfactory to the graduate committee and is ready for submission to the university library.

_____ _____

Date David W. Embley
Chair, Graduate Committee

Accepted for the Department

_____

David W. Embley
Graduate Coordinator

Accepted for the College

_____

G. Rex Bryce
Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

Grouping Search-Engine Returned Citations for
Person-Name Queries

Reema Al-Kamha

Department of Computer Science

Master of Science

In this thesis we present a technique to group search-engine returned citations for person-name queries, such that the search-engine returned citations in each group belong to the same person. To group the returned citations we use a multi-faceted approach that considers evidence from three facets: (1) attributes, (2) links, and (3) page similarity. For each facet we generate a confidence matrix. Then we construct a final confidence matrix for all facets. Using a threshold, we apply a grouping algorithm on the final confidence matrix. The output is a group of search-engine returned citations, such that the citations in each group relate to the same person.

ACKNOWLEDGMENTS

My foremost thank goes to my thesis adviser Dr. David W. Embley. Without him, this thesis would not have been possible. I thank him for his patience and encouragement that carried me on through difficult times, and for his insights and suggestions that helped to shape my research skills. His valuable feedback contributed greatly to this thesis.

I thank the rest of my thesis committee members: Dr. Stephen W. Liddle, Dr. Parris Egbert. Their valuable feedback helped me to improve the thesis in many ways.

I would like to thank my parents who always encourage me to continue my studies. I also want to thank my husband, Dr. Ghassan Wazan, my two children, Wael and Yaman for their support and patience.

I thank all the BYU data-extraction research group members for their help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Suppose a user is looking for information about the person William Barrett. Using Google, the query "William Barrett" returns about 11,300 citations[1]. Figure 1 shows the first 10 returned citations. The returned citations are for more than one person whose name is William Barrett. One citation refers to a professor in the Computer Science Department at Brigham Young University; two citations refer to a hero of the Texas Revolution; two citations refer to a Professor of Physics at the Royal College of Science in Dublin who lived between (1873-1910); one citation refers to a professor in the Department of Computer Engineering at San Jose State University; one citation refers to a person who lived between (1775-1860); two citations refer to a junior majoring in finance at East Tennessee State University; and one citation refers to a graphic designer. Normal search-engine ranking methods do not group citations by a specific person and therefore usually scatter citations referencing a single person throughout the returned results. It would be interesting to present the results in different ways. One way is to group the citations such that all those that refer to the same person would be together.

In this thesis we introduce a method that is able to group the returned citations from a search engine such as Google [Goo] or Yahoo [Yah] for a person-name query, such that each group of citations refers to the same person. Figure 2 represents the desired output for Figure 1. In the output we retain the basic search-engine returned

[1] We use citations to refer to the returned results that are related to a specific query in a search engine. Each citation usually contains the title of the web page found, text below the title that includes the keywords of the query, and the URL of the web page found.

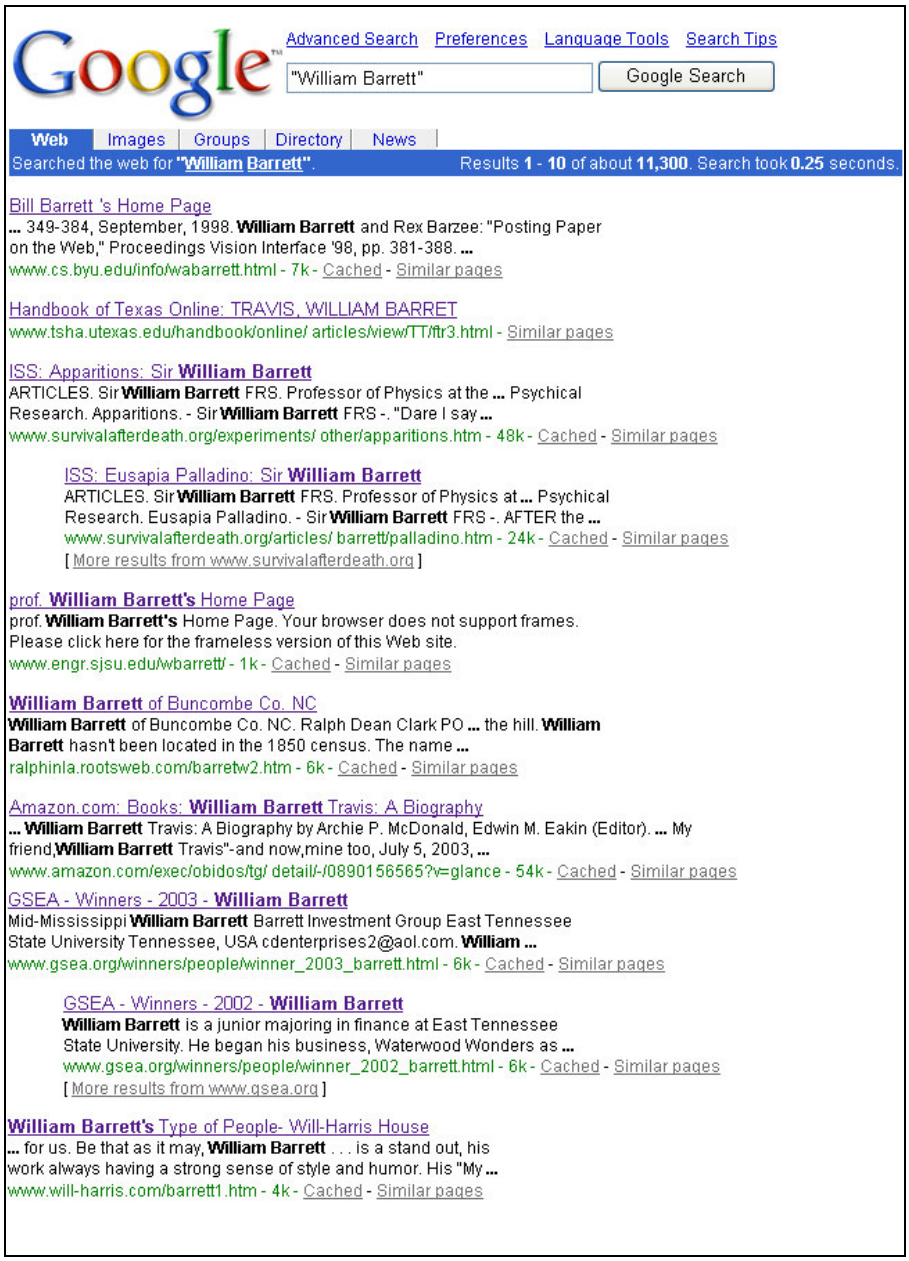

Figure 1: “William Barrett” Query—The First 10 Returned Citations.

Figure 2: “William Barrett” Grouping Result.

citations. Further, within each group we maintain the search engine ranking order, and among groups we maintain the relative order of citations as originally presented by the search engine.

