Towards better entity resolution techniques for Web document collections

Surender Reddy Yerva #1, Zoltán Miklós #2, Karl Aberer #3

EPFL LSIR
Lausanne, Switzerland
{[1] surenderreddy.yerva, [2] zoltan.miklos, [3] karl.aberer}@epfl.ch

Abstract— **As person names are non-unique, the same name on different Web pages might or might not refer to the same real-world person. This entity identification problem is one of the most challenging issues in realizing the Semantic Web or entity-oriented search. We address this disambiguation problem, which is very similar to the entity resolution problem studied in relational databases, however there are also several differences. Most importantly Web pages often only contain partial or incomplete information about the persons, moreover the available information is very heterogeneous, thus we are only able to obtain some uncertain evidence about whether two names refer to the same person using similarity functions. These similarity functions capture some aspects of the similarities between Web-pages, where the names occur, thus they perform very differently for the different names. We analyze some data engineering techniques to cope with the limited accuracy of the similarity functions and to combine multiple functions. Even with our simple techniques we could demonstrate systematic performance improvements and produce comparable results to state-of-the-art methods.**

I. Introduction

Entity resolution is a well studied problem in the context of relational databases [1], [2], [3], [4], [5], [6], [7], for a survey see [8]. Even if the papers are dated back quite early, this topic has also regained in importance recently. This is most likely because it is more and more common and easy to combine independent data sources and in this scenario identifying duplicate records is essential. One faces a very similar problem on the Web: for example, person names are not unique, but it is often important to identify which person name corresponds to which real world person. Such situations include web people search or aggregating information from Web-extracted data

Even if this problem looks very similar to the entity resolution problem in databases, there are important differences. In a database typically one has to identify duplicate records, which is very different from person names. Indeed, one can verify that for example the model for fuzzy duplicates [3] does not hold in our setting. The information what could help here is the content of the Webpage, where the name appears. They are on the one hand a rich source of information, but on the other hand this source is often not so straightforward to exploit, because it is very hard to distinguish the relevant information from noise and the relevant information might be even missing.

Many entity resolution techniques rely on pairwise similarity functions, which report the similarity based on some features of the pages. It is unlikely that one can design a single similarity function, which could be used for all pages in any larger collection to decide whether they are about the same person or not, because of the heterogeneity of the pages. Typically, the functions work better in some cases and worse in others.

In this paper we discuss some data engineering techniques, which help to improve the decision we can make about whether two entities shall be considered as the same. Our strategy is to define regions (i.e. intervals) in the value space (that is $[0, 1]$) and estimate the accuracy of the functions in each region. In other words, we partition the interval $[0, 1]$ into disjoint sub-intervals and with simple machine learning techniques we estimate how well does the similarity functions predict the equivalence in each sub-interval, based on a small training set. Then, we use both the similarity values and their accuracy estimations to decide whether we should consider two entities equivalent. We also study how to combine these decisions if we have multiple similarity values and multiple accuracy estimations.

The main contributions of the paper are that even with our simple techniques we could achieve results comparable to state-of-the-art methods and with our accuracy estimations we could demonstrate systematic performance improvements. We believe that such explicit analysis of accuracy of similarity functions can be used in combination with other entity resolution techniques, to improve their performance.

The rest of the paper is organized as follows. Section II gives a precise problem definition, Section III discusses similarity functions. Section IV gives details on how do we combine multiple evidences for entity resolution, while Section V presents experimental results, Section VI contains related work and finally Section VII concludes the paper.

II. Problem Description

We consider the following variant of the entity resolution problem. We are given a collection of unstructured documents D. Each document d in D contains a set of names, $names(d)$. The names correspond to real persons, but the set of real persons P is not known, even the size of P is unknown. There are multiple documents about a person with the same name in D. The person names are non-unique, therefore some of the documents might talk about different persons, if they share the names. For each name, we would like to partition the documents in the collection, such that two documents refer to the same person if and only if they are in the same partition.

