Efficient Strategies for Improving Partitioning-Based Author Coreference by Incorporating Web Pages as Graph Nodes

Pallika Kanani and Andrew McCallum
Department of Computer Science, UMass, Amherst
140 Governors Drive, Amherst, MA 01002
{pallika, mccallum}@cs.umass.edu

Abstract

Entity resolution in the domain of research paper authors is an important, but difficult problem. It suffers from insufficient contextual information, hence adding information from the web can significantly improve performance. We formulate the author coreference problem as one of graph partitioning with discriminatively-trained edge weights. Building on our previous work, this paper presents improved and more comprehensive results for the method in which we incorporate web documents as additional nodes in the graph. We also propose efficient strategies to select a subset of *nodes* to add to the graph and to select a subset of *queries* to gather additional nodes, without significant loss of performance gain. We extend the classic Set-cover problem to develop a node selection criteria, hence opening up interesting theoretical possibilities. Finally, we propose a hybrid approach, that achieves 74.3% of the total improvement gain using only 18.3% of all additional mentions.

Introduction

The problem of entity resolution deals with correctly assigning the given entity names, or "mentions," to the true underlying entity that they refer to. Often, the information available in the contexts of these mentions is insufficient and hence it is difficult to make these coreference decisions without using external information. Web is a rich source of information and can be leveraged to improve performance on a coreference task. We examine the problem of author coreference in the domain of research paper publications.

Given a set of research paper citations, referring to authors with the same last name and first initials, we need to assign them to the identity of the real author. Coreference in this domain is extremely difficult. Although there is rich contextual information available that is often helpful, in many situations it is not sufficient to make a confident decision about whether or not two citations refer to the same real author. Consider, for example, the following three citations all containing a "D. Miller."

- T. Dean, R.J. Firby and D.P. Miller, Hierarchical Planning with Deadlines and Resources Readings in Planning, pp. 369-388, Allen, Hendler, and Tate eds., Morgan Kaufman, 1990.
- D.P. Miller and C. Winton. Botball Kit for Teaching Engineering Computing. In Proceedings of the ASEE National Conference. Salt Lake City, UT. June 2004.
- A. Winterholler, M. Roman, T. Hunt, and D.P. Miller. Design Of A High-Mobility Low-Weight Lunar Rover. Proceedings of iSAIRAS 2005. Munich Germany. September 2005.

The titles relate to computer science, but there is not a specific topical overlap; "Miller" is a fairly common last name; publication years are far apart and there are no co-author names in common. Furthermore, in the rest of the larger citation graph, there is not a length-two path of co-author name matches indicating that some of the co-authors here may have themselves co-authored a third paper. So there is really insufficient evidence to indicate a match despite the fact that these citations do refer to the same "D. Miller."

We refer to the identity of each author as an *entity* and the document referring to that author, namely a citation, as *mention*. As in previous work (McCallum & Wellner 2003), we make Coreference merging decisions are typically made, not merely by examining separate pairs of names, but relationally, by accounting for transitive dependencies among all merging decisions. We formulate the problem of author coreference as graph partitioning, where each vertex represents a citation mention and the edge weights represent the probability that they both refer to the same real author. The weights are based on many features with parameters learned by a maximum entropy classifier. We then apply stochastic graph partitioning to this graph, such that each partition corresponds to citations referring the the same entity.

In our previous work, (Kanani, McCallum, & Pal 2007) we describe two methods for augmenting the coreference graph by incorporating additional helpful information from the web. In the first method, a web search engine query is formed by conjoining the titles from two citations. The edge weight between the citation pair is altered by adding a feature indicating whether or not any web pages were returned by the query. This leads to significant improvement in performance. In our previous work, we also briefly describe an alternative method that uses one of the returned pages (if any) to create an additional vertex in the graph. The additional transitive relations provided by the new document can

provide significant helpful information. For example, if the new mention is a home page listing all of an author's publications, it will pull together all the citations that should be coreferent.