In this thesis we answer the following questions. When a search-engine user types in a person name, what is the appropriate method to group search-engine returned citations? What are the facets that we should consider to help in the grouping? How can we combine the facets in such a way that we have a correct grouping?

Our method considers three facets: attributes, links, and page similarity. For each facet we generate a confidence matrix. Then we construct a final confidence matrix for all facets. Using a threshold, we apply a grouping algorithm on the final confidence matrix for all facets. The output is groups of the search-engine returned citations, such that the citations in each group relate to the same person.

We present our contribution of providing a solution to the interesting and useful problem of grouping person-name queries by person as follows. Chapter 2 presents related work. Chapter 3 introduces our multi-faceted approach to solving the problem by explaining the three facets we use (attributes, links, and page similarity), showing how to construct a confidence matrix for each facet, and how to combine all the confidence matrices into a final confidence matrix, and giving the algorithm we use to group returned citations. Chapter 4 discusses our experimental results. Chapter 5 draws conclusions and mentions potential future work.

Chapter 2

Related Work

We know of no literature directly related to the problem of grouping the citations returned by person-name queries for search engines. The problem however, is related to cross-document coreferencing [BB98], object identity [TKM01], and text classification [OTC01, OTC02].

A cross-document coreference occurs when the same person, place, event, or concept is discussed in more than one text source. Papers [BB98], [MY03], [WM02a], and [WM02b] all discuss approaches to coreferencing to distinguish between different entities that share the same, or a similar name. Papers [BB98], [WM02a], and [WM02b] use document vectors [Joa98] over terms that appear in the context in which the target name occurs. To adapt this idea for search engine returned citations for person-name queries, we would need to find a context, which is not straightforward for the mixture of structured, semistructured, and unstructured documents on the web. We nevertheless did some investigational experiments using this idea both with entire pages and with Google-returned text snippets in citations, but found that neither produced satisfactory results. Hence, we abandoned this idea in favor of the multi-faceted approach we developed instead. Similar to our idea of using attributes, [MY03] uses document vectors over biographical information such as birth year, birth place, spouse name, and occupation. If one document connects a name with a birth year, and another document connects the same name with the same birth year, typically, those two documents refer to the same person. [MY03] assumes, however, that documents are rich with biographic facts, which is not the case in our context because we are dealing

with different kinds of web pages that may, but usually do not, contain biographical information.

Object identification refers to the task of deciding that two observed objects are in fact one and the same object. This concept applies in our research because we are trying to decide if two or more citations are related to the same person. Paper [TKM01] surveys the various approaches to solving the object identity problem. All techniques that are mentioned in [TKM01] compare an object's shared attributes in order to identify matching objects, while our technique involves links and page similarity in addition to attributes.

With regard to attributes,[TKM01] mentions two models that are typically used to resolve object identity. One technique is vector space modeling [Joa98], and the other technique is probabilistic modeling [HR97, HR98]. In our research, it is not appropriate to apply vector space modeling over attributes because web pages do not usually contain all attributes; indeed, they often contain no attributes. Thus, vectors would likely have many missing components, which would make the cosine measure very low (possibly non existent) even when the pages are for the same person. Probabilistic modeling described in [HR97] and [HR98] also compares objects based on shared attributes and uses appearance probability to determine the similarity between objects. Appearance probability requires a comparison between observed attributes of the objects. In our case for pages on the web, citations that relate to the same person may not have any matching attributes.

The goal of text classification [Jao98] is to classify documents into a fixed number of predefined categories. Each document can be in zero, one, or several

categories. In our research we apply classification ideas, but we classify returned citations without knowing in advance how many different persons a person-name query will yield. We cannot apply standard classification techniques directly to our work because standard classification methods require predefined categories and training data to be able to distinguish between predefined categories. Since we do not know our categories in advance, we can neither predefine the categories nor specify training data for them.

Chapter 3

A Multi-Faceted Approach

When a user enters a first name and last name or a full name of a person as a query to a search engine, our objective is to put the returned citations in groups such that each group relates to one person. Our approach is multi-faceted. Each facet represents a relevant aspect of the problem space about which we can gather evidence that two citations reference the same person or different persons. In this thesis we consider attributes about a person, links within and among sites, and page similarity as facets. We consider each facet separately.

3.1 Attributes

We can obtain evidence about whether two citations refer to the same person by considering values for attributes. If identifying information about a person p appears in two different web pages w_1 and w_2 referenced respectively by citations c_1 and c_2, and if the identifying information is the same, then we can be highly confident about grouping c_1 and c_2 together for p.

To apply this idea, there are a number issues to consider. What identifying information are we likely to find? Can we recognize the identifying information? How do we know whether recognized identifying information refers to the same person for whom we are querying? To answer these questions, we looked for attributes that appear often in web pages of citations returned as results of person-name queries. Identifying

attributes we found by manual inspection that satisfy these criteria are phone number, email address, state, city, and zip code.

To extract values from a web page, we write regular expressions for each attribute. In addition for state, city, and zip code since we are looking for identifying information about a person (not information about references to a state or city and not isolated five-digit integers), we only extract state, city, and zip code values in an address context[2]. For example, to extract a city we extract all strings that match the regular expression *([A-Z](\w)+( )?){1,3}* and satisfy the context specification consisting of this string followed by an optional comma, white space, and a state name. We obtain states from a list that contains all state names and their abbreviations.

For a web page referenced by a person-name query, we extract all the attribute values that match the regular expressions and satisfy the context specifications for the attributes. Then when two web pages referenced by two citations for the same person-name query have the same value for a specific attribute or for several specific attributes, we can be reasonably confident that the identical person names in the two web pages refer to the same person. Note that we make no attempt to determine whether an extracted attribute's value is the attribute value of the person whose name is on the web page. Thus, for example, "Provo, UT 84604" might be the person's city, or the address of the web site provider hosting this particular web page—we do not know, and we assume it does not matter.

[2] In this initial investigative study our focus is on people in the U.S. With additional effort we can extract world address information.

3.2 Links

We can obtain evidence about whether two citations refer to the same person by considering links (URLs) among citations. People usually post information on only a few host servers, and often on only one. Thus, if two URLs of two returned citations for a person-name query share a common host, we can be reasonably confident that they refer to the same person. Figure 3 for example shows two citations for "David Embley" that share the host name *www.cs.byu.edu*.

Besides hosting information on the same server, people often link one page about a person to another page about that same person. Thus, if the URL of one citation has the same host as one of the URLs that belongs to the web page referenced by the other citation, we can be reasonably confident that they refer to the same person. Figure 4, for example, shows two citations and the web page for the second citation. The URL of the first citation has the same host *www.cs.byu.edu* as the URL *http://www.cs.byu.edu-/info/dwembley.html* that belongs to the web page referenced by the second citation.