We say that two entity references (names) n_i and n_j are equivalent ($n_i \equiv n_j$) if they refer to the same person. Clearly this relation is transitive. The relation of the entity references can be represented as a graph, in which for each entity reference there is a vertex in the graph, and two vertices are connected by an edge whenever the two corresponding entities are equivalent. We refer to this graph as the entity graph. The goal of the entity resolution algorithms is to reconstruct this entity graph as accurately as possible. Note that the entity graph has very specific properties: it is not a connected graph, it is a union of pairwise disjunct connected components and each component is a clique, i.e. a complete graph, because of the transitivity of the equivalence relation.

III. Similarity functions

Similarity functions associate a value from the interval $[0, 1]$ to a pair of entities. In our case, instead of comparing the entities themselves, we compare the related web-pages. As a preprocessing step we apply information extraction tools, so the input to the similarity functions is the extracted information and not the pages themselves. In other terms, we apply (dictionary-based) named entity recognition techniques.

Each similarity function compares two webpages based on a particular feature(like concepts, urls etc) using a similarity measure(like cosine similarity, number of overlaps etc)[9][10]. We use common observations in coming up with the following similarity functions. Two webpages are about a same person, if the concepts or organizations or person names etc mentioned on the pages are similar/overlap, or if the pages urls are on a same webdomain.

For extracting features from the webpages we used several information extraction tools, including “alchemy API” [11] to extract named entities, “GATE” [12], “openCalais”[13] to extract other types of entities, such as organizations and locations. We also extract wikipedia-based concepts using “semhacker” [14]. Finally for representing a webpage as document vector we use the services provided by lucene[15]. The similarity functions we consider are summarized in Table I.

TABLE I
Basic similarity function descriptions

Fn.	Feature	Similarity Measure
F1	Weighted Concept Vector	Cosine Similarity
F2	URL of the page	String Similarity
F3	Most frequent name on the page	String Similarity
F4	Concepts Vector	Number of overlapping concepts
F5	Organizations Entities on the page	Number of overlapping organizations
F6	Other Person-Names on the page	Number of overlapping persons
F7	The name closest to the search keyword	String Similarity
F8	TF-IDF (based weights) words vector	Cosine Similarity
F9	TF-IDF (based weights) words vector	Pearsons Correlation similarity
F10	TF-IDF (based weights) words vector	Extended Jaccard similarity

Note that the similarity functions are not transitive, in fact, it is very hard to define transitive functions. We use the similarity functions to identify equivalence relations among the entities. As equivalence relations are transitive, we must cope with our inability of designing transitive functions.

IV. Our entity resolution framework

In this section we present a simple entity resolution framework, which relies on pairwise similarity functions. First we explain how do we combine the accuracy estimations with the similarity values (Section IV-A), then we discuss how to combine multiple similarity values (Section IV-B). Finally we summarize the overall technique in Section IV-C.

A. Accuracy estimations

Given a single similarity function, we can consider two related persons equivalent if their similarity value is higher than an appropriately chosen threshold. Indeed, for each function we have chosen such a threshold, using the estimates from a small training sample, where we know the equivalence relations. We have chosen a threshold, which –based on the training set– maximizes the number of correct decisions.

However, lower similarity values might have many reasons, such as real dissimilarity, missing or incomplete information on the pages, the function does not capture the differences (only in special cases), the input to the functions (generated by information extractors) is uncertain or erroneous, or the inaccuracy of the function itself. A possible way to improve threshold-based decisions is to consider the accuracy of the function. One can estimate the overall accuracy (percentage of correct decisions) of a function, based on a small training set. If a function has an overall low accuracy, the reported high similarity values are not informative. However, our experiments indicated that the such overall accuracy estimations do not work very well, as the accuracy shows significantly different values for various subsets of the input. Even if a function has an overall acceptable accuracy, in some regions it might perform particularly well. We tried multiple ways to divide the input into regions and compute accuracy estimations separately for each region.

