Exploiting Web querying for Web People Search in WePS2

Rabia Nuray-Turan Zhaoqi Chen Dmitri V. Kalashnikov Sharad Mehrotra

Department of Computer Science
University of California, Irvine
Irvine, CA 92697, USA

ABSTRACT

Searching for people on the Web is one of the most common query types to the web search engines today. However, when a person name is queried, the returned result often contains webpages related to several distinct namesakes who have the queried name. The task of disambiguating and finding the webpages related to the specific person of interest is left to the user. Many Web People Search (WePS) approaches have been developed recently that attempt to automate this disambiguation process. Nevertheless, the disambiguation quality of these techniques leaves a major room for improvement. In this paper we describe our experience of applying our WePS approaches developed in [20] in the context of WePS-2 Clustering Task [14]. The approach is based on extracting named entities from the web pages and then querying the web to collecting co-occurrence statistics, which are used as additional similarity measures.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]:Information Search and Retrieval; H.3.5[**Information Storage and Retrieval**]:Online Information Services - *Web-Based Services*

General Terms: Algorithms, Experimentation, Measurement
Keywords: Clustering, Skyline-based classifier, Web People Search, WePS, Named Entity Web Co-Occurrences, Social Network Analysis, Web Querying

1. INTRODUCTION

The rapid growth of the Internet has made the Web a popular place for collecting information. Internet users access billions of web pages online using search engines. Searching for a specific person is one of the most popular search queries. Some statistics suggest that such searches account for as much as 5-10% of all search queries [13].

This research was supported by NSF Awards 0331707 and 0331690, and DHS Award EMW-2007-FP-02535.

Search engines return a collection of webpages, when they are queried with a person name $\mathcal{N}$. Let $D = \{d_1, d_2, \ldots, d_k\}$ be the set of the top k returned results of a search engine. The webpages in D are related to the set of one or more namesakes who have the queried name, where these namesakes are unknown beforehand. The goal of a Web People Search (WePS) system is to automatically cluster the webpages in D such that each cluster corresponds to a namesake. One of the key challenges that needs to be overcome to make the WePS functionality a reality, is to build a WePS system that is capable of reaching high disambiguation quality.

In this paper we discuss our experience of applying our WePS system developed in [20] to WePS-2 task [14]. The main contribution of our approach [20] was to show that the Web can be used as an external data source to improve the quality of Web People Search. The proposed solution employs two-phase clustering. First, initial clusters are formed by merging using TF/IDF cosine similarity with a conservative threshold on extracted Named Entities (NEs). This step results in many clusters which are pure but incomplete, that is, they do not contain all of the webpages of the corresponding namesake. Second, for the pairs of webpages d_i and d_j that are not merged during the first phase, the algorithm forms additional queries to a web search engine to collect additional similarity evidence in the form of co-occurrence statistics. The queries consist of various combinations of NEs from web pages d_i and d_j and of the queried name $\mathcal{N}$. The returned counts are then transformed to form similarity features for the d_i, d_j pair. We have developed a specialized skyline based classifier that takes these similarity features and decide whether or not d_i and d_j should be merged to form the final clustering.

To the best of our knowledge our solution [20] was the first to show that a web search engine could be queried as an external data source in the context of WePS. While the proposed solution was sufficient to demonstrate the above idea, we were aware of its limitations, including:

1. *Features.* The first and most crucial limitation was that the approach ignores many other features present in the data, including hyperlinks, emails, phone numbers, specifics of NEs, and so on. We have shown that these features are very useful in our another WePS approach [15, 18] that we had developed before [20].
2. *Clustering Algorithm.* We have used a simple agglomerative clustering algorithm. It simply merges any two webpages if they are considered sufficiently similar. This, for example, could cause two large clusters, that correspond to distinct namesakes, to be wrongly

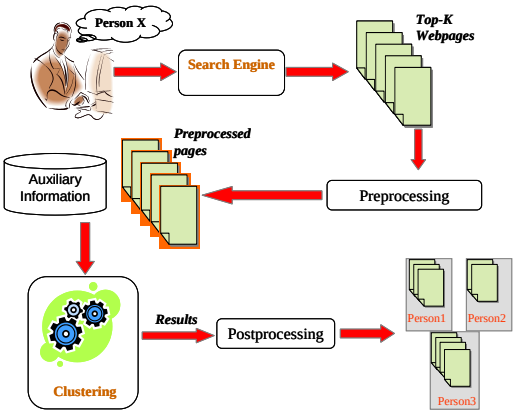

Figure 1: WePS Processing Steps.

merged when there are a few wrong "links" connecting pairs of webpages from these clusters. Again, in [15,18], we have shown that using Correlation Clustering could handle such situations and would result in better groupings than a simple agglomerative clustering.