To apply these ideas, there is an issue of interest to consider. It is common to have two different persons that have the same name in two citations that have a popular host like *www.yahoo.com*. Because many names often appear on popular hosts, when two citations share a popular host, we have less confidence that they refer to the same person. Thus, we need to find a way to determine if the host is popular so that we can observe this exception to the general rule. One solution might be to have a list of all popular hosts, but it is difficult to know and keep track of all of them. Furthermore, host

popularity is dynamic and changes over time. Another solution, which we decided to adopt, is to find the number of pages that point to a host. The query *link:siteURL* in Google shows all pages and gives a count of the number of pages that point to that URL. For example, *link:www.google.com* shows and counts all the pages that point to Google's home page. (Without having a simple way to obtain this count, it would be unreasonable to rely on this number.) We determined empirically that a host *h* is popular for person-name queries if more than 400 pages point to *h*.

Figure 3: Two Citations that have the Same Host, www.cs.byu.edu.

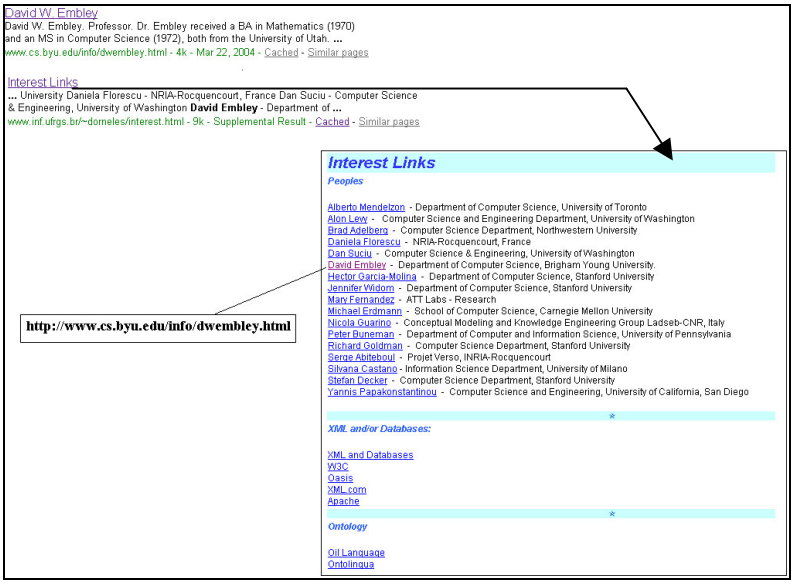

Figure 4: Two Citations with the Page of One Referring to the Host of the Other.

If two citations c_1 and c_2 that are results of a person-name query share the same non-popular host, or if the URL of one citation c_1 has the same non-popular host as one of the URLs that belongs to the web page referenced by the other citation c_2, then we can be confident about grouping c_1 and c_2 together for the same person.

3.3 Page Similarity

We can obtain evidence about whether two citations refer to the same person by considering the similarity between web pages referenced by the two returned citations. If two different web pages contain the same person name and the pages are similar, then we can be reasonably confident that they refer to the same person.

To apply this idea, there are a number of issues to consider. What are the useful shared words that we can consider? How can we use shared words to determine page similarity? How can we obtain a stop-word list to eliminate common words that appear in many web pages?

To answer these questions, we looked at many web pages referenced by person-name queries to see what kinds of words they share. We noticed that if two web pages refer to the same person, there are specific words associated with that person. For example, for David Embley, who is a professor and a co-director of the Data Extraction research group in the computer Science Department at Brigham Young University, two adjacent words such as *Data Extraction, Computer Science*, and *Brigham Young* appear in many web pages that have his name. As another example, many web pages that refer to Sandra Rogers contain *Lessons from the Light*, a book she wrote. Using these examples as a guide, we have chosen to consider pairs of words that start with a capital letter and that are either adjacent or separated by a connector (*and*, *or*, *but*) or by

a preposition which may be followed by an article (*a*, *an*, *the*) or by a single capital letter followed by dot. The form considered is thus:

Cap-Word (*Connector* | *Preposition (Article)*? | (Capital-Letter Dot))? *Cap-Word.*

Cap-Word is a word of two or more letters that starts with a capital letter. We call this pattern "adjacent cap-word pairs[3]."

We must, however, ignore adjacent cap-word pairs such as *Home Page* and *Privacy Policy* that often occur on web pages. We eliminate these pairs by constructing a stop-word list, which is a list of frequently appearing adjacent cap-word pairs. To construct our list, we collected approximately 10,000 web documents taken at random from the Open Directory Project, DMOZ [Dmo]. The Open Directory contains about 3.5 million web documents that are divided into categories; each category also contains subcategories. We obtained the DMOZ XML document that contains all listed categories and subcategories, and the URLs of the web pages that are in the subcategories. This resulted in a list of URLs that covers all the subcategories. From this list we obtained 10,000 documents from the 3,500,000 by selecting every 350th URL. After we collected 10,000 web documents, we constructed all adjacent cap-word pairs. We sorted the pairs according to their frequency and considered all pairs with a frequency greater than two to be stop words.

We consider the number of adjacent cap-word pairs as an indicator of the similarity between two web pages. In particular, we consider whether two web pages share exactly one, exactly two, exactly three, or four or more adjacent cap-word pairs. The greater the number of adjacent cap-word pairs, the greater the similarity between

[3] As a programming artifact since Java regular expressions do not recognize overlapping strings, we do not consider overlapping cap-word pairs.

the pages. Empirically, however, we found that four seems to be enough as long as we first eliminate adjacent cap-word pairs that appear in our stop list.

3.4 Confidence Matrix Construction

We construct a confidence matrix, one for each facet: attributes, links, and page similarity. The confidence matrix for each facet is an upper trianguler matrix over all pairs of the n returned citations C_1, C_2, …. , C_n. The value of each element C_{ij} $(i < j)$ in the confidence matrix represents the confidence that two returned citations C_i and C_j refer to the same person. The confidence value is 0 for a facet f if there is no evidence for f to indicate that citations C_i and C_j may refer to the same person. When there is evidence that C_i and C_j may refer to the same person for a facet f, C_{ij} is the conditional probability that C_i and C_j refer to the same person given the evidence for f.

In order to compute the conditional probabilities that represent confidence values, we construct a training set. We used the following criteria for the set of person names in the training set. First, the names set should contain male names, female names, and gender-neutral names. Second, the names set should contain names such that the returned citations are grouped in different size groups—small, medium, and large. Third, the names set should contain names such that the returned citations are grouped into different numbers of groups—few groups and many groups. Using these criteria, we selected 9 person names: Lynn Larson, Chris Webb, Dan Smith, David Embley, William Walker, Judy Green, Linda Bishop, Tracy Jones, and Sandra Rogers. This name set contains male names (Dan, David, William), female names (Judy, Linda, Sandra), and gender-neutral names (Lynn, Chris, Tracy). Every name in the name set

returns groups of small (1-2) and medium (4-10) sizes; only the name David Embley contains a large group with more than 40 citations. The number of groups varies from a small number of groups such as two groups in the case of David Embley, to a medium number of groups such as 28 groups in case of Sandra Rogers, to a large number of groups such as 30 to 37 groups for the rest of the names.

