Using Web Graph Structure for Person Name Disambiguation

Elena Smirnova[1], Konstantin Avrachenkov[2], and Brigitte Trousse[1]

[1] AXIS research team,
[2] MAESTRO research team,
INRIA Sophia Antipolis - Méditerranée,
2004 route des Lucioles, 06902 Sophia Antipolis Cedex, France
{elena.smirnova, konstantin.avrachenkov, brigitte.trousse}@inria.fr

Abstract. In the third edition of WePS campaign we have undertaken the person name disambiguation problem referred to as a clustering task. Our aim was to make use of intrinsic link relationships among Web pages for name resolution in Web search results. To date, link structure has not been used for this purpose. However, Web graph can be a rich source of information about latent semantic similarity between pages. In our approach we hypothesize that pages referring to one person should be linked through the Web graph structure, namely through topically related pages. Our clustering algorithm consists of two stages. In the first stage, we find topically related pages for each search result page using graph-based random walk method. Next, we cluster Web search result pages with common related pages. In the second stage, Web pages are further clustered using content-based clustering algorithm. The results of evaluation have showed that this algorithm can deliver competitive performance.

Keywords: Person name disambiguation, Web graph, Related pages

1 Introduction

Information search is a fundamental task on the Web. Among different types of information needs, information about people is frequently demanded by users [14]. In our scenario, a user wants to retrieve information about the person by his/her name. A typical response of a Web search engine consists of a set of Web pages that contain the name but can refer to different persons. Indeed, data from U.S. Census Bureau indicates that person names are highly ambiguous - approximately 90,000 names are shared by 100 million people (as cited by [3]). To assist user in finding the target person, it was proposed to cluster Web pages returned by search engine according to different people sharing the name query. Many studies in this direction have been done, in particular, within Web People Search (WePS) initiative [1, 2].

Following previous successful editions, WePS-3 evaluation campaign featured the name resolution (clustering) task for the third time. As before, the goal of the task was to group Web search result pages returned for name query according to found namesakes. The test data was composed of Web search results for each name including URL, title, rank information, search snippet and HTML content. The evaluation was done using extended versions of BCubed precision and recall [2].

We participated in WePS-3 campaign with the following approach. We analyzed patterns of Web graph structure in application to person name resolution task. Our key idea was to leverage of link relationships among Web pages for name resolution. To date, Web pages appeared in the search result for name query have been considered as independent. However, recent work in Web information retrieval [7, 8, 10, 12] suggests that link structure of the Web can be a rich source

of information about latent semantic similarity between pages. For instance, an assumption that a group of similar pages is likely to be closely linked is generally taken by numerous works in Web community discovery. Analogously, in our approach we hypothesize that pages referring to one person should be linked through the Web graph structure, namely through topically related pages.

We perform a two-stage clustering algorithm. In the first stage, we find related pages for each Web search result page using graph-based random walk method. Next, we cluster Web search result pages with common related pages. The resulted clustering is therefore built using only Web link information. In the second stage, Web pages are further clustered using content-based clustering algorithm. More particularly, we build a term profile with frequency score for all pages including related. Then we re-weight terms in each search result page profile according to its related pages profiles. Finally, we apply Hierarchical Agglomerative Clustering algorithm to the set of un-clustered Web page profiles.

From experiments we found that Web structure based clustering itself can quite successfully disambiguate Web search results. Further improvements can be achieved by considering the content of the pages. The results of evaluation have showed that our algorithm can deliver competitive performance.

The rest of the paper is organized as follows. In the next section we describe an idea of related pages and our approach for finding related pages. In Section 3 we present the detailed description of the algorithm. The runs that we carried out are described in Section 4, while the results of evaluation of those runs are detailed in Section 5. We conclude in Section 6.

2 Related pages

2.1 Motivation

In the following we explain our motivation behind the use of Web graph structure to disambiguate the referents. We start by defining a *related page* as the one that addresses the same topic or one of the topics mentioned in a *person page* - page that contain a person name. For example, given personal Web page of a scientist, related pages would include co-authors pages, conference and project pages where scientist has participated, department pages that scientist belong to. Ideally, if Web links would reflect semantic relationship between pages, topically related pages of a person page could be found in its graph-based neighborhood. Moreover, we would observe person pages referring to one person interconnected through their related pages. Indeed, an assumption that similar pages are likely to be closely linked is generally taken by numerous works in Web community discovery [6, 11].

Kleinberg in [8] has given an illustrative example that ambiguous senses of the query can be separated on a query-focused subgraph. Specifically, several densely linked parts of the graph can be uncovered using non-principal eigenvectors of AA^T, where A is a subgraph adjacency matrix. Author suggested building query-focused subgraph using semantically intrinsic forward and backward links of Web search result pages. In our context, we can name the pages pointed to by those links as semantically related. Therefore, we can form a hypothesis that for ambiguous name query linked parts of the semantic subgraph form clusters corresponding to different individuals.