3. *Multiple Namesakes per Webpage.* A webpage could refer to multiple namesakes whereas the solution in [20] assumed only one namesake per page is possible.
4. *Implementation Issues.* Due to restrictions of search engine APIs on the number of queries allowed per day per IP, the implementation of our algorithm was cumbersome. It required significant effort and time to collect the necessary co-occurrence statistics, which presented a challenge of applying this approach to WePS-2 data.

When submitting the results to WePS-2 we have attempted to address the last two issues. To deal with the third issue we have developed a simple segmentation code that splits a webpage into multiple sub-webpages (one per namesake) if it detects that the webpage describes multiple namesakes. To address the fourth issues we have modified the code to query Yahoo! BOSS API. But in doing so we have made an error in the code, so the algorithm has not been applied correctly, resulting in the drop of the achieved quality. The nature of the mistake is discussed in Section 3.

The rest of this paper is organized as follows. Section 2 provides an overview of the approach. The results of applying the approach to WePS-2 data is analyzed in Section 3. Section 4 summarizes the related research work. Finally, Section 5 concludes the paper.

2. APPROACH OVERVIEW

In this section we first cover the basic steps of the approach in Section 2.1. We then describe the most important clustering step in more detail in Section 2.2.

2.1 Steps of the Approach

The steps of the overall WePS approach, in the context of a *middleware* architecture, are illustrated in Figure 1. They include:

1. *User Input.* The user issues a query via the input interface.
2. *Top-K Retrieval.* The system (middleware) sends a query consisting of a person name to a search engine, such as Yahoo!, and retrieves the top-K returned web pages. This is a standard step performed by most of the current WePS systems.
3. *Pre-processing.* These top-K webpages are then preprocessed. The main pre-processing steps are:
   (a) *Extraction.* Named Entities, specifically people, locations, organizations are extracted using a third party named entity extraction software (i.e., Stanford's Named Entity Recognizer[1]). Some auxiliary data structures are built on this data.
   (b) *Segmentation.* If algorithm detects that a webpage might refer to multiple namesakes, this webpage is segmented into sub-webpages (one per namesake).
   (c) *NE Filtering.* Certain NEs are too ambiguous to be part of queries. We use four filters to detect such NEs to exclude them from queries.
4. *Clustering.* The top-K webpages are then clustered. The corresponding algorithms will be explained in more detail in Section 2.2. The clustering part consists of the following stages:
   (a) *TF/IDF Similarity.* TF/IDF similarity on NEs only is computed.
   (b) *Querying the Web.* For each pair of webpages d_i and d_j several co-occurrence queries are formed and issued to a Web search engine.
   (c) *Feature Generation.* The returned co-occurrence counts are then transformed into features.
   (d) *Making Pairwise Decisions.* Using the above features a Skyline-based classifier decides for each pair of webpages d_i and d_j whether they co-refer (positive decision) or not (negative decision).
   (e) *Applying Clustering.* A clustering algorithm, such as single-link clustering or correlation clustering, takes into account the TF/IDF similarity, generated features, and the decisions made by the Skyline-based classifier to output the resulting clustering.
5. *Post-processing.* The post-processing steps include:
   (a) *Cluster Sketches* are computed.
   (b) *Cluster Rank* is computed based on (a) the context keywords, if present and (b) the original search engine's ordering of the webpages.
   (c) *Webpage Rank* is computed to determine the relative ordering of webpages inside each cluster.
6. *Visualization.* The resulting clusters are presented to the user, which can be interactively explored.

Segmentation. We have developed a segmentation algorithm that preprocesses the webpages in order to decide if they contain more than one namesake. The heuristic we

[1]http://nlp.stanford.edu/software/CRF-NER.shtml

used reads each web page and analyzes whether it contains the queried name $\mathcal{N}$ with different middle initials. If this is so, the algorithm decides that the webpage contains multiple namesakes and segments it. For instance, if the queried name $\mathcal{N}$ is `John Smith` and the webpage contains `John Adam Smith` and `John B. Smith` then the heuristic detects the two different middle initials: A and B respectively. To segment such a webpage, the algorithm looks at the html tags that are before the next full name but after the current full name. If the tags are one of `<p>`,`<br>`,`<ul>`, `<li>`, `<ol>`, and `<tr>`, then it segments the document at that point.