To construct our training data, we entered each name as a query for Google, and we collected the first 50 returned citations for each name. For 50 returned citations there are 49+48+…+2+1 = 1,225 comparison pairs. Since we have 9 names, the total number of comparisons is 9*1225 = 11,025. Figure 5 shows the first 10 of the 11,025 lines of our training data. For each pair of citations in the 50 returned citations for each name, we recorded the following information:

- *Same Person*: whether the names are for the same person;
- *Phone*: whether the web pages to which they link contain the same phone number;
- *Email*: whether the web pages to which they link contain the same email address;
- *Zip*: whether the web pages to which they link contain the same address zip code;
- *City*: whether the web pages to which they link contain the same address city;
- *State*: whether the web pages to which they link contain the same address state;
- *Host1*: whether the citations have URLs in the same host;
- *Host2*: whether the URL of one citation has the same host as one of the URLs that belong to the web page of the other citation;
- *Share1*: whether the web pages referenced by the citations share exactly one adjacent cap-word pair;

- *Share2*: whether the web pages referenced by the citations share exactly two adjacent cap-word pair;
- *Share3*: whether the web pages referenced by the citations share exactly three adjacent cap-word pair; and
- *Share≥4*: whether the web pages referenced by the citations share four or more adjacent cap-word pairs.

The values are "Yes", "No", "N/A" (not available), and "P" which means the host name is popular (is referenced by more than 400 other sites).

	Same Person	**Phone**	**Email**	**Zip**	**City**	**State**	**Host1**	**Host2**	**Share1**	**Share2**	**Share3**	**Share≥4**
C_1,C_2	Yes	N/A	N/A	N/A	N/A	N/A	Yes	No	No	No	Yes	No
C_1,C_3	Yes	N/A	Yes	N/A	N/A	N/A	No	No	No	Yes	No	No
C_1,C_4	No	N/A	N/A	N/A	N/A	N/A	P	No	Yes	No	No	No
C_1,C_5	Yes	N/A	Yes	N/A	No	No	No	No	No	Yes	No	No
C_1,C_6	Yes	N/A	Yes	N/A	Yes	Yes	No	Yes	No	No	No	Yes
C_1,C_7	Yes	N/A	N/A	N/A	N/A	N/A	No	No	No	Yes	No	No
C_1,C_8	Yes	N/A	N/A	N/A	No	Yes	No	No	Yes	No	No	No
C_1,C_9	Yes	N/A	N/A	N/A	No	Yes	No	No	Yes	No	No	No
C_1,C_{10}	Yes	N/A	N/A	N/A	No	No	No	P	No	Yes	No	No
C_1,C_{11}	No	N/A	N/A	N/A	No	No	No	No	No	No	No	No

Figure 5: A Sample of the Training Set.

We use our training set to estimate the conditional probabilities as follows. For our attribute facet, we use the training set to estimate the probability that two citations refer to the same person knowing that the web pages referenced by the citations have either the same phone, email, address zip code, address city, address state, or any combination of these attributes. For example, we estimate *P*(*Same Person* = "Y*es*" | *Email* = "*Yes*"), which is the probability that two citations refer to the same person

knowing that the web pages referenced by them have the same email address, by dividing the number of citation pairs that are related to the same person and have the same email by the number of citation pairs that have the same email address in the training set. For pairs, triples, quadruples, and quintuples of attributes, we also compute conditional probabilities. For example, we estimate *P*(*Same Person* = "*Yes*" | *City* = "*Yes*" and *State* = "*Yes*") which is the probability that two citations refer to the same person knowing that the web pages referenced by them share the same address city and state, by dividing the number of citation pairs that are related to the same person and have the same address city and state by the number of citation pairs that share same address city and state in the training set.

For our link facet, we use our training set to estimate the probability that two citations refer to the same person knowing that the URLs of the citations share the same non-popular host, or the URL of one citation has the same non-popular host as one of the URLs on the web page referenced by the other citation, or the URLs of the citations share the same non-popular host and the URL of one citation has the same non-popular host as one of the URLs on the web page referenced by the other citation. For example, we estimate *P*(*Same Person = "Yes"* | *Host1 = "Yes"* and *Host is non-popular*) by dividing the number of citation pairs that are related to the same person and have the same non-popular host by the number of citation pairs that share a common, non-popular host.

For our page similarity facet, we use the training set to estimate the probability that two citations refer to the same person knowing that the web pages referenced by them share exactly one, two, three, four or more pairs of two adjacent cap-word pairs.

For example, we estimate $P(Same\ Person = "Yes" \mid Share2 = "Yes")$, which is the probability that two citations refer to the same person knowing that the web pages referenced by them share exactly two adjacent cap-word pairs in our training set, by dividing the number of citation pairs that are related to the same person and share two cap-word pairs by the number of citation pairs that share two cap-word pairs.

3.5 Final Confidence Matrix

We generate the final confidence matrix by combining the confidence matrices for the three facets using Stanford certainty theory [LS97]. Stanford certainty theory defines a confidence measure and generates some simple rules for combining independent evidence[4]. If evidence from two independent observations supports the same result, Stanford certainty theory gives the following rule to combine the evidence from these two independent observations. Suppose $CF(E_1)$ is the certainty factor associated with evidence E_1 for some observation B and $CF(E_2)$ is the certainty factor associated with evidence E_2 for the same observation B, then the new certainty factor CF of B, called the compound certainty factor of B, is calculated by $CF(E_1)+CF(E_2)-(CF(E_1)*CF(E_2))$. By using this rule repeatedly, it is possible to combine the results of evidence from any number of independent events that are used for determining B. Thus, each element in the final matrix is the Stanford measure for all the corresponding values in the matrixes of all facets and represents the confidence value that the two citations refer to the same person.

[4] In our approach we assume that the three facets are independent as is typical in Bayesian reasoning even though this might not be entirely true.

3.6 Grouping Algorithm

Our grouping algorithm takes as an input the final confidence matrix, and it returns as output groups of the search-engine returned citations, such that the citations of each group refer to the same person. The idea of the grouping algorithm is that if we are highly confident about grouping two citations C_i and C_j together in a set S_1, and we are highly confident about grouping two citations C_j and C_k together in a set S_2, and S_1 and S_2 share one or more citations (C_j in our example), then we are confident about grouping S_1 and S_2 together in one group S_3. We keep merging any two sets of citations that share one or more citations until no citation is shared between any two sets. The threshold we use for "highly confident" is 0.8, which we determined empirically.