2.2 Implementation

The major problem of applying HITS [8] algorithm and other community discovery methods to WePS dataset consists in the lack of information about Web graph structure. In particular, full

information about Web page backward links is not available without crawling the main part of the Web graph.

Personalized PageRank (PPR) [7] can be used to detect related pages of target page [12]. In this work [12], the personalization vector is a unit vector with all elements equal to zero and the entry corresponding to the target page equal to one. Theoretical and experimental results showed that, quite opportunely, Monte-Carlo method is a fast way to approximate top-k set of pages with the largest value of Personalized PageRank in a local manner, i.e., using only page forward links [4]. The lazy nature of Monte-Carlo iteration can be seen as a considerable advantage in terms of storage and time resources against methods requiring knowledge of Web graph structure. Moreover, Monte-Carlo method is highly parallelizable which reduces computational time on a cluster of computers. Since Personalized PageRank computes related pages of target page using only local forward-link information, globally related backward-link pages are usually missing. Therefore, generally we cannot expect overlap in related pages sets of two pages referring to one person - that would require global structure of the graph. Nevertheless, we found useful to examine content of related pages.

Alternatively to Personalized PageRank, we also consider related pages offered by Google service[3]. Although the algorithm is proprietary, the main idea expressed in [10] was formulated as finding pages frequently co-cited with a target page. Relying on this explanation, the computation of related pages involves both backward and forward links of a page. Therefore, we consider Google related pages to be based on global Web graph information.

3 System Description

3.1 Overview

An overview of our approach to name resolution problem is presented in Figure 1. In the first stage, we cluster person pages appeared in search results based on Web structure. Thereto, we determine related pages of each person page and then cluster person pages that share some of related pages in one cluster. We consider this clustering as *Web structure based* since it is formed based on link relationship. As the entire link structure of the Web is unknown to us, some global topically related pages are missing during Monte-Carlo random walk process. Due to this introduced sparseness, we perform the second stage clustering where the rest of the person pages that did not show any link preferences are clustered *based on the content*. More particularly, we build a term profile with frequency score for all pages including related. We re-weight terms in each person page profile according to its related pages profiles. In this process weights of terms that appear in the related page profile are increased. Finally, we apply Hierarchical Agglomerative Clustering (HAC) algorithm to the set of un-clustered Web pages profiles.

3.2 Web Structure based Clustering

Related pages. In the first implementation we compute related pages of person page using Personalized PageRank. While we assumed that the presence of the link between pages implies their semantic relationship, there are links that exist purely for navigational purpose. To avoid negative effect of these links we perform random walk of Monte-Carlo computation on links to pages with host name different from current. By host name we mean the first level in the URL string associated with the link. We found this heuristic useful since links within one domain typically serve a navigational function rather than indicate semantic similarity. In addition, we

[3] Google search 'related:' operator. http://www.google.com/intl/en/help/operators.html#related

lower the probability to follow the link that points to a page with high in-degree as inverse proportional to natural logarithm of in-degree. For example, main pages of large portals like Wikipedia or IMDB are universally popular and so, have a high number of incoming links that however do not necessarily carry semantic relationship. We note that at this point we employed a type of global information - the number of incoming links - that we found indispensable to avoid pathologies caused by the high value of global PageRank attributed to large portals. We requested the number of backward links for a given page from Google search engine[4].

We estimated top K set of related pages for each person page. In experiments we used two values of $K = \{8, 16\}$ and hence, two settings of Personalized PageRank computation. For $K = 8$ we set the number of iterations equal to 2000 and damping factor c equal to 0.2. In the second setting for $K = 16$ we doubled the number of iterations to 4000 and increased damping factor to 0.3.

An example of top 8 related pages list for personal Web page of a scientist is given in Table 1. For illustrative purposes related pages returned by Google service for the same person page are given in Table 2. Clearly, Web pages computed by Personalized PageRank refer to different topics related to the person but not necessarily contain the query-name in the content. These pages are homepages of current and previous workplaces, Web pages of co-authors and scientific activities undertaken by the person. Quite differently, a related pages set provided by Google contains pages of the scientist on large portals such as LinkedIn, DBLP and Videolectures. It is unlikely that these pages could be interconnected by short forward-link path and thus, it explains their absence in Personalized PageRank list.

Graph clustering. In the following step two person Web pages are merged in one cluster if they share some related pages. Since the whole link structure of the Web is unknown to us, related pages set is limited to pages reachable by forward links from a person page. We therefore address to the content of the pages in the next stage.