In this paper, we focus entirely on this second approach, in which we gather documents from the web and treat them as additional mentions of authors. We present an extensive set of experiments and show that using additional web mentions improves pairwise F1 from 59% to 92%. However, this performance gain involves investment of resources. The two main costs involved are: (1) computational cost for processing additional nodes, computing corresponding edge weights and partitioning the expanded graph, and (2) Querying the external source of information and obtaining additional mentions. Hence, we need a criterion to effectively select *nodes* to add to the graph and to select *queries* to gather additional web mentions. We propose efficient strategies to address these two aspects of *Resource-bounded information gathering* (RBIG).

We address the problem of selecting *nodes* by extending the classic Set-cover problem. The idea is to "cover" all the citations using the least possible number of web pages, where "covers" is loosely defined by some heuristic. We use information from the initial partitioning of data to address the problem of selecting *queries*. Queries are selected using citation pairs within and across tentative partitions. Considering both kinds of costs, we need to achieve a balance between the number of queries executed and the number of documents obtained. Finally, we combine the two methods to propose a hybrid approach, achieving 74% of the total performance gain using only 18% of all additional mentions.

The problem of author coreference has been studied by several people (Han, Zha, & Giles 2005; Huang, Ertekin, & Giles 2006; Bhattacharya & Getoor 2006). Learning and inference under resource limitations has been studied in various forms (Grass & Zilberstein 2000; Kapoor & Greiner 2005). For example, the value of information as studied in decision theory, measures the expected benefit of queries. Using web information in large scale systems for disambiguation has been used in (Dong *et al.* 2004). In our recent work (Kanani & McCallum 2007), we formulate the problem of resource bounded information gathering in a more abstract manner.

Leveraging Web Information

Conditional Entity Resolution Models

We use conditional models for entity resolution. We are interested in obtaining an optimal set of coreference assignments for all mentions contained in our database. In our approach, we first learn maximum entropy or logistic regression models for pairwise binary coreference classifications. We then combine the information from these pairwise models using graph-partitioning-based methods so as to achieve a good, consistent global coreference decision. As in (McCallum & Wellner 2003), we define the graph as follows.

Let $G_0 =< V_0, E_0 >$ be a weighted, undirected and fully connected graph, where $V_0 = \{v_1, v_2, ..., v_n\}$ is the set of vertices representing mentions and E_0 is the set of edges where $e_i =< v_j, v_k >$ is an edge whose weight w_{ij} is given by $(\log(P(y_{ij} = 1|x_i, x_j)) - \log(P(y_{ij} = 0|x_i, x_j)))/2$

Note that the edge weights defined in this manner are in $[-\infty, +\infty]$. The edge weights in E_0 are noisy and may contain inconsistencies. Our objective is to partition the vertices in graph G_0 into an unknown number of *M* non-overlapping subsets, such that each subset represents the set of citations corresponding to the same author. We define our objective function as $\mathcal{F} = \sum_{ij} w_{ij} f(i, j)$ where $f(i, j) = 1$ when x_i and x_j are in the same partition and -1 otherwise. We use N-run stochastic sampling technique to partition this graph, as described in (Kanani, McCallum, & Pal 2007)

Web as Feature vs. Web as Mention

Recall that sometimes there is simply not enough information available in the citations to correctly disambiguate entities in citation data. We would like to augment our graphs with information obtained from the web. This additional information can be incorporated using two alternative methods: (1) changing the weight on an existing edge, (2) adding a new vertex and edges connecting it to existing vertices.

The first method may be accomplished in author coreference, for example, by querying a web search engine with cleaned and concatenated titles of the citations, and examining attributes of the returned hits. In this case, a hit indicates the presence of a document on the web that mentions both these titles and hence, some evidence that they are by the same author. This binary feature is then added to an augmented classifier that is then used to determine edge weights. Our previous work focuses on this method and shows that this leads to substantial improvement in performance.

