Appeared in: IEEE International Conference on Data Engineering (IEEE ICDE), April 16-20, 2007.

Disambiguation Algorithm for People Search on the Web

Dmitri V. Kalashnikov Sharad Mehrotra Zhaoqi Chen Rabia Nuray-Turan Naveen Ashish
Department of Computer Science
University of California, Irvine

1 Introduction

Searching for entities, i.e., webpages related to a person, location, organization or other types of entities is a common activity in internet search today. For instance "people search" i.e., searching for webpages related to a person accounts for over 5% of the current Web searches [4]. Entity search today is done using keywords where a search engine such as Google or Yahoo returns a set of Web pages, in ranked order, that are deemed relevant to the search keyword entered (the person name in this case).[1] A search for a person, such as say "Andrew McCallum" will return pages relevant to *any* person with the name Andrew McCallum.

We envision a next generation search engine that can provide significantly more powerful models for entity search. Assume (for now) that for each such Web page the search-engine could determine which real entity (i.e., *which* Andrew McCallum) the page refers to. This information can be used to provide a capability of *clustered* entity search where instead of a list of Web pages of (possibly) multiple persons with the same name, the results are clustered by association to real person. The clusters can be returned in a ranked order determined by aggregating the rank of the Web pages that constitute the cluster. With each cluster we also provide a summary description that is representative of the real person associated with that cluster (for instance in this example the summary description may be a list of words such as "computer science, machine learning, professor"). The user can hone in on the cluster of interest to her and get all pages in that cluster, i.e., only the pages associated with *that* Andrew McCallum.

There is significant interest in the problem of Entity Search, with several research efforts addressing this and related challenges. The motivation for that is the fact that Entity Search can provide a way to browse and analyze the returned information in a more structured way, ultimately enhancing web search capabilities and the user experience. For instance, imagine searching for the webpages of a person who happened to have a famous namesake. This can be very tiring since the first several pages of the corresponding Google search returns pages only about the famous person. In the clustered approach, all of the famous person's pages will be folded into a single cluster giving his namesakes a chance to be displayed in the first page of search results.

While the example above shows the clustered approach in a positive light, in reality, it is not obvious that it indeed is a better option compared to searching for entities using keyword-based search supported by current search engines. The reason is that clustering algorithms can make mistakes and assign webpages to the wrong clusters. The key issue is the quality of clustering algorithms in disambiguating different web pages of the namesakes.

In this paper we develop a disambiguation algorithm and then study its impact on People Search. The proposed algorithm first uses extraction techniques to automatically extract 'significant' entities such as the names of other persons, organizations, and locations on each webpage. In addition, it extracts and parses HTML and Web related data on each webpage, such as hyperlinks and email addresses. The algorithm then views all this information in a unified way: as an Entity-Relationship Graph where entities (e.g., people, organizations, locations, webpages) are interconnected via relationships (e.g., 'webpage-mentions-person', relationships derived from hyperlinks, etc). The algorithm gains its power by being able to analyze several types of information: *attributes* associated with the entities (e.g., TF/IDF for webpages) and, most importantly, direct and indirect *interconnections* that exist among entities in the ER graph. We next outline our approach in Section 2 and then compare it with the state of the art solutions in Section 3.

2 Approach Overview

Architecture. There are several possible ways for implementing People Search and we take the middleware based approach. Given a query (a person name) the middleware submits the query to the standard search-engine and selects a fixed number (top K) of the results. A disambiguation algorithm is then applied to those pages. The result is a set of clusters of these pages with the aim being to cluster Web

[1] There are other people search services, e.g. http://find.intelius.com, that provide "background information" about people, e.g. addresses. Our focus instead is on *webpages* relevant to a person on the public Internet.

pages based on association to real person. Given these clusters the system returns clusters to the user in a ranked order with the rank based on some chosen criteria.[2] If the user explores a particular cluster from the set, then she first sees the webpages returned by the clustering algorithm. They are followed by the rest of the webpages from the set, sorted based on their similarity to the cluster. That is, the search is forgiving, and the user has the chance to examine all of the K webpages.

Disambiguation Algorithm. Disambiguation approaches do of course exist for a variety of data management applications and the approaches themselves can be classified along a variety of facets. One of these facets, of interest in this context, is the type of information the approach is capable of analyzing in making its co-reference decisions. The proposed disambiguation algorithm is based on analyzing *two* types of information. First, it analyzes object features, like many other techniques. Second, (most important) it also analyzes the Entity-Relationship Graph (ER graph) for the dataset.