One can define such regions based on some properties of the input (i.e. pair of entities) or based on the reported function value. We discuss here our experiments, where we defined the regions based on the similarity value. We considered two methods:

1) We defined the regions as equal sized sub-intervals: $[0, 0.1), [0.1, 0.2), \ldots, [0.9, 1]$. However, the similarity values do not have a uniform distribution in the $[0, 1]$ interval, thus choosing the regions as equal size intervals is not the best option, as some intervals contain significantly more values than others.
2) We clustered the similarity values corresponding to the training set using the $k - means$ clustering technique, the output of which are k-cluster heads and each cluster head representing a region.

Based on the training set, for each region we compute an accuracy estimate. From the training sample set, each region would contain certain sample points corresponding to link existence and non-existence. Accuracy for a region is then defined as the percentage of the sample points representing link existence. If this value is lower than 0.5 then it suggests that the majority pairs should not be considered as a link.

For each region we estimated the accuracy. Figure 1 shows the accuracy values for $k - means$ generated regions, for the similarity function F3, for the person "Cohen", in the WWW'05 dataset. The accuracy values varied significantly for all functions. Even if the actual values might depend on the actual dataset, the variation of accuracy is a common phenomenon. Note that the accuracy estimations are based on the small training set and not the entire data, so computationally the method remains feasible.

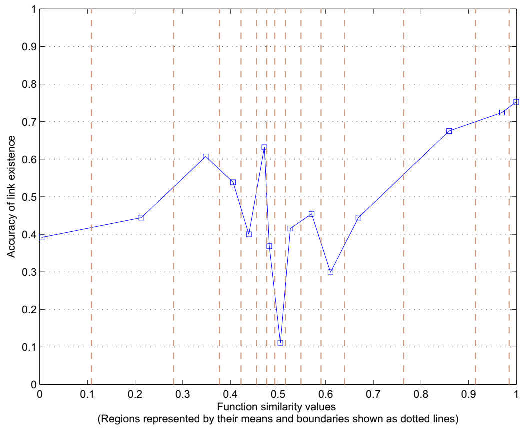

Fig. 1. Accuracy of a similarity function

B. Combining multiple functions

Given the heterogeneity of the web, we cannot expect that we can design a single similarity function which would perform optimally in all cases. Thus we need to compute several similarity functions and try to make our decision based on a combination of the similarity functions. To find an optimal way of combination involves a lot of challenges.

The different functions report similarity values with very different value distribution as they capture different aspects of similarity. Thus instead of combining the similarity values themselves, we try to combine the decisions (whether or not to consider two entities as equivalent) and the estimated accuracy values.

In this way, for each function f_i we obtain a graph G_{D_j}, together with accuracy estimations, where D_j is the decision criteria, i.e. whether we decide upon a single threshold or also consider the accuracy estimations. Our goal is to combine the the individual graphs G_{D_j} into a single graph $G_{combined}$. First we obtain a multi-graph, where the multiple edges between two nodes correspond to the edges from the individual graphs. We weight the edges with the individual accuracy estimations, which we consider as estimations of the probability of a link. Then we compute a weighted average and obtained an optimal threshold, based on our training set. If the combined value is above this threshold, we add an edge to $G_{combined}$.

We also considered other combination techniques. A very simple method is to estimate the overall accuracy of all G_{D_j} graphs, and chose the best one as a $G_{combined}$. Interestingly, this combination technique performed the best on our datasets, which might not always be the case. It is important to note that not always the same function performed the best.

