Improving Author Coreference by Resource-bounded Information Gathering from the Web

Pallika Kanani, Andrew McCallum, Chris Pal
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003 USA
{pallika, mccallum, pal} @ cs.umass.edu

Abstract

Accurate entity resolution is sometimes impossible simply due to insufficient information. For example, in research paper author name resolution, even clever use of venue, title and co-authorship relations are often not enough to make a confident coreference decision. This paper presents several methods for increasing accuracy by gathering and integrating additional evidence from the web. We formulate the coreference problem as one of graph partitioning with discriminatively-trained edge weights, and then incorporate web information either as additional features or as additional nodes in the graph. Since the web is too large to incorporate all its data, we need an efficient procedure for selecting a subset of web queries and data. We formally describe the problem of resource bounded information gathering in each of these contexts, and show significant accuracy improvement with low cost.

1 Introduction

Machine learning and web mining researchers are increasingly interested in using search engines to gather information for augmenting their models, *e.g.* [Etzioni *et al.*, 2004], [McCallum and Li, 2003], [Dong *et al.*, 2004]. However, it is impossible to query for the entire web, and this gives rise to the problem of efficiently selecting which queries will provide the most benefit. We refer to this problem as *resource-bounded information gathering from the web.*

We examine this problem in the domain of entity resolution. Given a large set of entity names (each in their own context), the task is to determine which names are referring to the same underlying entity. Often these coreference merging decisions are best made, not merely by examining separate pairs of names, but relationally, by accounting for transitive dependencies among all merging decisions. Following previous work, we thus formulate entity resolution as graph partitioning with edge weights based on many features with parameters learned by maximum entropy [McCallum and Wellner, 2004], and in this paper explore a relational, graph-based approach to resource-bounded information gathering.

The specific entity resolution domain we address is research paper author coreference. The vertices in our coreference graphs are citations, each containing an author name with the same last name and first initial.[1] Coreference in this domain is extremely difficult. Although there is a rich and complex set of features that are often helpful, in many situations they are not sufficient to make a confident decision. Consider, for example, the following two citations both containing a "D. Miller."

- Mark Orey and David Miller, Diagnostic Computer Systems for Arithmetic, Computers in the School, volume 3, #4, 1987
- Miller, D., Atkinson, D., Wilcox, B., Mishkin, A., Autonomous Navigation and Control of a Mars Rover, Proceedings of the 11th IFAC Symposium on Automatic Control in Aerospace, pp. 127-130, Tsukuba, Japan, July 1989.

The publication years are close; and the titles both relate to computer science, but there is not a specific topical overlap; "Miller" is a fairly common last name; 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 "Miller".

In this paper, we present two different mechanisms for augmenting the coreference graph partitioning problem by incorporating additional helpful information from the web. In both cases, a web search engine query is formed by conjoining the titles from two citations. The first mechanism changes the edge weight between the citation pair by adding a feature indicating whether or not any web pages were returned by the query. The second mechanism uses one of the returned pages (if any) to create an additional vertex in the graph, for which edge weights are then calculated to all the other vertices. The additional transitive relations provided by the new vertex can provide significant helpful information. For example, if the new vertex is a home page listing all of an author's publications, it will pull together all the other vertices that should be coreferent.

---

[1]Future work will address the problem of correctly merging names with typographic errors in the first initial and last name.

Gathering such external information for all vertex pairs in the graph is prohibitively expensive, however. Thus, methods that acknowledge time, space and network resource limitations, and effectively select just a subset of the possible queries are of interest. Learning and inference under resource limitations has been studied in various forms. For example, the value of information, as studied in decision theory, measures the expected benefit of queries [Zilberstein and Lesser, 1996]. Budgeted learning, rather than selecting training instances, selects new features [Kapoor and Greiner, 2005]. Resource-bounded reasoning studies the trade offs between computational commodities and value of the computed results [Zilberstein, 1996]. Active learning aims to request human labeling of a small set of unlabeled training examples [Thompson *et al.*, 1999], for example, aiming to reduce label entropy on a sample [Roy and McCallum, 2001].

In this paper we employ a similar strategy, and compare it with two baseline approaches, showing on 7 different data sets that leveraging web queries can reduce F1 error by 13.03%, and furthermore that, by using our proposed resource-bounded approach, 53.5% of this gain can be achieved with about 1% of the web queries. We also suggest that our problem setting will be of interest to theoretical computer science, since it is a rich extension to correlational clustering [Bansal *et al.*, 2002; Demaine and Immorlica, 2003].