3.3 Content based Clustering

Page profile. Preprocessing of Web pages include the following steps. First we convert Web pages into plain text using Apache HTML parser[5]. In one implementation we extract the full text of the page, while in the other we keep only the content of META tags. Next, we apply clean-up procedure. To distinguish the main content from navigational text, advertisements, etc. we use a simple heuristic that meaningful text in the page consists of at least 10 consecutive terms [9]. We consider as a term a sequence of letters and numbers of length more than one. Terms are stemmed using Porter's stemmer [13] and removed if presented in standard English stopword list.

We apply preprocessing step to all pages including person pages and pages related to them. Next, for each of these page we build a vector of terms with corresponding frequency score (tf) in the page. After that, we use a re-weighting scheme as follows. The term t score at person page p, $tf(p,t)$, is updated at each related page r in the following way:

$$tf'(p,t) = tf(p,t) + tf(p,t) * tf(r,t),$$

where r is in the related pages set of person page p and person page p is the one from search results. This step resembles voting process. Terms that appear in related pages get promoted and thus, random term scores found in the person page are lowered. At the end, vector is normalized and top 30 most frequent terms are taken as a person page profile.

[4] Google search 'link:' operator. http://www.google.com/intl/en/help/operators.html#link

[5] http://htmlparser.sourceforge.net

HAC clustering. Finally, we apply HAC algorithm on the basis of clustering from the first stage to the rest of the Web page profiles. Specifically, average-linkage HAC with cosine measure of similarity was used. Following previous work [5], the similarity threshold for HAC algorithm was fixed to 0.1.

Algorithm 1 Person Name Disambiguation using Web Graph Structure

Input: Web search results

1. Web structure based clustering.

for all page $t \in$ search results **do**
  compute related pages of t
end for
$C' \leftarrow$ cluster search result pages with shared related pages

2. Content based clustering.

for all page $t \in$ search results **do**
  build term profile of t
  for all page $r \in$ related(t) **do**
    build term profile of r
    update term profile of t
  end for
end for
$C \leftarrow$ cluster search result pages using HAC based on C'

return C

Fig. 1. Pseudocode of Person Name Disambiguation algorithm.

4 Runs

During evaluation period of WePS campaign we experimented with two ways to compute related pages, the number of related pages and the type of content extracted from pages. We have chosen to combine less number of related pages with the full content of the page and, the other way, larger number of related pages with less extracted content. Experimentally we found that with larger lists pages might be quite loosely related to the person page and might promote irrelevant terms in the person page profile. Therefore, we limited analyzed content of pages to a few words in meta tag description, title and snippet.

We submitted the following runs to clustering task. The run name in brackets is the name in official evaluation results.

PPR-HAC (AXIS_4). Top 8 related pages were computed using Personalized PageRank, the full content of the Web page was used in HAC step.

Table 1. Example of related pages set returned by PPR computation for Cyril Goutte Homepage.

URL
National Research Council Canada - current workplace
NRC for IT, Canada - current workplace
LT Research Center at NRC Canada - current workplace
Xerox Research Center Homepage - previous workplace
John Hopkins University Homepage - visitor of CLSP
CLSP workshop Homepage - team member
George Foster Homepage - co-author at Xerox
Simona Gandrabur Homepage - co-author at JHU

Table 2. Example of related pages set returned by Google service for Cyril Goutte Homepage.

URL
Old Cyril Goutte Homepage
Cyril Goutte LinkedIn profile
Book by namesake of Goutte on handicap accessibility
Book by Goutte et al. at MIT press
Cyril Goutte page at ScientificCommons.org
Cyril Goutte page at DBLP
Cyril Goutte page at Videolectures.net
Review of the book by Goutte et al.

G-HAC (AXIS_3). Top 8 related pages were provided by Google service, the full content of the Web page was used in HAC step.

G-HACext (AXIS_2). Top 16 related pages were provided by Google service, the content of meta tag description, title and snippet was used in HAC step.

PPR-HACext (AXIS_1). Top 16 related pages were computed using Personalized PageRank, the content of meta tag description, title and snippet was used in HAC step.

We also carried out four baseline runs:

HAC. The baseline method where HAC algorithm was applied to the content of Web pages. No link structure based clustering was performed.

PPR. The baseline method where clustering was based on top 8 related pages computed using Personalized PageRank. No content-based clustering was performed after.

G. The baseline method where clustering was based on top 8 related pages provided by Google service. No content-based clustering was performed after.

PPR-G. The baseline method where clustering was based on merged set of top 8 related pages computed using Personalized PageRank and provided by Google service. No content-based clustering was performed after.

Two other baselines were provided by organizers:

One-in-one. The baseline method where every Web page is assigned to a different cluster.

All-in-one. The baseline method where all Web pages are assigned to a single cluster.