In this paper, we turn our attention to the second method. We query the web in a similar fashion, and create a new vertex by using one of the returned web pages as a new mention. In the following sections, we discuss alternative methods of gathering web mentions. Various features $f(\cdot)$ will measure compatibility between the other "citation mentions" and the new "web mention," and with similarly estimated parameters λ, edge weights to the rest of the graph can be set.

In this case, we expand the graph G_0, by adding a new set of vertices, V_1 and the corresponding new set of edges, E_1 to create a new, fully connected graph, G'. Although we are not interested in partitioning V_1, we hypothesize that partitioning G' would improve the optimization of $\mathcal{F}$ on G_0. This can be explained as follows. Let $v_1, v_2 \epsilon V_0$, $v_3 \epsilon V_1$, and the edge $< v_1, v_2 >$ has an incorrect, but high negative edge weight. However, the edges $< v_1, v_3 >$ and $< v_2, v_3 >$ have high positive edge weights. Then, by transitivity, partitioning the graph G' will force v_1 and v_2 to be in the same subgraph and improve the optimization of $\mathcal{F}$ on G_0.

As an example, consider the references shown in Fig.1. Let us assume that based on the evidence present in the citations, we are fairly certain that the citations A, B and C are by H. Wang 1 and that the citations E and F are by H. Wang 2. Let us say we now need to determine the authorship of citation D. We now add a set of additional mentions from the web, {1, 2, .. 10}. The adjacency matrix of this expanded graph is shown in Fig. 2. The darkness of the

(A).., H. Wang, .. Background Initialization.., ICCV'05.
(B).., H. Wang, .. Tracking and Segmenting People.., ICIP'05.
(C).., H. Wang, .. Gaussian Background Modeling.., ICASSP'05.
(D)**.., H. Wang, .. Facial Expression.., ICCV'03.**
(E).., H. Wang, .. Tensor Approximation.., SIGGRAPH'05.
(F).., H. Wang, .. High Speed Machining.., ASME, (JMSE)'05.

Figure 1: Six Example References

circle represents the level of affinity between two mentions. Let us assume that the web mention 1 (e.g. the web page of H. Wang 1) is found to have strong affinity to the mentions D, E and F. Therefore, by transitivity, we can conclude that mention D belongs to the group 2. Similarly, values in the lower right region could also help disambiguate the mentions through double transitivity.

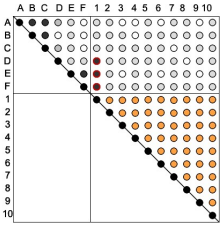

Figure 2: Extending a pairwise similarity matrix with additional web mentions. A..F are citations and 1..10 are web mentions.

Note that this matrix consists of three main regions. The upper left region corresponds to the similarity between the citations. We use citation-to-citation (c2c) classifier to fill the entries in this region. The upper right and the lower left regions together, correspond to the similarity between citations and web mentions. We use citation-to-web mention (c2w) classifier to fill the entries in this region. Finally, the lower right region corresponds to the similarity between web mentions. We use web mention-to-web mention (w2w) classifier to fill the entries in this region.

Resource-bounded Information Gathering

Gathering a large number of web mentions as described above, is clearly expensive. If we wish to build large scale systems that require the use of web information, we need an efficient strategy for information gathering. Under the constraints of resources, we wish to add only a subset of the gathered instances to the graph. This is the problem of *Selecting Nodes*. Furthermore, we also wish to query only a subset of the original instances for gathering additional mentions. This is the problem of *Selecting Queries*. In this section we describe methods that address each of these requirements.

Selecting Nodes

Incorporating additional nodes in the graph can be expensive. There can be some features between a citation and a web mention (c2w) with high computational cost. Furthermore, the running time of most graph partitioning algorithms depend on the number of nodes in the graph. Hence, instead of adding all the web mentions gathered by pairwise queries, computing the corresponding edge weights and partitioning the resulting graph, it is desirable to find a minimal subset of the web documents that would help bring most of the coreferent citations together. This is equivalent to selectively filling the entries of the upper right section of the matrix. We observe that this problem is similar to the classic Set-cover problem with some differences as noted below.