Filtering. The named entities employed in web queries are cleaned using a set of four filters, see [20] for details. These NEs serve as context to identify the namesake and the idea of using filters is to make the context as precise as possible. When we looked at the extracted NEs we found that most of the NEs are too ambiguous to be used as a context. Such NEs are filtered out from further consideration. The goal is to minimize the number of queries to the web search engine, therefore the filters are specifically designed not to use any extra queries. The first filter removes any NE that can be a location name, because they are too ambiguous for namesake disambiguation. The second filter deals with the NEs that consist of one-word person names, such as "John". Such NEs are highly ambiguous since they can appear in the context of many namesakes on the Web. Similarly, the third filter handles NEs that are common English words. The last filter, removes the common named persons whose last name is the same as the last name specified in the original query.

2.2 Clustering

The most important part of the WePS system is clustering. It consists of the following steps.

TF/IDF Similarity. The algorithm computes TF/IDF similarity of the top-k webpages. The similarity is computed only on the named entities extracted from these webpages, as general overall text of the webpages context was found to be too ambiguous, frequently leading to the wrong merge decisions.

Querying the Web. Unlike many other WePS solutions, including our own [15, 18], the solution we have used for WePS2 does not limit its analysis to the information stored in the *Top-k* returned web pages. Rather it employs the Web as an external data source to get additional information that could be used to compute similarity of the webpages.

The purpose of using Web queries is to evaluate the degree of interaction of the social networks for two namesakes represented by webpages d_i and d_j. If there is evidence on the web that two social networks are closely related, then two webpages are merged into one cluster. The guiding principles in formulating the queries are:

- *Quality.* The queries should be chosen such that their results should allow creating a set of features that would enable high quality disambiguation.
- *Efficiency.* The overall number of the required queries should be minimized for efficiency reasons.

Two major types of Web queries are utilized:

1. N `AND` $\mathcal{C}_i$ `AND` $\mathcal{C}_j$
2. $\mathcal{C}_i$ `AND` $\mathcal{C}_j$.

Here, $\mathcal{C}_i$ represents the context for d_i. It can be either the set of people NEs $\mathcal{P}_i$, or organization NEs $\mathcal{O}_i$. Context $\mathcal{C}_j$ is defined similarly for document d_j. Since $\mathcal{C}_i$ and $\mathcal{C}_j$ can have two possible assignments each, this creates 4 context combinations. Given that there are 2 types of queries, this leads to 8 queries in total.

For example, assume that the user searches for the webpages related to "William Cohen". Suppose that the algorithm extracts 2 namesakes of each type per webpage, that is, $m = 2$. Assume that webpage d_i contains names "Jamie Callan" and "Tom Mitchell", and webpage d_j contains names "Andrew McCallum" and "Andrew Ng". Then the first query will be:

> "William Cohen" `AND` ("Jamie Callan" `OR` "Tom Mitchell") `AND` ("Andrew McCallum" `OR` "Andrew Ng").

The web search engine API has a function call that computes the number of webpages relevant to the query, without actually retrieving those webpages.

Observe that dataset D alone might not have any evidence to merge d_i and d_j. For instance, the TF/IDF similarity between d_i and d_j might be low. Also, among the webpages in D, names "Jamie Callan" and "Tom Mitchell" might be only mentioned in d_i, whereas "Andrew McCallum" and "Andrew Ng" only in d_j, and otherwise D might not contain any information revealing interactions among these people. However, querying the Web allows the algorithm to gain *additional information* to support the merge. In this case, the counts will be high enough to indicate that the people mentioned in the query are closely related.

Feature Creation. To estimate the degree of overlap of two contexts $\mathcal{C}_i$ and $\mathcal{C}_j$ for webpages d_i and d_j for the queried name N we can compute the co-occurrence count $|N \cdot \mathcal{C}_i \cdot \mathcal{C}_j|$ for query $N \cdot \mathcal{C}_i \cdot \mathcal{C}_j$. However, it might be difficult to interpret this absolute value without comparing it to certain other values. For instance, if this count value is high, does it mean the contexts overlap significantly and thus d_i and d_j should be merged? Or, is it simply because N is a common name and thus there are lots of webpages that contains it under many contexts? Or, is it because contexts $\mathcal{C}_i$ and $\mathcal{C}_j$ are too unspecific, and thus too many webpages contain them?

