Web People Search

Cameron L Palmer

December 9, 2007

1 Introduction

Searching for people on the Internet is of high interest to users of search engines; however, names are relatively common in relation to the size of the population. Since many people share the same name, the problem becomes how to tell one person from another. To quote the SemEval motivaton:

> Around 30% of search engine queries include person names. Person names, however, are highly ambiguous: for instance, only 90,000 different names are shared by 100 million people according to the U.S. Census Bureau.

In this project, the names provided are fairly common names that should be fairly difficult to disambiguate. The names provided are:

- Sarah Wilson
- Mary Johnson
- Helen Miller
- Christine King
- Brenda Clark
- Ann Hill
- Lisa Harris
- Samuel Baker
- Nancy Thompson

The goal of my project is to implement a first pass at clustering web pages in a way that, based on context, distinguishes between different people with same name. In other words, the program should be able to place web pages that relate to Sarah Wilson, the computer science student, in one cluster and place web pages that relate to Sarah Wilson, the veterinarian, in another cluster.

2 Dataset

The data provided was in the form of three files per name, all in XML. 'Pages' files consisted of the actual web pages. Each web page consisted of an XML wrapper with a

unique numeric identifier. The HTML was unfiltered and contained quite a lot of javascript and other typical junk contained in websites. This HTML code may or may not conform to HTML current standards, as is the case in the following example.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<WEPS:document_set name="Sarah_Wilson">
  <WEPS:document id="001">
    <![CDATA[
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;charset=iso-8859-1"><!DOCTYPE
 HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<HEAD>
        <TITLE>Authors: Sarah Wilson</TITLE>
<META NAME="date" CONTENT="11/23/04">
<!-- begin meta_info -->
<META NAME="date" CONTENT="November 23, 2004">
<!-- end meta_info -->
...
```

The basic problems associated with any HTML processing are apparent in this snippet; for example, the all-capital tags based on an older HTML standard, as well as the META tags carrying content. Text encoding is Latin-1, which is not a problem, but text encoding issues are a problem for any general solution. The majority of the page text is actually javascript, style information, and structural tags. I count 40 words in the body of the page text. Another problem is that the author's name in the text has the first character of Sarah replaced with an image of the letter 'S' instead of actual text. This example demonstrates that it is common to find a lot of noise and very little data on the web, and furthermore, what little data there is might not be machine readable.

The 'Data' files are XML-returned results from Google. Given a name input, the search engine returns numerous results like the following example:

```
<?xml version="1.0" encoding="utf-8"?>
<corpus search_string="Sarah Wilson">
  <doc rank="1" title="Authors: Sarah Wilson"
  url="http://www.twbookmark.com/authors/55/748/">
    <snippet>Sarah Wilson She is also a frequent lecturer on dog
    behavior, a journalist, video producer, and media commentator.
    Sarah Wilson lives in Bedford, New York.</snippet>
    <link rank="1"
    title="Chapter Excerpt: Good Owners, Great Dogs by Sarah Wilson and Brian"
    url="http://www.twbookmark.com/books/25/0446675385/chapter_excerpt8622.html">
      <snippet>Email this page to a friend! Sign Up to Receive Our
```

```
      Newsletters</snippet>
    </link>
    <related rank="1" title="Authors: Brian Kilcommons"
    url="http://www.twbookmark.com/authors/57/750/">
      <snippet>Brian Kilcommons orld-famous animal trainer Brian
      Kilcommons has appeard on Oprah, 20/20, and PBS&amp;amp;#39; The
      Gentle Doctor: Veterinary</snippet>
    </related>
    <related rank="2" title="Hotels at a Discount"
    url="http://www.smartnewyorker.com/hotel.htm">
      <snippet>The Best Hotel rates you will ever find. Discounted
      Hotels in New York and the Rest of the World. From Boston to
      LA, Canada to Texas, and London to Hong Kong. Check</snippet>
    </related>
    <related rank="3" title="FREE New York STUFF"
    url="http://www.smartnewyorker.com/freestuff.htm">
      <snippet>NEW YORK GUIDE TO FREE SUFF.</snippet>
    </related>
    <related rank="4"
    title="Tell Your Friends about http://www.twbookmark.com"
    url="http://www.tell-a-friend-wizard.com/cgi-bin/tell_opt_new2.cgi?uid=olmkt&amp;url
      <snippet>To send http://www.twbookmark.com to a friend, Just
      fill out this form and click Submit. To Name: To E-mail:*,
      From Name, From E-mail:*,</snippet>
    </related>
  </doc>
...
```

While I am not using this XML data in my project, it could be a source of information if you wanted to develop other solutions.

The 'Clustering' file is really the hand-annotated, gold standard for the clustering. This is how you judge your result. It is unfortunate that the files are missing id tags on some of the documents. I had to match up the answer by hand with some of the documents because the information is missing. This impacted my ability to achieve a more automated scoring. Following is an example of a partial cluster file:

```
<?xml version="1.0" encoding="utf-8"?>
<clustering>
  <cluster type="person" id="sarah_wilson-01">
    <page url="http://www.twbookmark.com/authors/55/748/"
    id="001" />
    <page url="http://search.barnesandnoble.com/booksearch/results.asp?ath=sarah+wilson"
    id="003" />
```