Algorithm 1 RBIG-Set-cover Algorithm

Input:
Set of citations U
Collection of web documents $C : \{S_1, S_2, ..., S_n\}$
$O \Leftarrow \emptyset$
while U is "coverable" by C **do**
  $S_k \Leftarrow arg\max_{S_i \in C} |S_i|$
  $O \Leftarrow O \cup \{S_k\}$
  $U \Leftarrow U \cap S_k$
  $C \Leftarrow \{S_i | S_i = S_i \cap S_k\}$
end while
return O
U is "coverable" by $C \equiv \exists_{(e \in U \wedge S_i \in C)}(e \in S_i)$

RBIG as Set-cover The standard Set-cover problem is defined as follows. Given a finite set U and a collection $C = \{S_1, S_2, ..., S_m\}$ of subsets of U. Find a minimum sized cover $C' \subseteq C$ such that every element of U is contained in at least one element of C'. It is known that greedy approach provides an Ω(ln n) approximation to this NP-Complete problem.

We now cast the problem of *Resource-bounded information gathering* using additional web mentions as a variant of Set-cover. The goal is to "cover" all the citations using the least possible number of web pages, where "covers" is loosely defined by some heuristic. Assuming a simplistic, "pure" model of the web (i.e. each web page "covers" citations of only one author), we can think of each web page as a set of citations and the set of citations by each author as the set of elements to be covered. We now need to choose a minimal set of web pages such that they can provide information about most of the citations in the data.

There are some differences between Set-cover and our problem that reflect the real life scenario as follows. There can be some elements in U which are not covered by any elements in C. That is, $\bigcup S_i \neq U$. Also, in order for the additional web page to be useful for improving coreference accuracy in the absence of a strong w2w classifier, it has to cover at least two elements. Keeping these conditions in mind, we modify the greedy solution to Set-cover as shown in Algorithm 1.

Selecting Queries

In many scenarios, issuing queries and obtaining the results is itself an expensive task. In our previous methods, we used all possible pairwise queries to obtain additional web documents. In this section, we will use the information available in the test data (upper left section of the matrix) to selectively issue queries, such that the results of those queries

would have most impact on the accuracy of coreference.

Inter-cluster queries The first method for reducing the number of web queries is to query only a subset of the edges between current partitions. We start by running the citation-to-citation classifier on the test data and obtain some initial partitioning. For each cluster of vertices that have been assigned the same label under a given partitioning, we define the centroid as the vertex with the largest sum of weights to other members in its cluster. We connect all the centroids with each other and get a collection of queries, which are then used for querying the web. Let n be the number of citations in the data and m be the number of currently predicted authors. Assuming that the baseline features provide some coreference information, we have reduced the number of queries to be executed from $O(n^2)$ to $O(m^2)$. A variation of this method picks multiple centroids, proportional to the size of each initial partition, where the proportion can be dictated by the amount of resources available.

Intra-cluster queries The second method for reducing the number of web queries is to query only a subset of the edges within current partitions. As before, we first start by running the citation-to-citation classifier on the test data and then obtain some initial partitioning. For each initial partition, we select two most tightly connected citations to form a query. Under the same assumptions stated above, we have now reduced the number of queries to be executed from $O(n^2)$ to $O(m)$. A variation of this method picks more than two citations in each partition, including some random picks.

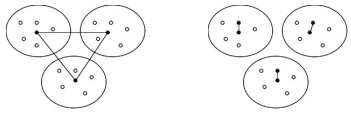

Figure 3: Inter-cluster and Intra-cluster queries

Inter-cluster vs Intra-cluster queries Both these approaches are useful in different ways. Inter-cluster queries help find evidence that two clusters should be merged, whereas intra-cluster queries help find additional information about a hypothesized entity. The efficiency of these two methods depend on the number of underlying real entities as well as the quality of initial partitioning. We are currently investigating the correlation between the performance of these query selection criteria with the characteristics of data.