2 Conditional Entity Resolution Models

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 global and consistent coreference decision. We use the term, "mention" to indicate the appearance of an author name in a citation and use x_i to denote mention $i = 1, \ldots, n$. Let y_{ij} represent a binary random variable that is true when mentions x_i and x_j refer to the same underlying author "entity." For each pair of mentions we define a set of l feature functions $f_l(x_i, x_j, y_{i,j})$ acting upon a pair of mentions. From these feature functions we can construct a local model given by

$$P(y_{i,j}|x_i, x_j) = \frac{1}{Z_x} \exp(\lambda_l f_l(x_i, x_j, y_{ij})), \qquad (1)$$

where $Z_x = \sum_y \exp(\lambda_l f_l(x_i, x_j, y_{ij}))$. In McCallum and Wellner [2003] a conditional random field with a form similar to (1) is constructed which effectively couples a collection of pairwise coreference models using equality transitivity functions $f_*(y_{ij}, y_{jk}, y_{ik})$ to ensure globally consistent configurations. These functions ensure that the coupled model assigns zero probability to inconsistent configurations by evaluating to $-\infty$ for inconsistent configurations and 0 for consistent configurations. The complete model for the conditional distribution of all binary match variables given all mentions $\mathbf{x}$ can then be expressed as

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left(\sum_{i,j,l} \lambda_l f_l(x_i, x_j, y_{ij}) + \sum_{i,j,k} \lambda_* f_*(y_{ij}, y_{jk}, y_{ik}) \right), \qquad (2)$$

where $\mathbf{y} = \{\mathbf{y_{ij}} : \forall_{\mathbf{i,j}}\}$ and

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp \left(\sum_{i,j,l} \lambda_l f_l(x_i, x_j, y_{ij}) + \sum_{i,j,k} \lambda_* f_*(y_{ij}, y_{jk}, y_{ik}) \right) \qquad (3)$$

As in Wellner and McCallum [2002], the parameters λ can be estimated in local fashion by maximizing the product of Equation 1 over all edges in a labeled graph exibiting the true partitioning. When $f_l(x_i, x_j, 1) = -f_l(x_i, x_j, 0)$ it is possible to construct a new undirected and fully connected graph consisting of nodes for mentions, edge weights $\in [-\infty, \infty]$ defined by $\sum_l \lambda_l(x_i, x_j, y_{ij})$ and with sign defined by the value of y_{ij}. In our work here we define a graph in a similar fashion 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 $P(y_{ij} = 1|x_i, x_j) - P(y_{ij} = 0|x_i, x_j)$ or the difference in the probabilities that that the citations v_j and v_k are by the same author. Note that the edge weights defined in this manner are in $[-1, +1]$. The edge weights in E_0 are noisy and may contain inconsistencies. For example, given the nodes v_1, v_2 and v_3, we might have a positive weight on $<v_1, v_2>$ as well as on $<v_2, v_3>$, but a high negative weight on $<v_1, v_3>$. 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.

[Bansal *et al.*, 2002] provide two polynomial-time approximation schemes (PTAS) for partitioning graphs with mixed positive and negative edge weights. We obtain good empirical results with the following stochastic graph partitioning technique, termed here *N-run stochastic sampling*.

Algorithm 1. – N-Run Stochastic Sampling:
We define a distribution over all edges in G_0, $P(w_i) \propto e^{-\frac{w_i}{T}}$ where T acts as temperature. At each iteration, we draw an edge from this distribution and merge the two vertices. Edge weights to the new vertex formulated by the merge are set to the average of its constituents and the distribution over the edges is recalculated. Merging stops when no positive edges remain in the graph. This procedure is then repeated $r = 1...N$ times and the partitioning with the maximum $\mathcal{F}$ is then selected.

3 Coreference Leveraging the Web

Now, consider that we have the ability to augment the graph with additional information using two alternative methods:

(A)..., H. Wang, ... Background Initialization..., ICCV,...2005.
(B)..., H. Wang, ... Tracking and Segmenting People..., ICIP, 2005.
(C)..., H. Wang, ... Gaussian Background Modeling..., ICASSP, 2005.
(D)**..., H. Wang, ... Facial Expression Decomposition..., ICCV, 2003.**
(E)..., H. Wang, ... Tensor Approximation..., SIGGRAPH. 2005.
(F)..., H. Wang, ... High Speed Machining..., ASME, (JMSE), 2005.