The idea behind analyzing features of objects u, v is based on the assumption that similarity of features of two objects defines certain affinity/attraction between those objects $f(u, v)$. If this attraction $f(u, v)$ is sufficiently large, then the objects are likely to be the same (co-refer). The intuition behind analyzing paths in the ER graph is similar. The assumption is that each path/connection/link p between two objects u, v can serve as *evidence* that they co-refer. So if the combined evidence, stored in all the u-v paths, is sufficiently large, the objects are likely to be the same. An in-depth insight into the motivation for this methodology is elaborated in [5]. Formally, the attraction between two nodes u and v via paths is measured using the *connection strength* measure $c(u, v)$ which is defined as the sum of attractions contributed by each path: $c(u, v) = \sum_{p \in P_{uv}} c(p)$. Here P_{uv} denotes the set of all simple paths between u and v (of limited length), and $c(p)$ is the contribution of path p.

The proposed algorithm is capable of learning $c(p)$ from (past) data, thus tuning itself to a given domain. That is, it employs an *adaptive* connection strength model, instead of a fixed one. It then applies correlation clustering techniques, which employ both $f(u, v)$ and $c(u, v)$, in order to produce the final grouping of the webpages. Due to space limit, we omit the details of that algorithm.

ER Graph. An interesting peculiarity in applying the proposed disambiguation algorithm to Web data is that entities

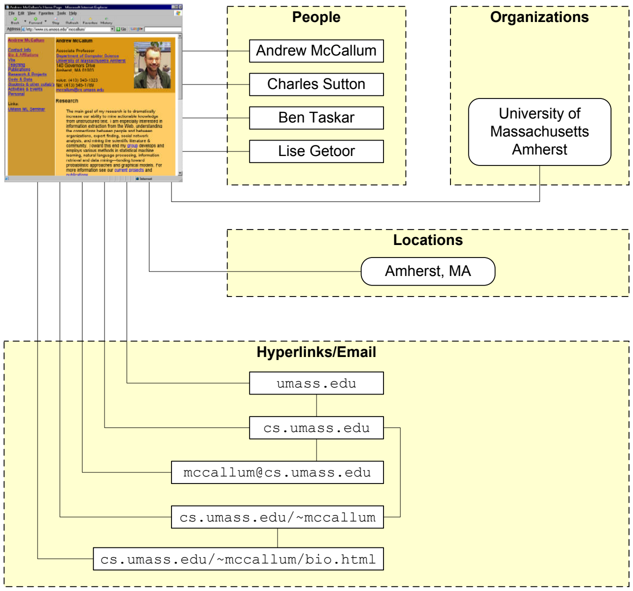


Figure 1. Extraction.

and relationships, which the algorithm employs in its analysis, are not readily available in the dataset for use. Rather such entities and relationships need to be first *extracted* off the Web pages. We do that using information extraction (IE) software. In addition to Named Entities (NEs), we also extract hyperlinks and email addresses from the Web pages, see Figure 1. A node is created for each extracted entity (person, organization, location), each of the (top K) Web pages, each extracted url/email and the derivatives of their decompositions (domains, subdomains, etc). A relationship edge is created between a node representing a Web page and each the nodes corresponding to each item extracted from that Web page. Edges are also created for the url decompositions, e.g. 'subdomain-of', see Figure 1. At the end of this process we have a complete graph representation of the information that a clustering or disambiguation algorithm can now work with. The algorithm is now abstracted from any of the extraction details and can in fact self-tune itself to optimize based on the nature of the graph.

Ontology-Enhanced TF/IDF. We use an ontology (DMOZ) to enhance the $f(u, v)$ part of the algorithm, which is computed using TF/IDF. The ontology is applied to derive concepts; e.g. 'machine learning' is grouped into one concept 'machine_learning'. Also, when a concept in a webpage is found in the ontology, the webpage is enriched with the corresponding classification terms, on some conditions.

3 Experimental Results

We report experiments on two real datasets used by Javier Artiles et al. in SIGIR'05 [1] (Table 1) and by Ron

[2] We use the following ranking method. The original order in which each page is returned by Google is known. For each resulting cluster, we find the webpage with the lowest Google order. We order clusters based on the order of those pages. This achieves two goals. First, the clusters are ordered roughly according to the Google-computed importance. Second, it can be proven that when the clustering is perfect, under certain search scenarios, the amount of work required to locate the relevant pages cannot be worse than that of using Google.

Name	#	K	$\mathbf{F_{0.5}}$	$\mathbf{F_{0.2}}$
Ann Hill	55	58	93.1(+5.1)	91.5(+3.5)
Brenda Clark	23	20	85.1(-2.9)	92.0(+7.0)
Christine King	29	52	84.4(+17.4)	81.9(+11.9)
Helen Miller	38	31	70.8(+8.8)	78.2(+18.2)
Lisa Harris	30	45	84.8(+1.8)	82.7(-0.3)
Mary Johnson	54	47	89.5(+14.5)	92.2(+9.2)
Nancy Thompson	47	57	88.5(+7.5)	91.9(+10.9)
Samuel Baker	38	23	79.3(+0.3)	85.3(-1.7)
Sarah Wilson	62	59	91.1(+21.1)	90.6(+9.6)
Mean/Overall	**42**	**44**	**85.2(+8.2)**	**87.4(+7.6)**

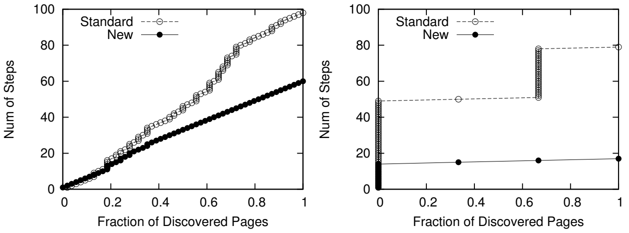
Table 1. SIGIR'05 Dataset.