C. Entity resolution algorithm

Our technique is the following. First we compute a complete weighted graph $G_w^{f_i}$ for each similarity function f_i. (The nodes of the graph $G_w^{f_i}$ correspond to the Web pages, while the weights on the edges are the similarity values reported by f_i.) To avoid computational bottlenecks, we apply a basic blocking technique, so essentially we only compute the similarity values between documents, which are about a person with the same name.[1] From the graph $G_w^{f_i}$ we would like to obtain a graph G_{D_j}, a (not-weighted) graph, where an edge between two nodes shall indicate whether the entities corresponding to the nodes are the same. This transformation depends on the decision criteria D_j. These decision criteria include to chose values above a threshold or also consider accuracy estimations, as it is explained in Section IV-A. Once we have all the graphs G_{D_j}, for all functions f_i and all decision criteria D_j, we obtain a combined graph $G_{combined}$, which is explained in Section IV-B. For this we also use accuracy estimations $acc(G^i_{D_j})$, based on the training set. Finally, we apply clustering techniques to obtain the final entity resolution. In our recent implementation we compute the transitive closure of the graph $G_{combined}$, but we also experimented with several other clustering techniques, such as correlation clustering [16]. The overall procedure is summarized in Algorithm 1.

Algorithm 1 Entity resolution

```
**compute** the graph $G_w^{f_i}$ for each $f_i$ (per block)
**obtain** the decision criteria $D_j$(threshold, regions, etc.) from the training set
**apply** the decision $D_j$ to the data, to compute $G^i_{D_j}$, for each $i$ and $D_j$
**compute** the accuracy $acc(G^i_{D_j})$
**combine** them, for all $i$, $D_j$
**apply** a clustering algorithm
**output** the final entity resolution
```

V. Experimental evaluation

A. Experimental setup

We performed our experiments on a 2GB RAM, Genuine Intel(R) T2500 @ 2.00 GHz CPU. Linux Kernel 2.6.24, 32-bit machine. We implemented our methods using matlab and java.

[1] Such blocking strategy is very natural in the datasets we used, where the documents already organized around person names. In general, one needs to consider the applicable blocking schemes more carefully.

1) Datasets: For our experiments we used two different datasets: the WWW'05 and the WePS. The WWW'05 dataset was created in [17]. This dataset was also used in a series of papers, which enabled us to compare our methods with other techniques. The dataset contains Web documents for 12 different person names. The dataset was created by querying the Web using the google search engine with the different person names. The top 100 returned web documents for the web search were gathered and labeled manually. For each person, the correct resolution is available together with the data. We used this ground truth to measure the quality of our techniques. The number of clusters for each person name is different, it varies from 2 to 61.

WePS-2 test data is provided by the web people search clustering task [18]. The test data consisted 30 Web page collections, each one corresponding to one ambiguous name. These 30 person names were chosen from three different sources: wikipedia, ACL'08 (Association for Computational Linguistics Program committee members) and US census data. Each person name was queried using yahoo search API and the top 150 results were included into the dataset. We have evaluated our techniques on WePS-2 dataset. We report the performance figures we observed on the 10 person names chosen from the ACL'08.

2) Methods: Given the dataset, we use 10% of the complete dataset as the training set. The performance of the ER algorithm depends on how well the training set represents the features of the complete dataset. In order to avoid any bias, we repeated the experiments for 5 runs and the averages of the observed results are presented. On each run we randomly choose the training subset from the complete dataset.

3) Measures of interest: Various measures are considered to assess the quality of entity resolution. Precision, recall and F-measure are widely used in information retrieval. We also measure the Rand-index [9] and the F_p-measure [10], which is the harmonic mean of purity and inverse purity.

We note here that the above measures rely on the fact, that we know the ground truth, which is unrealistic in the Web context. We could apply them for the document collections in our experiments, as we had this information available.

B. Experimental results

Figure 2 shows the performance of the individual similarity functions on the entire WWW'05 dataset. The figure shows three metrics, namely F_p-measure, F-measure and $Rand$-index. The final column, depicted as black in the figure, is the combined performance of our proposed technique, which clearly shows improved performance. Similarly, Figure 3 shows the experimental results on the WePS dataset.