Figure 1: Six Example References

(1) changing the weight on an existing edge, (2) adding a new vertex and edges connecting it to existing vertices. This new information can be obtained by querying some external source, such as a database or the web.

The first method may be accomplished in author coreference, for example, by querying a web search engine as follows. Clean and concatenate the titles of the citations, issue this query and examine 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. Let f_g be this new boolean feature. This feature is then added to an augmented classifier that is then used to determine edge weights.

In the second method, a new vertex can be obtained by querying the web in a similar fashion, but creating a new vertex by using one of the returned web pages as a new mention. 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 $v_1, v_2 \epsilon V_0$, $v_3 \epsilon V1$, 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 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.

4   Resource Bounded Web Usage

Under the constraint on resources, however, we must select only a subset of edges in E_0, for which we can obtain the cor-
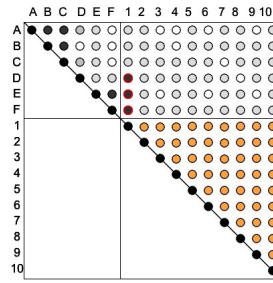

Figure 2: Extending a pairwise similarity matrix with additional web mentions. A..F are citations and 1..10 are web mentions.

responding piece of information i_i. Let $E_s \subset E_0$, be this set and I_s be the subset of information obtained that corresponds to each of the elements in E_s. The size of E_s is determined by the amount of resources available. Our objective is to find the subset E_s that will optimize the function $\mathcal{F}$ on graph G_0 after obtaining I_s and applying graph partitioning.

Similarly, in the case of expanded graph G', given the constraint on resources, we must select $V'_s \subset V_1$, to add to the graph. Note that in the context of information gathering from the web, $|V_1|$ is in the billions. Even in the case when $|V_1|$ is much smaller, we may choose to calculate the edge weights for only a subset of E_1. Let $E'_s \subset E_1$ be this set. The sizes of V'_s and E'_s are determined by the amount of resources available. Our objective is to find the subsets V'_s and E'_s that will optimize the function $\mathcal{F}$ on graph G_0 by applying graph partitioning on the expanded graph. We now present the procedure for the selection of E_s.

Algorithm 2. – Centroid Based Resource Bounded Information Gathering and Graph Partitioning For each cluster of vertices that have been assigned the same label under a given partitioning, we define the centroid as the vertex v_c with the largest sum of weights to other members in its cluster. Denote the subset of vertex centroids obtained from clusters as V_c. (We can also optionally pick multiple centroids from each cluster.) We begin with graph G_0 obtained from the base features of the classifier. We use the following criteria for finding the best order of queries: expected entropy, gravitational force, uncertainty-based and random. The uncertainty criteria uses the entropy of the binary classifier for each edge. For each of these criteria, we follow this procedure.

1. Partition graph G_0 using N-run stochastic sampling.

2. From the highest scoring partitioned graph G_i^*, find the subset of vertex centroids V_c

3. Construct E_s as the set of all edges connecting centroids in V_c.

4. Order edges E_s into index list I based on the criteria.

5. Using index list I, for each edge $e_i \subset E_s$

   (a) Execute the web query and evaluate additional features from result

   (b) Evaluate classifier for edge e_i with the additional features and form graph G_i from graph G_{i-1}

   (c) Using graph G_i, perform N-run stochastic sampling and compute performance measures

Criterion 1 - Expected Entropy

1. Force merge of the vertex pair of e_i to get a graph G_p

2. Peform N-run stochastic sampling on G_p. This gives the probabilities p_i for each of the edges in G_p

3. Calculate the entropy, H_p of the graph G_p as follows:
$H_p = -\sum_i P_i \log P_i$

4. Force split of the vertex pair of e_i to get a graph G_n

5. Repeat steps 2-3 to calculate entropy, H_n for graph G_n

6. The expected entropy, H_i for the edge e_i is calculated as:
$H_i = \frac{(H_p)+(H_n)}{2}$ (Assuming equal probabilities for both outcomes)