The proposed algorithm uses the Dice similarity to get the normalized version of $|N \cdot \mathcal{C}_i \cdot \mathcal{C}_j|$ count:

$$Dice_2(N, \mathcal{C}_i \cdot \mathcal{C}_j) = \frac{2|N \cdot \mathcal{C}_i \cdot \mathcal{C}_j|}{|N| + |\mathcal{C}_i \cdot \mathcal{C}_j|}.$$

In [20] we explain the various advantages of using this specific formula.

Making Pairwise Decisions. The web features are then used to make merge or do-not-merge decisions for each pair of pages d_i and d_j. The observation is that the larger the value of the feature the more evidence is there to merge. Thus, if feature f_1 dominates feature f_2 and f_2 has been decided to be classified as a positive merge decision, then logically f_1 should also be classified as a positive merge decision, since it has more evidence to merge. Thus, any classifier that takes this rule into account will essentially learn a classification Skyline, where everything is on and above the skyline should be classified as a merge.

One of the key contributions of [20] is a new Skyline-based classifier for deciding which d_i and d_j webpages should be

merged based on the corresponding feature vector. It is a specialized classifier that we have designed specifically for the clustering problem at hand. It learns the Skyline directly from data using supervised learning. Skyline-based classifier gains its advantage due to a variety of functionalities built into it, including:

- It takes into account dominance that is present in the feature space.
- It also fine tunes itself to the quality measure being used.
- It takes into account transitivity of merges: that is, accounts for the fact that two large clusters can be merged by a single merge decision, and, thus, one direct merge decision can lead to multiple indirect ones.

These properties allow it to easily outperform other classification methods (which are generic), such as DTC or SVM. The approach is discussed in detail in [20].

Applying Clustering. A clustering algorithm takes the information collected above and produces the resulting clustering. The clustering algorithm we used for the WePS2 submission is a single-link hierarchical agglomerative clustering. It first merges all of the webpages having very high TF/IDF similarity. It then merges all of the pairs webpages which the skyline based classifier classifies as a "merge". In this paper we also tested correlation clustering which we used for WePS in [15].

3. EXPERIMENTAL RESULTS

In this section we report and analyze the quality of our algorithm on the WePS-2 dataset.

Training Skylines. We trained the classification skylines on the WePS-1 dataset [2] (including trial dataset) plus the WWW'05 dataset [4]. Our skyline-based classification algorithm can adapts itself to the given quality measure. For the WEPS2 submission we selected to optimize for the harmonic mean of purity and inverse purity measures, that is, F_P. The threshold for the first phase TF/IDF clustering was chosen as 0.35 based on the results from [20]. We collected co-occurrence statistics using the Yahoo! API.

Runs. The two main runs we have submitted we called UCI-1 and UCI-2. The only difference between the two runs is that UCI-2 uses the segmentation code and UCI-1 does not. Our official WEPS2 results are presented in Table 1.

Mistake in the Code. The official results indicated low purity of our algorithm. Given that typically our algorithm achieves higher purity over inverse purity we have reexamined the code and found an error in it. The error has arisen as a result of addressing the implementation issue with the Yahoo! BOSS API, mentioned in the introduction. The results in [20] have not been affected by the error.[2]

[2]The error was in the issued queries. The "N" query was missing, that is, the one that counts how many pages are returned for the queried name in total. The queries were saved in a 9-element array. N query was the first element in the array. Since it was missing everything shifted in the array. For example, instead of computing $NP_1N_2/(N + P_1P_2)$, we computed $P_1P_2/(NP_1P_2 + NP_1O_2)$. As the result, the feature set was supposed to be: $|NP_1P_2|$, $|NP_1O_2|$, $|NO_1P_2|$,$|NO_1O_2|$, $\frac{|NP_1P_2|}{|N|+|P_1P_2|}$, $\frac{|NP_1O_2|}{|N|+|P_1O_2|}$, $\frac{|NO_1P_2|}{|N|+|O_1P_2|}$,$\frac{|NO_1O_2|}{|N|+|O_1O_2|}$ however, in the official runs, we used the following: $|P_1P_2|$, $|P_1O_2|$, $|O_1P_2|$,$|O_1O_2|$, $\frac{|P_1P_2|}{|NP_1P_2|+|NP_1O_2|}$, $\frac{|P_1O_2|}{|NP_1P_2|+|NO_1P_2|}$, $\frac{|O_1P_2|}{|NP_1P_2|+|NO_1O_2|}$,$\frac{|O_1O_2|}{|NP_1P_2|}$.