Hybrid Approach

For large scale system, we can imagine combining the two approaches, i.e. *Selecting Nodes* and *Selecting Queries* to form a hybrid approach. For example, we can first select queries using, say intra-cluster queries to obtain additional mentions. This would help reduce querying cost. We can then reduce the computation cost by selecting a subset of the web mentions using the Set-cover method. We show experimentally in the next section that this can lead to a very effective strategy.

Corpus	Sets	Authors	Cits	Pairs	WebMen
DBLPTrain	11	55	497	20142	259
DBLPTest	6	42	344	17796	180
RexaTrain	4	98	526	47652	327
RexaTest	4	121	776	98183	438
PennTrain	3	41	436	32379	243
PennTest	4	52	1152	169904	601
Total	32	409	3731	386056	2048

Table 1: Summary of Data set properties.

Cost-Benefit Analysis

It should be noted that the choice of strategy for *Resource-bounded information gathering* in the case of expanded graph should be governed by a careful Cost-Benefit analysis of various parameters of the system. For example, if the cost of computing correct edge weights using fancy features on the additional mentions is high, or if we are employing a graph partitioning technique that is heavily dependent on the number of nodes in the graph, then the Set-cover method described above would be effective in reducing the cost. On the other hand, if the cost of making a query and obtaining additional nodes is high, then using inter-cluster or intra-cluster methods is more desirable. For a large scale system, a hybrid of these methods could be more suitable.

Experimental Results

Dataset Description and Infrastructure

We use the Google API for searching the web. The data sets used for these experiments are a collection of hand labeled citations from the DBLP and Rexa corpora (see table 1). The portion of DBLP data, which is labeled at Pennstate University is referred to as 'Penn.' Each dataset refers to the citations authored by people with the same last name and first initial. The hand labeling process involved carefully segregating these into subsets where each subset represents papers written by a single real author. Table 1 describes the data. The column WebMen refers to the number of web mentions obtained by the auto-labeling process described in the next section.

It is interesting to note that these three corpora have very different characteristics in terms of the average cluster size, percentage of data in the largest cluster and the proportion of positive and negative pairwise edges. These characteristics have a strong effect on the performance of resource bounded information gathering criteria. We leave the analysis of this effect as part of our future work.

It is also important to note that since these datasets were gathered from different sources, they also have different characteristics in terms of the amount of features available. For example, the Rexa corpus has very rich features like full names of the authors, author e-mail, institution, publication venue, etc. On the other hand, the Penn corpus only has the first initial and last names for each author and title. This leads to variation in performance of the classifiers.

There has been extensive work in the clustering community about the use of good evaluation metrics. Note that there

is a large number of negative examples in this dataset and hence we always prefer pairwise F1 over accuracy as the main evaluation metric.

Web as Feature vs. Web as Mention

In these experiments we investigate the effect of using information from the web as an additional feature vs. including additional documents gathered from the web as nodes in the graph. The maximum entropy classifier between two citations (c2c) is built using the features described in (Kanani, McCallum, & Pal 2007). The baseline results in Table 2 indicate the results of running partitioning on the graph with only citation mentions and no additional information from the web. We use 100 runs of the N-run stochastic graph partitioning algorithm with a temperature setting of 0.01. The Google Feature results in Table 2 refer to the results of running N-run stochastic graph partitioner on the graph with only citation mentions. The edge weights are set by running the same c2c classifier, in the presence of Google title feature. Clearly, using this additional feature significantly improves the F1 performance. Note that, this feature fails to show a significant impact on the Rexa corpus, as there are some large datasets with very high baseline recall present in this corpus.