5 Results

Table 3 shows the results achieved by our methods. Concerning the baseline runs, we note that performance of link-based clustering methods is quite notable. Taking Web structure as the only input information, it is possible to deliver the same or superior performance as when processing the content. Remarkably, the combination of related pages computed using Personalized PageRank and obtained from Google service achieved the highest score among link-based baselines. We see that link-based baselines are effective in terms of precision, while content-based baseline showed higher recall values. High precision value of link-based baselines indicates that sharing related pages between two pages is a strong evidence to their semantic similarity. In this case, an improvement of recall value is attributed to the problem of finding strong Web graph paths between two pages. We also found link-based methods benefical in clustering flash or image made pages with little text where content-based algorithm experiences difficultes.

The significant improvement over the baseline methods (16-35% F-0.5) has been achieved by combination of link-based and content-based clusterings. However, we have not found any significant difference among submitted combinations (as indicated by two-tailed paired t-test). All submitted runs are characterized by higher BCubed precision value compared to recall. As we noted above, the more balanced result could be obtained by discovering more link relatioships among pages in the Web graph. Results indicate that all submitted runs improved the corresponding baselines. This observation is confirmed by two-tailed paired t-test at significance level $\alpha < 0.001$ for all submitted runs against corresponding link-based and content-based baselines.

The official performance ranking over WePS-3 participants showed that our algorithm took the second place (in F-0.5 measure) among 8 competitors with in total 27 submitted runs. In addition to achieved performance, we note that our algorithm can be efficiently implemented due to parallel nature of Monte-Carlo computation.

Table 3. Results for the clustering task. The metrics are: BCubed precision (BP) and recall (BR), harmonic mean of BP and BR (F-0.5). The run name in brackets is the name in official evaluation results.

Run	**BP**	**BR**	**F-0.5**
PPR-HAC (AXIS_4)	0.7	0.45	0.5
G-HAC (AXIS_3)	0.68	0.46	0.5
G-HACext (AXIS_2)	0.69	0.46	0.5
PPR-HACext (AXIS_1)	0.71	0.43	0.49
PPR-G	0.78	0.34	0.43
HAC	0.43	0.4	0.41
PPR	0.93	0.27	0.39
G	0.87	0.27	0.37
One-in-one	1	0.23	0.35
All-in-one	0.22	1	0.32

6 Conclusion

We have described our approach to person name disambiguation task at WePS-3 evaluation campaign. Our main idea was to make use of patterns of Web structure to disambiguate Web

search results. From our experiments we concluded that it is possible to quite successfully resolve a person name using Web structure as the only input information. Web structure based clustering showed to be effective in terms of precision. Significant improvement of recall value (+60%) was achieved by combining link-based and content-based clusterings. However, we did not found any significant difference in behaviour between different combinations. Overall we concluded that our algorithm can deliver competitive performance in comparison with other systems participated in WePS-3 campaign. In addition to that, our algorithm can be implemented efficiently and is suitable for use within a large-scale Web service.

7 References

[1] Artiles, J., Gonzalo, J., Sekine, S.: The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007) (2007)

[2] Artiles, J., Gonzalo, J., Sekine, S.: Weps 2 evaluation campaign: overview of the web people search clustering task. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference (2009)

[3] Artiles, J., Gonzalo, J., Verdejo, F.: A testbed for people searching strategies in the www. In: SIGIR'05 (2005)

[4] Avrachenkov, K., Litvak, N., Nemirovsky, D., Osipova, N.: Monte carlo methods in pagerank computation: When one iteration is sufficient. SIAM Journal on Numerical Analysis 45(2), 890–904 (2007)

[5] Balog, K., He, J., Hofmann, K., Jijkoun, V., Monz, C., Tsagkias, M., Weerkamp, W., de Rijke, M.: The university of amsterdam at weps2. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference (2009)

[6] Flake, G., Lawrence, S., Giles, L.: Efficient identification of web communities. In: KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 150–160. ACM, New York, NY, USA (2000)

[7] Haveliwala, T.: Topic-sensitive pagerank. Proceedings of the 11th WWW Conference pp. 517–526 (2002)

[8] Kleinberg, J.: Authoritative sources in a hyperlinked environment. J. ACM 46(5), 604–632 (1999)

[9] Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: WSDM '10: Proceedings of the third ACM international conference on Web search and data mining. pp. 441–450. ACM, New York, NY, USA (2010)

[10] Law, K.L., Harik, G.R.: Techniques for finding related hyperlinked documents using link-based analysis (December 2009), assignee: Google Inc.

[11] Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: WWW '08: Proceeding of the 17th international conference on World Wide Web. pp. 695–704. ACM, New York, NY, USA (2008)

[12] Ollivier, Y., Senellart, P.: Finding related pages using green measures: an illustration with wikipedia. In: AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence. pp. 1427–1433. AAAI Press (2007)

[13] Porter, M.F.: An algorithm for suffix stripping pp. 313–316 (1997)

[14] Wan, X., Gao, J., Li, M., Ding, B.: Person resolution in person search results: Webhawk. In: CIKM'05 (2005)