Table III contains the achieved F_p values, for each individual person, by each individual function in the WWW'05 dataset. One can observe that each function performs differently for different persons. For example, for "Voss" the function F8 has the highest F_p-value, while for "Mulford" the best function is F6.

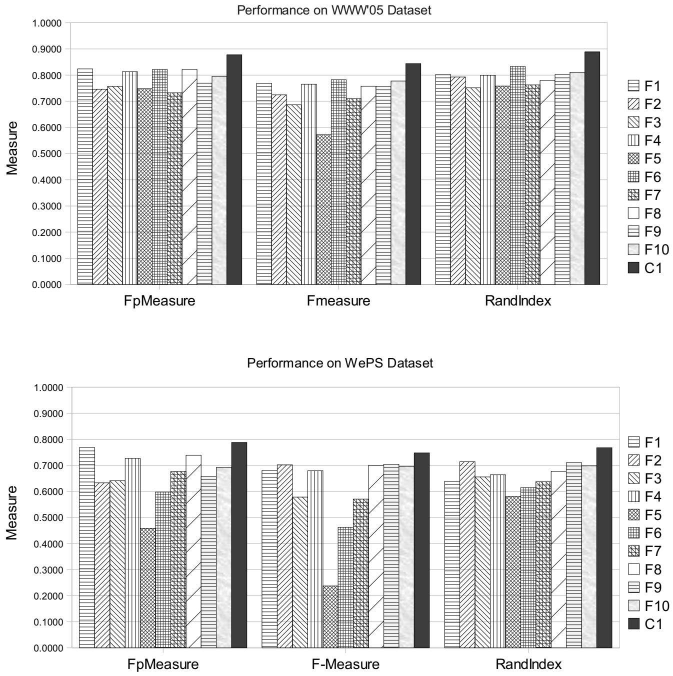

Fig. 3. WEPS results graph.

The Table II shows that by considering more and more functions we indeed get a better performance, for both datasets. The first three columns show the maximal performance considering just the threshold-based technique, by including functions $I4 = \{F4, F5, F7, F9\}$, $I7 = \{F3, F4, F5, F7, F8, F9, F10\}$, $I10 = \{F1, \ldots, F10\}$, respectively. The columns $C4$, $C7$ and $C10$ take the same functions as the first three columns respectively, but there we chose the best decision criteria, based on accuracy estimation of the regions. The column W shows the performance of weighted average combination result. The table also contains a comparison with the figures reported in the literature. The best results for the WWW'05 dataset were reported in the paper [19], however they manually improved the available ground truth, which is not available, therefore the comparison is not precise. The last column contains the result achieved by the WePS competition winner. We found this result in [20], but we could not obtain the original reference.

VI. Related work

The entity resolution and related problems, such as for example duplicate detection have an extensive literature in the database community, a few important references include [1], [2], [21], [6]. For a survey see [8]. Many papers suggest (for example [2]) incremental clustering-based methods, while others propose pairwise comparison-based techniques. A recent paper [5] presents a pairwise comparison-based method, where the authors also consider confidence values during the resolution process. They propose to merge database records, which refer to the same entity, right away, as they are found to be equivalent by the algorithm. The algorithm also computes a new combined confidence value for the merged record. A more complete analysis of results can be found in [7], where the authors also study, how to chose the sequence of the records

TABLE II
COMPARISON OF RESULTS

dataset		I4	I7	I10	C4	C7	C10	W	Related work
WWW'05	F_p-measure	0.8128	0.8211	0.8232	0.8537	0.8732	0.8774	0.8371	0.864 [20], 0.9000 [19]
	F-measure	0.7654	0.7773	0.7822	0.8338	0.8376	0.8438	0.8168	0.8000 [17], 0.8 [19]
	RandIndex	0.8018	0.8109	0.8326	0.8747	0.8814	0.8886	0.8531	
WePS	F_p-measure	0.7270	0.7388	0.7682	0.7560	0.7659	0.7880	0.7785	0.791 [20], WePS: 0.7800
	F-measure	0.7042	0.7042	0.7042	0.7127	0.7231	0.7476	0.7190	
	RandIndex	0.7102	0.7102	0.7139	0.7492	0.7531	0.7675	0.7290	