The Web Mention row in Table 2 corresponds to the experiment with including additional web mentions in the graph. As described in Section 2, there are three parts of the matrix. The c2c classifier, c2w classifier and w2w classifier. Due to the sparsity of training data, we set the values of the lower right region of the matrix to zero, indicating no preference.

Hand labeling the data for the c2w classifier is very time consuming. Hence, we issue queries for only those citation pairs which are believed to be referring to the same real author, and label the resulting web pages as coreferents. We use following features. Appearance of raw author names from the citation, occurrence of title words, bigrams and trigrams (after removing stop words), author e-mail and institution bigrams and trigrams (if available) in the web document. There are many instances for which none of the feature values are set. In order to avoid the bias of the classifier affecting the edge weights, we set the value of zero on such instances.

After setting the weights on the three different types of edges as described above, we partition this larger graph and measure the precision, recall and F1 values on only the citation region of the graph. This is because, ultimately, we are interested only in the performance improvement in this region. Table 2 shows that use of information from the web as an additional mention clearly outperforms its use as a feature.

Corpus	Method	Pr	Rec	F1
DBLP	Baseline	88.55	44.33	59.09
	Google Feature	92.48	68.95	79.00
	Web Mention	91.89	92.57	92.22
Rexa	Baseline	84.66	97.44	90.60
	Google Feature	82.75	98.64	90.00
	Web Mention	84.22	96.39	89.90
Penn	Baseline	98.74	14.40	25.15
	Google Feature	98.57	17.37	29.55
	Web Mention	95.57	22.01	35.77

Table 2: Comparison of Web as a Feature and Web as a Mention

Corpus	WebMen	SetCover	InterClust	IntraClust
DBLP	180	53	16	84
Rexa	438	99	2	45
Penn	601	195	240	335
Total	1219	347	258	464

Table 3: Number of documents gathered by Set-cover, Inter-cluster and Intra-cluster Queries

RBIG: Set-cover

In order to implement the *globally greedy* Set-cover method, we first define the "cover" heuristic as follows. If any of the citation's title word trigrams match any of the web page's text trigrams, or if an email address mentioned in the citation is found on a web page, then we consider that citation "covered" by the web page. We also remove all the web pages which do not "cover" at least two citations, as described in the algorithm. The subset of web pages returned by the algorithm are added to the graph and we continue with the web-as-mention procedure to measure the performance.

Table 3 shows the number of documents obtained by applying this algorithm on the web mentions gathered using pairwise queries. The graph in Fig. 4 shows the corresponding pairwise F1 values. On the DBLP corpus, we observe that using 29.4% of the web mentions, we obtain 77.9% of the possible improvement. On the Rexa corpus, we see an improvement of 10.8% in precision, but at the cost of recall, using 22.6% of the web mentions. Set-cover method is not very effective in the case of Penn corpus, where we achieve only 15.3% of the possible improvement, by using 32.4% instances. This may be due to the sparsity of features in this corpus. Also, note that in the real world, the "purity" assumption that we made about web pages does not hold. In our future work, we would like to examine the effect of adding the web pages incrementally to analyze the effectiveness of this ordering criterion.

RBIG: Inter-cluster vs. Intra-cluster Queries

We now turn to experiments that selectively query the web to gather additional web mentions. In our case querying cost is negligible, hence we focus on the number of documents obtained by applying the two criteria. Note that the Rexa corpus shows very small number of documents for inter cluster queries due to the presence of few large size clusters.

On the DBLP corpus, Intra-cluster queries perform much better than inter-cluster. The same effect is observed in the individual datasets of the Rexa corpus, with the exception of a large high-recall cluster. On the Penn corpus, the effect is not very pronounced, however, we do see improvement over the baseline using intra-cluster queries. Overall, intra-cluster queries achieve a higher reduction in F1 error with

fewer additional documents, as compared to inter-cluster queries. It is interesting to note that the average citations to number of true authors ratio in the DBLP, Rexa and Penn corpora are 8.2, 6.4 and 22.15 respectively. This shows that performance of the two query selection criteria is heavily correlated with the characteristics of the dataset as discussed before.