Topic	BEP	BER	F_B	P	IP	F_P
UCI-2 (segmentation)	0.66	0.84	0.71	0.75	0.89	0.80
UCI-1 (no segment.)	0.65	0.84	0.70	0.74	0.89	0.79

Table 1: Official Results with a Mistake in our Code.

Topic	BEP	BER	F_B	P	IP	F_P
Correlation Clust.	**0.86**	**0.76**	**0.80**	**0.90**	**0.84**	**0.87**
Original Algo.	0.82	0.76	0.78	0.87	0.84	0.85
Submission	0.66	0.84	0.71	0.75	0.89	0.80

Table 2: Results

Results Without the Error. Had we not introduced the error in the code, the original algorithm from [20] would have produced the results shown in Table 2 for UCI-2 run. The original algorithm, as expected, achieves higher purity compared to the inverse purity, reaching the overall F_B of 78% which is a 7% increase from our official results of 71%.

Different Clustering. As we have mentioned, a simple agglomerative clustering could merge large clusters of different entities due to a few wrong links. To address this issue we have used correlation clustering in our other WePS solution [15] with the skylines. If we use correlation clustering for WEPS-2 dataset, than our solution reaches F_B of 80%.

To summarize, the results of the approach using correlation clustering is B-cubed Precision of 0.86 and Recall of 0.76. Given that the purity is relatively high, this suggests that if we want to improve the results further, the algorithm should collect more evidence for merges. That is, we have to address the first limitation mentioned in the introduction - analyze the plethora of other features present in the data, including hyperlinks, emails, phone numbers, and so on.

4. RELATED WORK

The Web People Search challenge is closely related to the well-studied Entity Resolution problem. In our previous work we also have developed interrelated techniques to solve various Entity Resolution challenges, e.g. [8, 9, 15–19, 24]. The approach covered in the paper, however, is *not* related to those techniques. The key algorithm for training the skyline-based classifier is new. In fact, we are unaware of any Entity Resolution technique that would use a similar approach under any context.

There are several research efforts that address specifically Web Person Search and related challenges [1–5,7,23,25]. The approach of [4] is based on exploiting the link structure of pages on the Web, with the hypotheses that Web pages belonging to the same real person are more likely to be linked together. Three algorithms are presented for disambiguation, the first is just exploiting the link structure of Web pages and forming clusters based on link analysis, the second algorithm is based on word similarities between documents and does clustering using Agglomerative/Conglomerative Double Clustering (A/DC), the third approach combines link analysis with A/DC clustering. The work in [22] clusters documents based on the entity (person, organization, and location) names and can be applied to the disambiguation of semi-structured documents, such as Web pages. The primary new contribution is the development of a document

generation model that explains, for a given document how entities of various types (other person names, locations, and organizations) are "sprinkled" onto the document.

In Sem-Eval 2007, a workshop on WePS task was held [2]. Sixteen different teams from different universities have participated to the task. The participated systems utilized named entities, tokens, URLs, etc that exist in the documents. It has been shown that as the extracted information increases the quality of the clustering increases. Use of different NE recognition tools affects the results of clustering as well. The NE based single link clustering was the one of the top three systems [11], because the named entity network alone enables to identify most of the individuals.

There are also a few publicly available Web search engines that offer related functionality, in that Web search results are returned in clusters. Clusty (http://www.clusty.com) from Vivisimo Inc. and Kartoo (http://www.kartoo.com) are search engines that return clustered results. However the clusters are determined based on intersection of broad topics (for instance research related pages could form one cluster and family pages could form another cluster) or page source, also the clustering does not take into account the fact that multiple persons can have the same name. For all of these engines, clustering is done based on entire Web page content or based on the title and abstract from a standard search-engine result. ZoomInfo (http://www.zoominfo.com/) and Spock (http://www.spock.com/) are commercially available people search engines.