TABLE III
F_p MEASURE FOR EACH NAME IN WWW'05 DATASET

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	C10	W
Cheyer	0.9686	0.9948	1.0000	0.9686	0.7950	0.9948	1.0000	0.9948	0.9948	0.9948	1.0000	0.9948
Cohen	0.8724	0.3827	0.7368	0.8859	0.8444	0.8991	0.8839	0.8746	0.8746	0.8718	0.8991	0.8816
Hardt	0.8680	0.8828	0.8985	0.8680	0.4717	0.9074	0.8985	0.8828	0.8828	0.8779	0.9074	0.8828
Israel	0.8206	0.7568	0.7881	0.8312	0.8093	0.8476	0.7257	0.8315	0.7536	0.7568	0.8476	0.8690
Kaelbling	0.9831	0.9944	0.9711	0.9831	0.9012	0.9467	0.9711	0.9944	0.9888	0.9944	0.9944	0.9944
Mark	0.7871	0.7871	0.7228	0.7871	0.7871	0.7871	0.7668	0.7871	0.7915	0.7871	0.8104	0.7871
Mccallum	0.7921	0.7391	0.6642	0.7812	0.8066	0.8248	0.4667	0.8024	0.5851	0.8187	0.9670	0.8597
Mitchell	0.8473	0.7756	0.5796	0.7417	0.7981	0.7733	0.4448	0.5966	0.7097	0.7382	0.8575	0.6448
Mulford	0.7471	0.7467	0.7569	0.7471	0.7337	0.7582	0.7569	0.7467	0.7467	0.7467	0.8053	0.7467
Ng	0.8607	0.7111	0.7493	0.8660	0.7938	0.8163	0.7031	0.8086	0.7082	0.7082	0.8813	0.8845
Pereira	0.7215	0.5420	0.6362	0.7180	0.6389	0.6942	0.5571	0.7326	0.5554	0.5519	0.7573	0.7438
Voss	0.6094	0.6365	0.5813	0.5760	0.5993	0.6073	0.6135	0.8016	0.6391	0.6979	0.8016	0.7567

to be processed, such that the running time of the algorithm remains low.

Chauduri et al. [3] introduce a model for detecting fuzzy duplicates in databases. They extended their model also to a more general setting in [22]. Their paper is particularly important from methodological point of view, as they systematically derive their entity resolution algorithms from an axiomatic model. Unfortunately their model cannot be easily extended to the Web context and the properties of similarity functions do not show the same properties as in the case of fuzzy duplicates, so the basic assumptions of their model are not satisfied.

Entity resolution in Web context was studied by Kalashnikov et al. [23]. They propose to create an entity resolution graph, using the feature-based similarities. The graph witnesses the uncertainty of the features by having multiple nodes. The authors apply heuristic graph measures to measure the connectedness of entities. The underlying idea behind their heuristic is the "context principle": if two entities are related, then there are multiple paths in the entity resolution graph between their corresponding nodes. The authors further improved their techniques in [20]. In [20] and in many other approaches, such as for example in [6], the authors consider a more complex graph, which captures more complex relations, rather than the similarities between the entities as in our work. We limited ourselves to a simple representation and to focus the issues in this simpler case, our framework could be later extended to a more complex setting.