3.7 Example

As an example, we apply our ideas to the first 10 returned citations for the person-name query "Kelly Flanagan" Figure 6 shows the results of the Google query. We label the first 10 returned C_1 through C_{10}.

Figure 7 shows the confidence matrix for the attributes facet. Pages referenced by the two citations C_1 and C_2 have the same zip, city, and state, which are "Provo", "UT", and "84604". From our training data we have *P(Same Person = "Yes" | City = "Yes"* and *State= "Yes"* and *Zip= "Yes")* = 0.99, so the confidence value that C_1 and C_2 are related to the same person is 0.99. Also, pages referenced by the two citations C_1 and C_8 and the two citations C_2 and C_8 have the same city and state, which are "Provo" and "UT". Pages referenced by the two citations C_4 and C_7 have the same city and state, which are

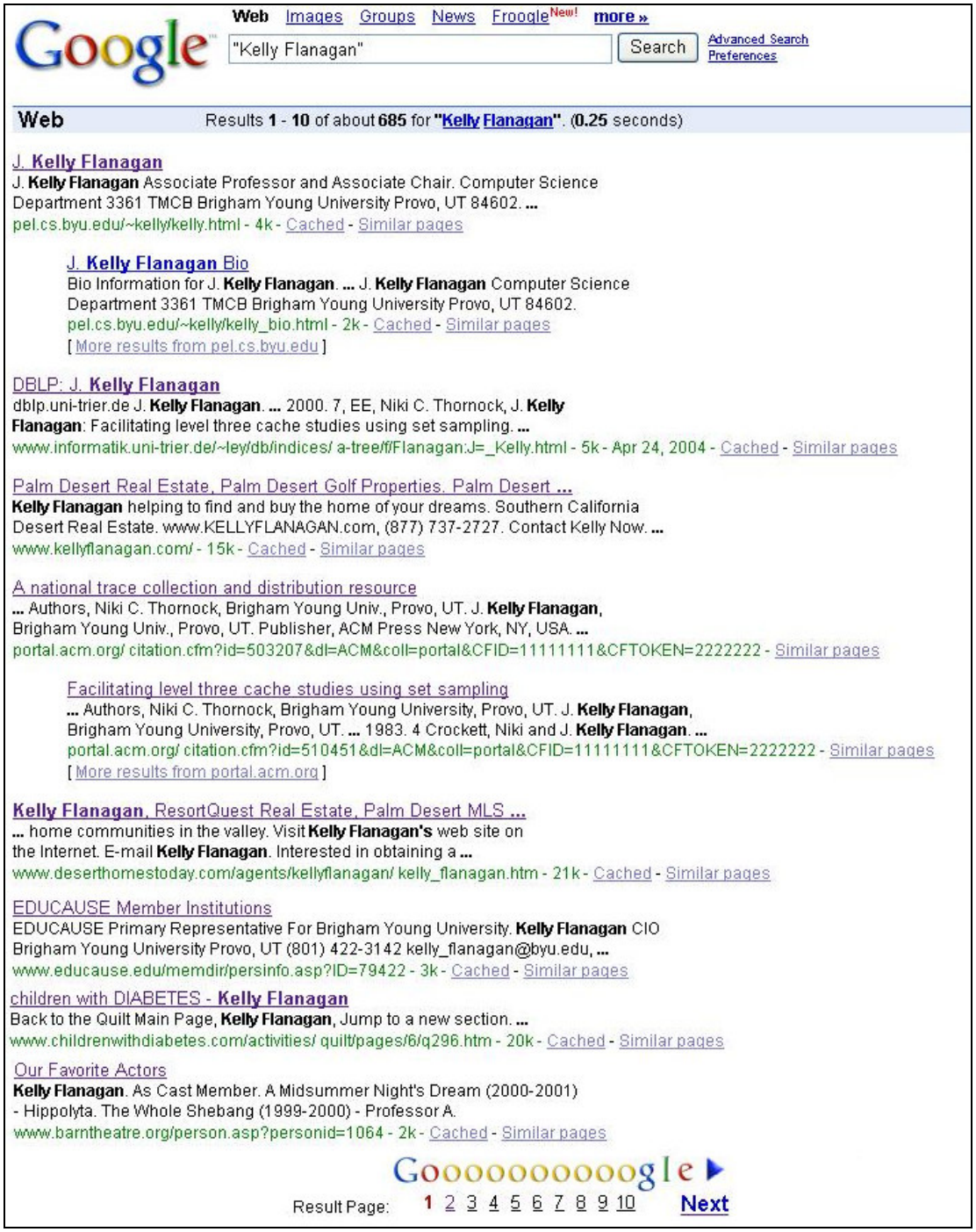

Figure 6: "Kelly Flanagan" Query —The First 10 Returned Citations.

"Palm Desert" and "California". From our training data we have *P(Same Person ="Yes" | City = "Yes"* and *State = "Yes")* = 0.96, so the confidence value that C_1 and C_8 are related to the same person is 0.96, the confidence value that C_2 and C_8 are related to the same person is 0.96, and the confidence value that C_4 and C_7 are related to the same person is 0.96.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_1	1	0.99	0	0	0	0	0	0.96	0	0
C_2		1	0	0	0	0	0	0.96	0	0
C_3			1	0	0	0	0	0	0	0
C_4				1	0	0	0.96	0	0	0
C_5					1	0	0	0	0	0
C_6						1	0	0	0	0
C_7							1	0	0	0
C_8								1	0	0
C_9									1	0
C_{10}										1

Figure 7: Confidence Matrix for Attributes Facet.

Figure 8 shows the confidence matrix for the links facet. Citations C_1 and C_2 have the same host name, and also C_1 refers to the host of C_2. From our training data we have *P*(*Same Person = "Yes" | Host1 = "Yes"* and *Host1 is non-popular* and *Host2 = "Yes"* and *Host2 is non-popular)* = 0.99, so the confidence value that citations C_1 and C_2 are related to the same person is 0.99. Citations C_5 and C_6 have the same host name, and from the training data *P*(*Same Person = "Yes" | Host1 = "Yes"* and *Host is non-popular)* = 0.99. Thus the confidence value that C_5 and C_6 are related to a same person is 0.99. In addition, C_3 refers to the host of C_5 and C_3 refers to the host of C_6 .From the training data we have that *P(Same Person = "Yes" | Host2 = "Yes"* and *Host is non-popular)* = 0.99. Thus the confidence value that C_3 and C_5 are related to the same

person is 0.99, and the confidence value that C_3 and C_6 are related to a same person is 0.99.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_1	1	0.99	0	0	0	0	0	0	0	0
C_2		1	0	0	0	0	0	0	0	0
C_3			1	0	0.99	0.99	0	0	0	0
C_4				1	0	0	0	0	0	0
C_5					1	0.99	0	0	0	0
C_6						1	0	0	0	0
C_7							1	0	0	0
C_8								1	0	0
C_9									1	0
C_{10}										1

Figure 8: Confidence Matrix for Links Facet.