```
    <page url="http://thinks.com/cgi-bin/books/books.pl/mode-books/string-sarah+wilson/s
    id="007" />
    <page url="http://www.puppyworks.com/speaker/wilson.html"
    id="022" />
    <page url="http://books.lockergnome.com/sys/products/mode:books/search_type:authorse
    id="016" />
    <page url="http://www.ofspirit.com/tw-childproofingyourdog.htm"
    id="105" />
    <page url="http://www.frontlist.com/detail/0446516759"
    id="089" />
    <page url="http://www.lovegevity.com/aboutlovegevity/bio-swilson.html"
    id="045" />
    <page url="http://www.twbookmark.com/books/82/0446521515/press_release.html"
    id="035" />
    <page url="http://www.familydoginc.com/" id="038" />
  </cluster>
  ...
```

Note that the snippets for the data and clustering files indicate utf-8 encoding. However, the encoding was often not UTF-8. It is important to consider text encoding because the web is full of these sorts of problems and will break programs that expect perfect data.

3 Approach

My goal was to write a program that required no prior knowledge about the clustering to generate some level of automatic clustering. I abitrarily chose to perform 20 rounds of clustering to determine what sort of results could be achieved. I also wanted a very simplistic approach for my first attempt that didn't rely on URL information. Therefore, I chose the Bagga, Baldwin approach that was used for clustering documents.

The Vector Space information retrieval technique is used to measure the similarity between two documents based upon the bag of words in each text. To prepare the text you need to clean it up by stripping all HTML tags, punctuation, and stop words and by stemming the terms. Then you need to calculate the required values with the following formula:

$$Sim(S_1, S_2) = \sum_{\text{common terms } t_j} w_{1j} \times w_{2j}$$

where S is a document, j is a term common to both documents, and the weight w is calculated as,

$$w_{ij} = \frac{tf \times \log \frac{N}{df}}{\sqrt{s_{i1}^2 + s_{i1}^2 + ... + s_{in}^2}}$$

tf is the frequency of the term t_j in the document, N is the number of documents in the collection, and df is the number of documents in the collection that contain the term t_j. The square root is the *cosine normalization factor* and is equal to the Euclidean length of the vector S_i.

You run the formula for all pairs of documents and combined the pair with the greatest similarity. Then you perform the test again, and stop at some predetermined threshold. I set the threshold to 20 rounds of clustering.

4 Results

The Vector Space approach did not produce very good results when used with the web data set. In general, I got about 5 out of 20 attemtped rounds of clustering correct for each name. My target was to achieve a 50% accuracy. Clustering a pair of documents for evaluation should reinforce the similiarity as the matches build up, but it also broadens the 'appeal' of the document to match things it shouldn't. I base this on the fact that nearly every name generated a very large 9-10 group clustering with a couple of valid clusters, but 7-8 invalid clusters. The majority of clusters were comprised of two to three items. In the case of highly ambiguous names, and trying to disambiguate them with limited information, this Vector Space Model as implemented performs poorly.

The sort of data that causes this algorithm to fail, can be understood with an example. In looking at Helen Millers, Helen Miller Bailey has a library named for her, which has an unfortunate collision with Helen Miller Jones who also has a collection within a different library. This clustered with a third item which was a search for Helen Miller on a movie memorabilia website. The library collision can be understood because much of the page on the library revolved around how to check out books, and library policy which are largely identical. The movie memoribilia website was initially puzzling until you consider the site consisted of a failed search attempt for Helen Miller and the only real text on the page contained the words *special, request, copyright,* and *2004*, which matched up with the other two documents.

5 Future Work

To make a general solution that performs better than this approach, I need to evaluate some different techniques for maximizing the little information provided in the web page.

This means that things like tags are important information and must not be thrown out since they convey the hierachy or relationship between the information on the page. URL and link information on a page must also be preserved and put to work. If the HTML structural tags represent hierarchy of information on a page, the links represent hierarchy of information in relation to each other. It may be the case that the information contained in the links is actually a manual annotation of clustering information. In surveying the clustering data, you could simply look at the domain name and cluster many more documents.

Within the text, attempts to use features around the word in a addition to the bag of words may prove a better strategy. In addition to web pages being fairly data sparse, the pages also often contain a lot of unrelated data. Therefore, it will be important to improve result based upon words close to our target. Referring back to the Helen Miller example, if we were using exclusively features, two before and two after the target, we would have likely not clustered the pages.

The Web People Search problem could benefit from a tool that automatically allows you to evaluate different algorithms and have it pull together all the clustering information. I hand-edited HTML pages and the clustering information, including metric; however, if you could enter the items your algorithm clustered and the program would automatically generate HTML, open it in a browser, and allow you to display the values associated with words on that page, it might be easier to get an idea of how to proceed with this problem. For example, if you could see the web page and automatically highlight the words that match other documents, you might be able to make a better decision on whether these matches are relevant. Also, by turning stemming on and off, performing cross document word matchings, you might be able to develop a better intuition of what would or would not work for this problem.

Given the SemEval 2007 scoring system, it would be helpful to measure results through this software, rather than evaluating the results by hand.