Combining multiple classifiers is studied in the machine learning community. The techniques can be broadly divided into two main categories: 1) Classifiers fusion, in which the final decision on a sample point is based on the fusion of decisions of individual classifiers, in some sense similar to achieving consensus. Examples include majority voting, weighted voting. 2) Dynamic Classifier selection: In this scenario, the decision of one of the classifier is chosen as the combined decision. Here, the classifier is chosen based on which classifier best represents the sample point. Woods et al. [24] discuss a method, which divides the sample space into partitions either on predefined criteria or on the features. Each classifiers performance is estimated for each partition. This estimates would be used in choosing a best classifier for each partition. Liu et al. [25] propose a novel way of combining classifiers: which is by a technique called as clustering and selection. The input sample space is partitioned into several regions and clustering the correct and incorrect decisions separately. Each classifier performance is estimated for each region. On seeing a new sample, the region to which it belongs to is identified and the classifier with best performance for that region is chosen for the final decision. In our work we used similar combination techniques.

Chen et al. [19] studied the combination of multiple classifiers, where the classifiers are applied for performing entity resolution. They also suggest that the performance of the classifiers depends on the context. Their method introduces techniques to exploit the context and find regions, where the classifier work better. Their method highly depends on their estimation of the total number of clusters (entities), which can be highly unreliable. Once they obtained the combination of the clustering methods, they also apply further techniques to improve their method, such as correlation clustering [16] and related heuristic approximation techniques.

Cudré-Mauroux et al. [26] take a different approach to entity resolution in the web context. They propose a graphical model-based probabilistic framework to capture the relations among the entities. Their framework also includes trust assessments about the providers of the entity equivalence assertions. These trust assessment values are later adjusted as their probabilistic reasoning framework eliminates the detected inconsistencies. While this approach has many advantages, it is not fully applicable to our case, as the underlying factor graph model would have very large cliques, as subgraphs, which could easily lead to poor convergence of the probabilistic reasoning.

On the Semantic Web person names might be annotated with a globally accepted ontology. This direct link between the ontology helps to disambiguate the person names. However, such globally accepted ontologies are not present in the emerging Semantic Web. Instead, ontologies are very often used as local schemas, thus one needs to relate the existing annotation to the ontology one would like to use. The Semantic Web community has developed a plethora of such techniques, see [27]. The OKKAM project suggests a different approach, [28]. They propose a service, which provides globally unique identifiers on large scale for entities, for (semantic) web applications. Their approach relies on the existence of a large and clean (i.e. resolved) collection of entity profiles. Entity profiles collect relevant attributes of real world entities. Our techniques can contribute to create or extend such an entity profile collection.

To summarize, we used a simple entity resolution framework, while there are more involved frameworks known in the literature. We applied data engineering techniques, which improved the quality of our results. These techniques (explicit analysis of the accuracy of similarity functions) can be used in combination with other techniques. Creating a large and resolved collection of entity profiles can open new perspectives for semantic web applications. Our work also contributes to this line of research.

VII. Conclusion and future work

We studied entity resolution methods for Web data collections, in particular to realize Web people search. Our techniques rely on pairwise comparisons by similarity functions. By estimating the accuracy of the similarity functions and by combining multiple functions we could demonstrate improvements in performance.

In our future work we plan to address the effect of incomplete information available in the Web pages on the accuracy of the similarity functions, by considering entropy based metrics, similar to [29]. We would like to find methods both to better combine multiple similarity functions, and to better cluster entities. Even if clustering methods are widely studied, none of the methods is fully compliant with the objectives of entity resolution in the Web context.

Acknowledgements

This work is partially supported by the by the FP7 EU Large-scale Integrating Project **OKKAM – Enabling a Web of Entities** (contract no. ICT-215032). We are grateful to the anonymous referees; their comments helped to improve the presentation of the paper.

References

[1] I. Fellegi and A. Sunter, “A theory for record linkage,” *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, December 1969.

[2] M. A. Hernández and S. J. Stolfo, “The merge/purge problem for large databases,” *ACM SIGMOD Record*, vol. 24, no. 2, pp. 127–138, May 1995.

[3] S. Chaudhuri, V. Ganti, and R. Motwani, “Robust Identification of Fuzzy Duplicates,” in *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005, pp. 865–876.

[4] Z. Chen, D. V. Kalashnikov, and S. Mehrotra, “Adaptive graphical approach to entity resolution,” in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, 2007, pp. 204–213.