Figure 9 shows the confidence matrix of the page similarity facet. The citations C_1 and C_2 share more than four adjacent cap-word pairs which are *Associate Professor, Brigham Young, Performance Evaluation, Trace Collection, Computer Organization,* and *Computer Architecture*. Also, citations C_2 and C_3 share more than four adjacent cap-word pairs which are *Memory Hierarchy, Brent E. Nelson, System-Assisted Disk, Simulation Technique, Stochastic Disk, Winter Simulation, Chordal Spoke, Interconnection Network, Transaction Processing, Benchmarks Using, Performance Studies, Incomplete Trace,* and *Heng Zho*. From the training data $P(\text{Same Person} = \text{“Yes”} \mid \text{Share} \geq 4 = \text{“Yes”}) = 0.95$. Thus, the confidence value that C_1 and C_2 are related to a same person is 0.95, and the confidence value that C_2 and C_3 are related to a same person is 0.95. Citations C_1 and C_8 share one adjacent cap-word pair, which is *Brigham Young*. Also, citations C_2 and C_8 share one adjacent cap-word pair, which is *Brigham Young*. From the training data $P(\text{Same Person} = \text{“Yes”} \mid \text{Share1} = \text{“Yes”}) = 0.78$. Thus, the confidence value that C_1 and C_8 are related to a same person is 0.78, and the

confidence value that C_2 and C_8 are related to a same person is 0.78. In addition, citations C_4 and C_7 share three adjacent cap-word pairs, which are *Palm Desert, Real Estate,* and *Desert Real*. From the training data *P(Same Person = "Yes" | Share3 = "Yes")* = 0.92. Thus, the confidence value that C_4 and C_7 are related to a same person is 0.92.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_1	1	0.95	0	0	0	0	0	0.78	0	0
C_2		1	0.95	0	0	0	0	0.78	0	0
C_3			1	0	0	0	0	0	0	0
C_4				1	0	0	0.92	0	0	0
C_5					1	0	0	0	0	0
C_6						1	0	0	0	0
C_7							1	0	0	0
C_8								1	0	0
C_9									1	0
C_{10}										1

Figure 9: Confidence Matrix for Page Similarity Facet.

Figure 10 shows the final confidence matrix. For example, we obtain the final confidence value between citations C_1 and C_8 using Stanford certainty theory as 0.96 + 0 + 0.78 - 0.96*0 - 0.96*0.78 - 0.78*0 + 0.96*0*0.78 = 0.9912.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
C_1	1	0.99	0	0	0	0	0	0.99	0	0
C_2		1	0.95	0	0	0	0	0.99	0	0
C_3			1	0	0.99	0.99	0	0	0	0
C_4				1	0	0	0.99	0	0	0
C_5					1	0	0	0	0	0
C_6						1	0	0	0	0
C_7							1	0	0	0
C_8								1	0	0
C_9									1	0
C_{10}										1

Figure 10: Final Confidence Matrix.

Finally, we apply the grouping algorithm on the final confidence matrix. First we obtain all citations pairs whose confidence value is more than 0.8, as follows:

$\{C_1, C_2\}, \{C_2, C_3\}, \{C_3, C_5\}, \{C_3, C_6\}, \{C_4, C_7\}, \{C_1, C_8\}, \{C_2, C_8\}$

We then merge groups that share at least one citation, and we continue merging until there is no merge we can do. The result is as the follows:

Group 1: $\{C_1, C_2, C_3, C_5, C_6, C_8\}$, *Group 2*: $\{C_4, C_7\}$, *Group 3*: $\{C_9\}$, *Group 4*: $\{C_{10}\}$

Figure 11 shows the output of our system.

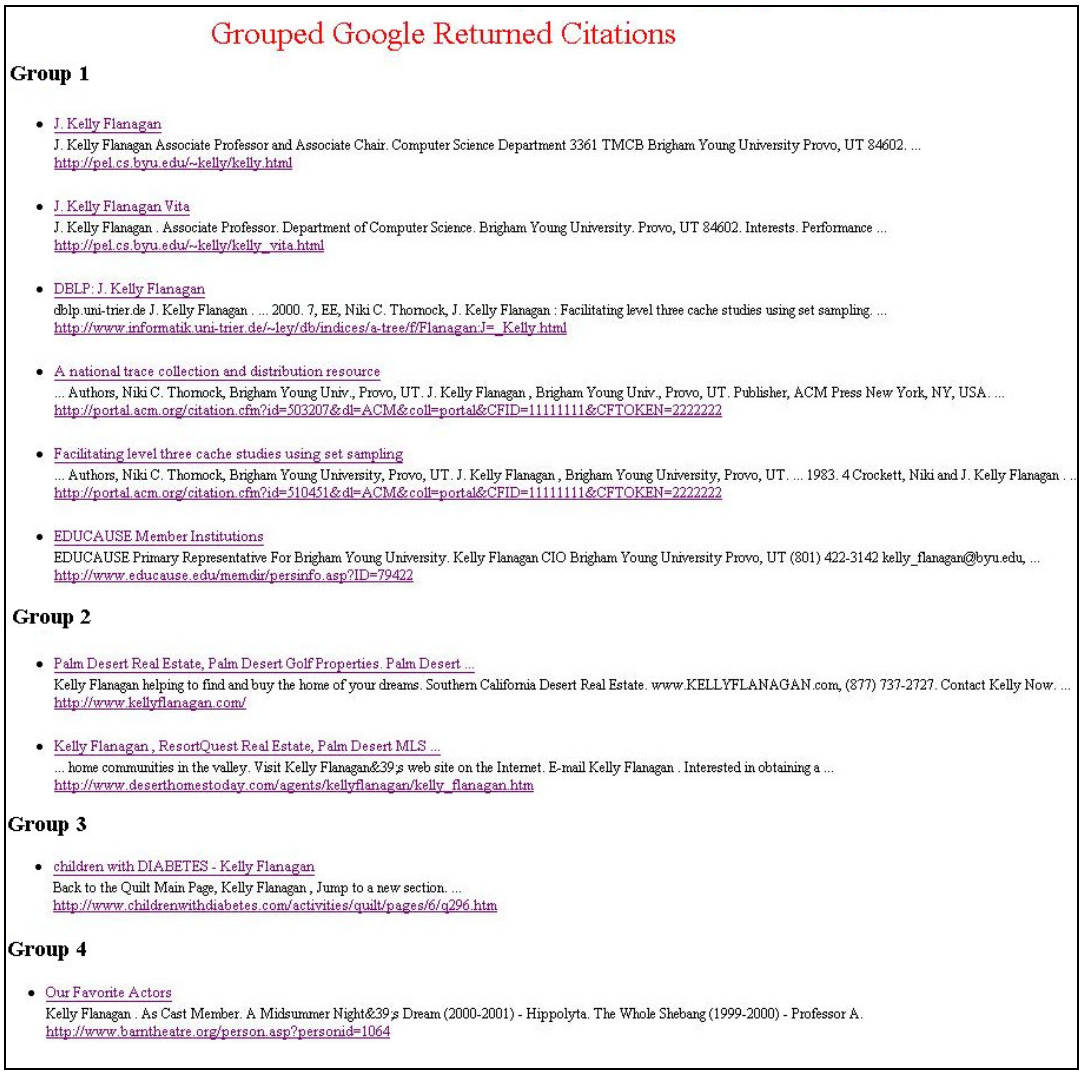

Figure 11: “ Kelly Flanagan” Grouping Result.