Criterion 2 - Gravitational Force

This selection criteria is inspired by the inverse squared law of the gravitational force between two bodies. It is defined as $F = \Gamma \frac{M_1 * M_2}{d^2}$, where Γ is a constant, M_1 and M_2 are analogous to masses of two bodies and d is the distance between them. This criteria ranks highly partitions that are near each other and large, and thus high-impact candidates for merging. Let v_j and v_k be the two vertices connected by e_i. Let C_j and C_k be their corresponding clusters. We calculate the value of F as described above, where M_1 and M_2 are the number of vertices in C_j and C_k respectively. We define $d = \frac{1}{x^{w_i}}$, where w_i is the weight on the edge e_i and x is a parameter that we tune for our method.

5 Theoretical Problem

There has been recent interest in the general problem of correlational clustering [Bansal *et al.*, 2002; Demaine and Immorlica, 2003]. We now present a new class of problems that are concerned with resource bounded information gathering under graph partitioning.

Consider the matrix as presented in Fig.2. Suppose we care about the partitioning in the upper left part of the matrix, and all the values in the upper right part of the matrix are hidden. As we have seen before, obtaining these values would impact the graph partitioning in the section that we care about.

Now, suppose, we have access to an adversarial oracle, who unveils these values in the requested order. In the worst case, no useful information is obtained till the last value is unveiled. In the best case, however, requesting a small fraction of the values, leads to perfect partitioning in the section that we care about. The question that we now ask is, what is the best possible order to request these values. Making reasonable assumptions about the nature of the oracle and imposing restrictions on the edge weights makes this problem interesting and useful. This is one way to formulate this problem. This paper opens up many such possibilities for the theory community.

6 Experimental Results

6.1 Dataset 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.

Corpus	# Sets	# Authors	# Citations	# Pairs
DBLP	18	103	945	43338
Rexa	8	289	1459	207379
Penn	7	139	2021	455155
Rbig	18	103	1360	126205

Table 1: Summary of Data set properties.

The 'Rbig' corpus consists of a collection of web documents which is created as follows. For every dataset in the DBLP corpus, we generate a pair of titles and issue queries to Google. Then, we save the top five results and label them to correspond with the authors in the original corpus. The number of pairs in this case corresponds to the sum of the products of the number of web documents and citations in each dataset.

All the corpora are split into training and test sets roughly based on the total number of citations in the datasets. We keep the individual datasets intact because it would not be possible to test graph partitioning performance on randomly split citation pairs.

6.2 Baseline, Web Information as a Feature and Effect of Graph Partitioning

The maximum entropy classifier described in Section 2 is built using the following features. We use the first and middle names of the author in question and the number of overlapping co-authors. The US census data helps us determine how rare the last name of the author is. We use several different similarity measures on the titles of the two citations like the cosine similarity between the words, string edit distance, TF-IDF measure and the number of overlapping bigrams and trigrams. We also look for similarity in author emails, institution affiliation and the venue of publication if available. We use a greedy agglomerative graph partitioner in this set of experiments and are interested in investigating the effect of using a stochastic partitioner.

The baseline column in Table 2 shows the performance of this classifier. Note that there is a large number of negative examples in this dataset and hence we prefer pairwise F1 over accuracy as the main evaluation metric. Table 2 shows that graph partitioning significantly improves pairwise F1. We also use area under the ROC curve for comparing the performance of the pairwise classifier, with and without the web feature.

Note that these are some of the best results in author coreference and hence qualify as a good baseline for our experiments with the use of web. It is difficult to make direct comparison with other coreference schemes [Han *et al.*, 2005] due to the difference in the evaluation metrics.

Table 2 compares the performance of our model in the absence and in the presence of the Google title feature. As described before, these are two completely identical models, with the difference of just one feature. The F1 values improve significantly after adding this feature and applying graph partitioning.

Method		AROC	Acc	Pr	Rec	F1
Baseline	class.	.847	.770	.926	.524	.669
DBLP	part.	-	.780	.814	.683	.743
W/ Google	class.	.913	.883	.907	.821	.862
DBLP	part.	-	.905	.949	.830	.886
Baseline	class.	.866	.837	.732	.651	.689
Rexa	part.	-	.829	.634	.913	.748
W/ Google	class.	.910	.865	.751	.768	.759
Rexa	part.	-	.877	.701	.972	.814
Baseline	class.	.688	.838	.980	.179	.303
Penn	part.	-	.837	.835	.211	.337
W/ Google	class.	.880	.913	.855	.672	.752
Penn	part.	-	.918	.945	.617	.747

Table 2: Effect of using the Google feature. Top row in each corpus indicates results for pairwise classification and bottom row indicates results after graph partitioning.