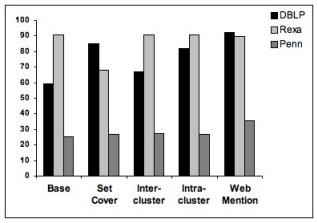

Figure 4: Pairwise F1 for various methods

RBIG: Intra-Setcover Hybrid Approach

Finally, we present the results of the hybrid approach on the DBLP corpus. In Fig. 5, the black series plots the ratio of the number of documents added to the graph in each method to the number of documents obtained by all pairwise queries. This represents cost. The gray series plots the ratio of the improvement obtained by each method to the maximum achievable improvement (using all mentions and queries). This represents benefit. For the Intra-Setcover hybrid approach, we achieve 74.3% of the total improvement using only 18.3% of all additional mentions.

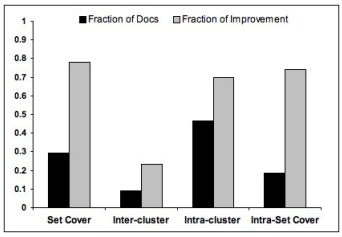

Figure 5: DBLP: For each method, fraction of the documents obtained using all pairwise queries and fraction of the possible performance improvement obtained. Intra-Setcover hybrid approach yields the best cost-benefit ratio

Conclusions and Future Work

We discuss methods for improving performance on author coreference task by leveraging web information. Building on our previous work, we present improved and more comprehensive results for the method in which we incorporate web documents as additional nodes in the graph. We also propose efficient strategies to select a subset of nodes to add to the graph and to select a subset of queries to gather additional nodes. We show that these methods, when applied appropriately, and taking data characteristics into consideration, can help achieve high performance gain using fewer resources. Finally, we show that using a hybrid approach, we achieve 74.3% of the total improvement using only 18.3% of all additional mentions.

It would be interesting to further investigate the correlation between the data characteristics and behavior of resource bounded information gathering criteria in this context. We have also been experimenting with alternative query methods. The connection to Set-cover problem opens up many interesting possibilities. We would also like to explore the value of information approach to solve this problem. The ultimate goal is to abstract away form the problem of author coreference and web data. We would like to formulate a theoretical analysis of an optimal query selection criteria when gathering external information to improve graph partitioning.

Acknowledgements

We thank Avrim Blum, Katrina Ligett, Chris Pal, Ramesh Sitaraman and Aron Culotta for helpful discussions. This work was supported in part by the Center for Intelligent Information Retrieval and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

Bhattacharya, I., and Getoor, L. 2006. A latent dirichlet model for unsupervised entity resolution. In *SDM*.

Dong, X.; Halevy, A. Y.; Nemes, E.; Sigurdsson, S. B.; and Domingos, P. 2004. Semex: Toward on-the-fly personal information integration. In *Workshop on Information Integration on the Web (IIWEB)*.

Grass, J., and Zilberstein, S. 2000. A value-driven system for autonomous information gathering. *Journal of Intelligent Information Systems* 14:5–27.

Han, H.; Zha, H.; and Giles, L. 2005. Name disambiguation in author citations using a k-way spectral clustering method. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL 2005)*.

Huang, J.; Ertekin, S.; and Giles, C. L. 2006. Efficient name disambiguation for large-scale databases. In *PKDD*, 536–544.

Kanani, P., and McCallum, A. 2007. Resource-bounded information gathering for correlation clustering. In *Conference on Computational Learning Theory, Open Problems Track*.

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

Kapoor, A., and Greiner, R. 2005. Learning and classifying under hard budgets. In *ECML*, 170–181.

McCallum, A., and Wellner, B. 2003. Object consolidation by graph partitioning with a conditionally-trained distance metric. *KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation*.