[5] D. Menestrina, O. Benjelloun, and H. Garcia-Molina, “Generic Entity Resolution with Data Confidences,” in *In First International VLDB Workshop on Clean Databases*, 2006.

[6] X. Dong, A. Halevy, and J. Madhavan, “Reference reconciliation in complex information spaces,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 85–96.

[7] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom, “Swoosh: a generic approach to entity resolution,” *The VLDB Journal*, vol. 18, no. 1, pp. 255–276, January 2009.

[8] P. G. Ipeirotis, V. S. Verykios, and A. K. Elmagarmid, “Duplicate record detection: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, January 2007.

[9] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[10] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen, “Enhancing text clustering by leveraging Wikipedia semantics,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 179–186.

[11] “alchemiAPI,” http://www.alchemyapi.com/.

[12] H. Cunningham, “GATE, a General Architecture for Text Engineering,” *Computers and the Humanities*, vol. 36, pp. 223–254, 2002.

[13] “OpenCalais,” http://www.opencalais.com/documentation/calais-web-service-api.

[14] “Semantic Hacker,” http://www.semantichacker.com/.

[15] “lucene,” http://lucene.apache.org/.

[16] N. Bansal, A. Blum, and S. Chawla, “Correlation Clustering,” *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[17] R. Bekkerman and A. McCallum, “Disambiguating web appearances of people in a social network,” in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 463–470.

[18] “Second Web People Search Evaluation Workshop, WePS-2-dataset,” http://nlp.uned.es/weps/weps-2-data/, 2009.

[19] Z. Chen, D. V. Kalashnikov, and S. Mehrotra, “Exploiting context analysis for combining multiple entity resolution systems,” in *Proceedings of the 35th SIGMOD international conference on Management of data*, 2009, pp. 207–218.

[20] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray-Turan, “Web People Search via Connection Analysis,” *IEEE Transactions on Knowledge and Data Engineering* , vol. 20, no. 11, pp. 1550–1565, November 2008.

[21] V. S. Verykios, G. V. Moustakides, and M. G. Elfeky, “A bayesian decision model for cost optimal record matching,” *The VLDB Journal*, vol. 12, no. 1, pp. 28–40, 2003.

[22] S. Chaudhuri, A. D. Sarma, V. Ganti, and R. Kaushik, “Leveraging aggregate constraints for deduplication,” in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 437–448.

[23] D. V. Kalashnikov and S. Mehrotra, “Domain-independent data cleaning via analysis of entity-relationship graph,” *ACM Transactions on Database Systems*, vol. 31, no. 2, 2006.

[24] K. Woods, W. P. Kegelmeyer, Jr., and K. Bowyer, “Combination of multiple classifiers using local accuracy estimates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, 1997.

[25] R. Liu and B. Yuan, “Multiple classifiers combination by clustering and selection,” *Information Fusion*, vol. 2, no. 3, pp. 163–168, 2001.

[26] P. Cudré-Mauroux, P. Haghani, M. Jost, K. Aberer, and H. de Meer, “idMesh: Graph-Based Disambiguation of Linked Data,” in *Proceedings of the 18th International World Wide Web Conference (WWW'09)*, 2009.

[27] J. Euzenat and P. Shvaiko, *Ontology matching*. Springer, 2007.

[28] P. Bouquet, T. Palpanas, H. Stoermer, and M. Vignolo, “A conceptual model for a web-scale entity name system,” in *The Semantic Web, Fourth Asian Conference, ASWC 2009*, ser. LNCS, no. 5926. Springer, 2009, pp. 46–60.

[29] P. Cudré-Mauroux, A. Budura, M. Hauswirth, and K. Aberer, “PicShark: mitigating metadata scarcity through large-scale P2P collaboration,” *The VLDB Journal The International Journal on Very Large Data Bases*, vol. 17, no. 6, pp. 1371–1384, November 2008.