Recently, researchers have started to use external databases, such as ontology and Web Search engine in order to improve the classification and clustering qualities in different domains [6, 10, 12, 21]. For example, querying the Web and utilizing the search results are used for Word Sense Disambiguation (WSD) [6] and record linkage in publications domain [10, 21]. The number of queries to a search engine is a bottleneck in these approaches. Hence, the study in Kanani and McCallum [21] tries to solve the problem in the case of limited resources. The suggested algorithm increased the accuracy of data cleaning while keeping the number of queries to a search engine minimal. Similarly the approach in [10] uses the web as a knowledge source for data cleaning. The study proposed a way to formulate queries and used some standard measures like TF/IDF similarity to compute the similarity of two different references to an entity. The work in [11] is a complementary work to the one in [21]. In [6] the authors proposed to use the co-occurrence counts for disambiguation of different meanings of words. The authors used web-based similarity measures like WebJaccard, WebDice, and so on. These measures are also utilized as features for the SVM based trainer along with a set of token based features, where the trainer learns the probability of two terms being the same.

In this paper, we study a similar approach, where our aim is to increase the quality of the Web Person search. Our study differs from the others in the way we utilize the web search results. In addition, we have used a different way to formulate queries.

5. CONCLUSIONS AND FUTURE WORK

In this paper we describe our experience of applying our skyline based classification approach proposed in [20] on WEPS-2 dataset. The approach employs the Web as an external data source to collect additional similarity information to better the quality of WePS. As future work we plan to develop a new solution that would utilize the plethora of other features available in the data, including hyperlinks, emails, phone number, and so on.

6. REFERENCES

[1] R. Al-Kamha and D. W. Embley. Grouping search-engine returned citations for person-name queries. In *WIDM04*, 2004.

[2] J. Artiles, J. Gonzalo, and S. Sekine. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *SemEval*, 2007.

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

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

[5] D. Bollegala, Y. Matsuo, and M. Ishizuka. Extracting key phrases to disambiguate personal names on the web. In *CICLing*, 2006.

[6] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *WWW'07*, 2007.

[7] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Comput. Netw. ISDN Syst.*, 29(8-13):1157–1166, 1997.

[8] S. Chen, D. V. Kalashnikov, and S. Mehrotra. Adaptive graphical approach to entity resolution. In *Proc. of ACM IEEE Joint Conference on Digital Libraries (ACM IEEE JCDL 2007)*, Vancouver, British Columbia, Canada, June 17–23 2007.

[9] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting relationships for object consolidation. In *Proc. of International ACM SIGMOD Workshop on Information Quality in Information Systems (ACM IQIS 2005)*, Baltimore, MD, USA, June 17 2005.

[10] E. Elmacioglu, M.-Y. Kan, D. Lee, and Y. Zhang. Web based linkage. In *WIDM07*, 2007.

[11] E. Elmacioglu, Y. F. Tan, S. Yan, M.-Y. Kan, and D. Lee. Psnus: Web people name disambiguation by simple clustering with rich features. In *SemEval*, 2007.

[12] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[13] R. Guha and A. Garg. Disambiguating people in search. In *Stanford University*, 2004.

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

[15] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray. Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, 20(11), Nov. 2008.

[16] D. V. Kalashnikov, Z. Chen, R. Nuray-Turan, S. Mehrotra, and Z. Zhang. WEST: Modern technologies for Web People Search. In *Proc. of the 25th IEEE IntŠl Conference on Data Engineering*

(IEEE ICDE 2009), Shanghai, China, March 29 - April 4 2009. demo publication.

[17] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (ACM TODS)*, 31(2):716–767, June 2006.

[18] D. V. Kalashnikov, S. Mehrotra, S. Chen, R. Nuray, and N. Ashish. Disambiguation algorithm for people search on the web. In *Proc. of the IEEE 23rd International Conference on Data Engineering (IEEE ICDE 2007)*, Istanbul, Turkey, April 16–20 2007. short publication.

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

[20] D. V. Kalashnikov, R. Nuray-Turan, and S. Mehrotra. Towards breaking the quality curse. A web-querying approach to Web People Search. In *Proc. of Annual International ACM SIGIR Conference*, Singapore, July 20–24 2008.

[21] P. Kanani, A. McCallum, and C. Pal. Improving author coreference by resource-bounded information gathering from the web. In *IJCAI*, 2007.

[22] X. Li, P. Morie, and D. Roth. Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*, pages 45–68, 2005.

[23] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka. Polyphonet: an advanced social network extraction system from the web. In *WWW*, 2006.

[24] R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra. Self-tuning in graph-based reference disambiguation. In *Proc. of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007), Spinger LNCS*, Bangkok, Thailand, April 9–12 2007.

[25] X. Wan, J. Gao, M. Li, and B. Ding. Person resolution in person search results: Webhawk. In *CIKM*, 2005.