Bekkerman and Andrew McCallum in WWW'05 [2] (Table 2). Both datasets have been collected similarly and then hand labeled to distinguish among the namesakes. First, Google is queried with a person name, e.g. 'Andrew McCallum'. Then only the top 100 webpages are considered among the webpages returned by Google. From Table 1 we can see that 9 person names are queried in SIGIR'05 dataset and 12 names in WWW'05 dataset. The '#' field shows the number of namesakes for a particular name in the corresponding 100 webpages. The field 'K' is the number of clusters the proposed disambiguation algorithm computes. Ideally K should match the number of the namesakes. $F_{0.5}$ is the harmonic mean of the Purity and Inverse Purity measures. $F_{0.2}$ is similar to $F_{0.5}$, but gives more preference to the Inverse Purity, see [1] for the motivation of these measures. The parameters of our disambiguation algorithm are set via leave one out cross validation. The values in the brackets in those tables show the improvement over the approaches in [1, 2]. The improvement is achieved since the proposed approach is simply capable of analyzing more information, hidden in the datasets, and which [1, 2] do not analyze. The proposed approach outperforms [1] by 8.2% wrt $F_{0.5}$, and by 7.6% wrt $F_{0.2}$. In [2], the authors solve a related-but-different problem, than the one studied in this paper. We modified our algorithm to apply it to that problem: the algorithm outperforms [2] by 9.5% of F-measure, see Table 2.[3] The field '#W' in Table 2 is the number of the to-be-found webpages related to the namesake of interest.

Name	#	$\mathbf{F_{0.5}}$	$\mathbf{F_{0.2}}$	#W	F-measure
Adam Cheyer	2	98.4	97.5	96	98.4(+19.9)
William Cohen	10	83.8	76.8	6	66.7(-8.3)
Steve Hardt	6	79.6	71.3	64	75.7(+36.7)
David Israel	19	84.0	81.1	20	82.1(-6.3)
Leslie Kaelbling	2	98.3	97.3	88	98.3(+1.2)
Bill Mark	8	76.6	83.8	11	77.8(+31.6)
Andrew McCallum	16	96.3	95.3	54	100.0(+1.8)
Tom Mitchell	37	83.8	80.4	15	84.6(+2.3)
David Mulford	13	86.7	87.0	1	0.0(-100.0)
Andrew Ng	29	85.3	82.3	32	75.4(-12.8)
Fernando Pereira	19	77.9	71.8	32	76.2(+13.5)
Lynn Voss	52	85.3	89.9	1	0.0(+0.0)
Mean/Overall	**18**	**86.3**	**84.5**	**35**	**89.8(+9.5)**

Table 2. WWW'05 Dataset.

Impact on Search. To assess the impact of disambiguation on search, we test the following scenario. A user queries the search engine with the name of the person of interest, e.g. 'Andrew McCallum'. The user then scans through top K pages in order to satisfy her objective of finding all the webpages of that person among the top K pages. The user can use the traditional or the new interface, where she first sees clusters and then can examine the webpages inside each cluster, using the forgiving search. We measure how many steps the user needs to do to discover a certain fraction of the all webpages of a particular namesake of her interest. Figure 2 (a) plots that measure for Andrew McCallum the UMass Professor. His pages tend to appear first in Google, they form the first group, which is also the largest one. Figure 2 (b) plots the same measure for Andrew McCallum the Customer Support person. His cluster consists of 3 pages that appear more toward the end in Google search. His group is one of the last groups. Both figures demonstrate that in these two scenarios the user can locate the desired webpages faster when using the new interface.



(a) UMass Professor. (b) Customer Support.

Figure 2. New Interface.

References

[1] J. Artiles, J. Gonzalo, and F. Verdejo. A testbed for people searching strategies in the WWW. In *SIGIR*, 2005.

[2] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *WWW*, 2005.

[3] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting relationships for object consolidation. In *ACM IQIS*, 2005.

[4] Guha and Garg. Disambiguating people in search. WWW'04.

[5] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (TODS)*, 31(2), June 2006.

[6] D. V. Kalashnikov, S. Mehrotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *SIAM Data Mining (SDM)*, Newport Beach, CA, April 21–23 2005.

[3] The improvement is 9.8% if we remove webpages unlabeled in [2] (labeled "other" there). F-measure is computed as in [2].