Chapter 4

Experimental Results and Analysis

To test our system, we chose 10 arbitrary different names. We chose the names by opening an arbitrary page from a phone book and choosing an arbitrary name from the page. The names were: Amanda Miller, Jared White, Steven Taylor, Susan Green, Christopher Young, Adam Wright, Jason Johnson, Lily Wu, William Barry, and Larry Wilde. We entered each name as a query in our system, and the system returned the grouping result for the first 50 returned citations for each name. Thus, the size of our test set was 500 citations.

To evaluate the performance of our system, we used split and merge measures, which are unique to this study, but similar to the idea of edit distance [RY98]. For each of the 10 returned result sets, we first counted how many splits we should do over all the groups to make the citations in each group relate to one person. Then, we counted how many merges we should do between the groups to ensure that no two groups relate to one person. For example, assume that the correct grouping result for eight returned citations C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8 is: *Group* 1: {C_1, C_2, C_4, C_6, C_7}, *Group* 2: {C_3, C_8}, *Group* 3: {C_5}, and the grouping result of our system is: *Group* 1: {C_1, C_2, C_4}, *Group 2* :{C_3, C_6, C_7}, *Group* 3: {C_5, C_8}. In order to fix the results that our system returns to match the correct results, we first split groups. We leave *Group 1* intact, we do one split of *Group 2* obtaining {C_3} and {C_6, C_7}, and we do one split of *Group 3*, obtaining {C_5} and {C_8}. Thus, the number of splits over all the citations is 0+1+1=2. Next we count how many merges are necessary. We should do one merge of {C_1, C_2,

C_4} with {C_6, C_7} and one merge of {C_3} with {C_8}. Thus, the total number of merges is 2.

Because the number of splits and merges can depend on the total number of citations, we normalized the split and merge scores to range between 0 and 1. To normalize a set of *n* returned citations, we divided by *n*-1 because the maximum number of splits or merges is *n*-1.

For each name (see Table 1) we obtained normalized split and merge scores for each and all facets by taking the average score across all the names. Table 1 shows that the average normalized score for splits for all facets is 0.004 and that the average normalized score for merges is 0.014. The results indicate that our system works well because the closer the split and merge scores are to 0, the better the performance. We also observe that no facet, by itself, performs as well as all facets together.

	All Facets		**Attribute Facet**		**Links Facet**		**Page Similarity Facet**	
Name	**Split**	**Merge**	**Split**	**Merge**	**Split**	**Merge**	**Split**	**Merge**
Amanda Miller	0.02	0	0	0.20	0	0.08	0.02	0.08
Jared White	0	0	0	0.43	0	0.14	0	0.08
Steven Taylor	0	0	0	0.20	0	0.14	0	0.08
Susan Green	0	0	0	0.20	0	0.04	0	0.08
Christopher Young	0	0	0	0.69	0	0.55	0	0.02
Adam Wright	0	0	0	0.10	0	0.14	0	0.12
Jason Johnson	0.02	0.04	0	0.24	0.02	0.38	0	0.06
Lily Wu	0	0.02	0	0.14	0	0.22	0	0.08
William Barry	0	0	0	0.06	0	0.04	0	0
Larry Wilde	0	0.08	0	0.82	0	0.65	0	0.20
Average	0.004	0.014	0	0.31	0.002	0.24	0.002	0.08

Table 1: Split and Merge Scores.

For all names except Amanda Miller and Jason Johnson, the split scores were 0 for all the facets together and all the individual facets. That means the citations in each

generated group (except two) related to the same person. In the case of Amanda, there was a group of two citations that should be split. The web pages referenced by the two citations shared three cap-word pairs: *Official College*, *Sports Network*, and *Student Advantage*. Since these pairs were not on our stop word list, the confidence value that the two citations refer to the same person in the attribute facet matrix is 0.92. This was the only non-zero confidence value which made the Stanford measure in the final confidence matrix also 0.92. Thus, our grouping algorithm grouped the two citations together. In the case of Jason Johnson, one citation that refers to a football player was merged with 14 citations that refer to a baseball player. This happened because the web page referenced by one of the 14 citations contains *www.pro-football-reference.com*, which is the host name of the citation that is related to the football player. According to our system the host name *www.pro-football-reference.com* is a non-popular host because the number of pages that link to it is less than 400. Thus, the confidence value for the links facet was 0.99, as was also the Stanford measure in the final confidence matrix.

Concerning merges, when we consider each individual facet, there were many merges needed for all names. When we used all facets together, however, the number of merges became 0 for all but three names and was close to zero for these three. Using a multi-faceted approach gave us a greater chance to gather evidence that two citations reference the same person or different persons. Thus using a multi-faceted approach gave much better performance than using each facet separately. The following paragraphs discuss the cases that caused missing merges when using each facet separately and when using all facets together.

For the attributes facet, there were two cases.

1. Web pages referenced by two citations that should have been merged did not share any attributes. In the 41 groups that should have been merged for Larry Wilde, for example, 1030 pairs (out of 1036 pairs) from distinct groups had no attributes in common.
2. Web pages referenced by two citations that should have been merged shared only a value for the attribute *State*. The confidence value to merge two citations knowing that the web pages referenced by them share only a *State* value is 0.49, which is less than our threshold value of 0.80. In the 41 groups that should have been merged for Larry Wilde, for example, 6 pairs from distinct groups shared only the *State* value.

For the links facet, there were four cases.

1. No link facet evidence was found between two citations that referred to the same person. In the 33 groups that should have been merged for Larry Wilde, for example, 1027 pairs (out of 1031 pairs) from distinct groups had no links facet evidence.
2. Two citations for the same person had only a popular common host. In the 19 groups that should have been merged for Jason Johnson, for example, 2 pairs (out of 208 pairs) from distinct groups had the same popular host name in common. One pair referred to the same person, and the other pair did not refer to the same person.
3. The web page of one citation contained a popular host of another citation for the same person. In the 19 groups that should have been merged for Jason Johnson,

for example, 6 pairs (out of 208 pairs) from distinct groups were such that in each pair a web page referenced by one of the two citations contained the host name of the URL of the other citation. All hosts were popular; 5 pairs referred to the same Jason Johnson, and one pair referred to two different Jason Johnsons.

4. Two citations had both of the previous cases. In the case of Larry Wilde, for example, there were 4 pairs such that the two citations in each pair had the same popular host and also a web page referenced by one citation contained the host name of the URL of the other citation and that host was popular. All 4 pairs referred to the same person.

For the page similarity facet, there were two cases.

1. Web pages referenced by two citations did not share any cap-word pair. In the 11 groups that should have been merged for Larry Wilde, for example, 417 pairs (out of 484) from distinct groups did not share any cap-word pair.
2. Web pages referenced by two citations shared one cap-word pair, and these two citations referred to the same person. The confidence value to merge two citations knowing that the web pages referenced by them share only one cap-word pair is 0.78, which is less than our threshold value of 0.80[5]. In the 11 groups that should have been merged for Larry Wilde, for example, 67 pairs from distinct groups share only one cap-word pair.

For all facets together, there were two cases.

[5] It would be tempting to just lower our threshold to 0.78, but our preliminary tests showed that lowering the threshold overly increased false merges. Thus, we left the threshold as generally determined before running our tests.

1. The confidence value between two citations in the final confidence matrix was less than our threshold value. In the case of Jason Johnson, for example, for the results when using all facets together we needed to merge a group of 15 citations, a group of 6 citations, and a group of one citation. Several pairs of citations from different groups that should have been merged shared one cap-word pair, but had no shared attributes and no links evidence. Thus, sharing only one cap- word pair with a confidence value of 0.78 made the Stanford measure in the final confidence matrix also 0.78, which was less than our threshold.
2. No evidence from any of the three facets was found between two citations in different groups that should have been merged. In the case of Larry Wilde, for example, we need to merge a group of 41 citations, 2 groups of two citations, and 2 groups of one citation in one group. For these 5 groups that should have been merged, none of the 259 pairs from distinct groups had any evidence they should have been merged. In the case of Lily Wu we needed to merge a group of 5 citations with a group of one citation. No two citations from these two groups that should have been merged had any evidence they should have been merged.

For groups that should have been merged, but no evidence or only weak evidence was found to group them, the question should arise, "How did the human expert decide to group them?" This also leads to the question, "Is there something more the machine could do to group them?" One technique the human expert used was to look at pictures (this technique is currently not possible for machines.) In the case of

Jason Johnson, for example, many citations from the different groups that should have been merged together contained a picture of the same baseball player. In the case of Larry Wilde two web pages that were referenced by 2 citations from one group that should have been merged shared the same picture with 2 citations from another group. Another technique the human expert used was to look for unusual distinctive characteristics. In the case of Larry Wilde, for example, 3 citations from 3 groups that should have been merged contained distinctive quotes: “Never worry about the size of your Christmas tree. In the eyes of chi...”, “Never worry about the size of your Christmas tree. In the eyes of children, they are all 30 feet tall.”, and “Christmas is the season when people run out of money before they run out of friends.” From looking at the first two quotes (even though the first quote was cut short) the human expert was able to easily judge that their citations referred to the same person. Since the third quote is about Christmas, the human expert guessed that its citation may relate to the other two citations. Note that we are not 100% sure that the human expert was always correct. A final technique the human expert used was a deeper understanding of the meaning of distinguishing phrases. In the case of Lily Wu, for example, the titles of web pages referenced by two citations of the two groups that should have been merged were “Lutheran Ministries in Higher Education” and “Lutheran Peace Fellowship”. Our cap-word pairs are not strong enough to detect these similarities, but with a deeper understanding it is reasonable to infer a match.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

We designed and implemented a system that can automatically group the returned citations from a search engine person-name query, such that each group of citations refers to the same person. We used a multi-faceted approach that considers three facets: attributes, links, and page similarity. We gave experimental evidence to show that our approach can be successful. In particular we tested 10 arbitrary names and found both a low normalized split score (0.004) and a low normalized merge score (0.014). The results also showed that no individual facet scored better than using all facets together. Thus, every individual facet and an appropriate combination of all facets appear to be necessary.

5.2 Future Work

There is reason to believe that it may be useful to adjust thresholds based on name popularity. John Smith is much more common than Stephen Liddle for example. To accomplish this research, we would first need to determine how to recognize if a name is popular or not. We would then need to determine how to set thresholds as a function of popularity.

It would be interesting to extend the research to deal with general proper-noun queries, which involve places and things as well as persons. The idea of using the multi-faceted approach would stay the same. We would, however, have to determine new attributes for each kind of proper noun. We would also have to obtain training data

and use it to establish the conditional probabilities. We may also need to adjust the threshold values.

Bibliography

[BYRN99] R. Baeza-Yates and B. Riberro-Neto. *Modern Information Retrieval*. Addison Wesley Longman, New York, New York, May 1999.

[BB98] A. Bagga, and B. Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Canada. June 1998, pp. 79–85.

[Dmo] DMOZ home page: http://dmoz.org/.

[Goo] Google home page: http://www.google.com/.

[Joa98] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning* Chemnitz, Germany, April 1998, pp. 137-142.

[HR97] T. Huang and S. Russell. Object Identification in Bayesian Context. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, August 1997, pp. 1276-1283.

[HR98] T. Huang and S. Russell. Object Identification: Analysis with Application to Traffic Surveillance. *Artificial Intelligence*, August 1998, Volume 103, Issue 1-2, pp. 77-93.

[LS97] G. Luger and W. Stubblefield. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Third Edition. Addison Wesley Longman, Reading, Massachusetts, September 1997.

[MY03] G. Mann and D. Yarowsky. Unsupervised Personal Name Disambiguation. In *Proceedings of the 7th Conference on Natural Language Learning*, Edmonton, Canada, May and June 2003, pp. 33-40.

[RY98] E. Ristad and P. Yianilos. Learning String Edit Distance. In *Proceeding of IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 20, Number 5, pp. 522-532, May, 1998.

[TKM01] S. Tejada, C. A. Knoblock, and S. Minton. Learning Object Identification Rules for Information Integration. *Information Systems*, Volume 26, Number 8, December, 2001, pp. 607-633.

[WM02a] D. Winchester and M. Lee, Cross-Document Co-Reference of Proper Names. In *Proceedings of the 5th Computational Linguistics in the UK.* Leeds, United Kingdom, January 2002.

[WM02b] D. Winchester and M. Lee. Using Proper Names to Cluster Documents. In *Acquiring (and Using) Linguistic (and World) Knowledge for Information Access: Papers from the spring Symposium (Technical Report SS-02-09)*, Menlo Park, California, 2002, pp. 3-8.

[OTC01] *Workshop on Operational Text Classification Systems*, New Orleans, Louisiana, September 2001.

[OTC02] *Workshop on Operational Text Classification Systems*, Tampere, Finland, August 2002.

[Yah] Yahoo home page: http://www.yahoo.com/.