6.3 Expanding the Graph by Adding Web Mentions

In this case, we augment the citation graph by adding documents obtained from the web. We build three different kinds of pairwise classifiers to fill the entries of the matrix shown in Fig 2. The first classifier, between two citations, is the same as the one described in the previous section. The second classifier, between a citation and a web mention, predicts whether they both refer to the same real author. The features for this second classifier include, occurrence of the citation's author and coauthor names, title words, bigrams and trigrams in the web page. The third classifier, between two web mentions, predicts if they both refer to the same real author or not. Due to the sparsity of training data available at this time, we set the value of zero in this region of the matrix, indicating no preference. We now run the greedy agglomerative graph partitioner on this larger matrix and finally, measure the results on the upper left matrix.

We compare the effects of using web as a feature and web as a mention on the DBLP corpus. We use the Rbig corpus for this experiment. Table 3 shows that the use of web as a mention improves the performance on F1. Note that alternative query schemes may yield better results.

Data	Acc.	Pr.	Rec.	F1
Baseline	.7800	.8143	.6825	.7426
Web Feature	.9048	.9494	.8300	.8857
Web Mention	.8816	.8634	.9462	.9029

Table 3: DBLP Results when using Web Pages found by Google as Extra Mentions(Rbig).

6.4 Applying the Resource Bounded Criteria for Selective Querying

We now turn to the experiments that use different criteria for selectively querying the web. We present the results on test datasets from DBLP and Rexa corpora. As described earlier in Section 4, the query candidates are the edges connecting centroids of initial clustering. We use multiple centroids

Method	Precision	Recall	F1
Merge Only			
Expected Entropy	73.72	87.92	72.37
Gravitational Force	63.10	92.37	64.55
Uncertainty	64.95	87.83	63.54
Random	63.97	89.46	64.23
Merge and Split			
Expected Entropy	76.19	58.56	60.90
Gravitational Force	64.10	53.06	53.56
Uncertainty	66.56	54.45	55.32
Random	66.45	50.47	52.27
No Merge			
Expected Entropy	91.46	38.46	51.06
Gravitational Force	91.53	37.84	50.47
Uncertainty	87.01	41.91	52.70
Random	86.96	43.77	54.03

Table 4: Area Under Curve for different Resource Bounded Information Gathering criteria

and pick top 20% tightly connected vertices in each cluster. We experiment with ordering these query candidates according to the four criteria: expected entropy, gravitational force, uncertainty-based and random. For each of the queries in the proposed order, we issue a query to Google and incorporate the result into the binary classifier with an additional feature.

If the prediction from this classifier is greater than a threshold ($t = 0.5$), we force merge the two nodes together. If lower, we have two choices. We can impose the force split, in accordance with the definition of expected entropy. We call this approach "split and merge". The second choice is to not impose the force split, because, in practice, Google is not an oracle and absence of co-occurence of two citations on the web is not an evidence that they refer to different people. We call this approach "merge only". The third choice is to simply incorporate the result of the query into the edge weight.

After each query, we rerun the stochastic partitioner and note the precision, recall and F1. This gives us a plot for a single dataset. Note that the number of proposed queries in each dataset is different. We get an average plot by sampling the result of each of the datasets for a fixed number of points, n (n = 100). We interpolate when queries fewer than n are proposed. We then average across these datsets and calculate the area under these curves, as shown in Table 4.

These curves measure the effectiveness of a criteria in achieving maximum possible benefit with least effort. Hence, a curve that rises the fastest, and has the maximum area under the curve is most desired. Expected entropy approach, gives the best performance on F1 measure, as expected.

It is interesting to note that the gravitational-force-based criteria does better than the expected entropy criteria on recall, but worse on the precision. This is because this approach captures the sizes of the two clusters and hence tends to merge large clusters, without paying much attention to the 'purity' of the resulting clusters. The expected entropy approach, on the other hand, takes this into account and hence emerges as the best method. The force-based approach is a much faster approach and it can be used as a heuristic for
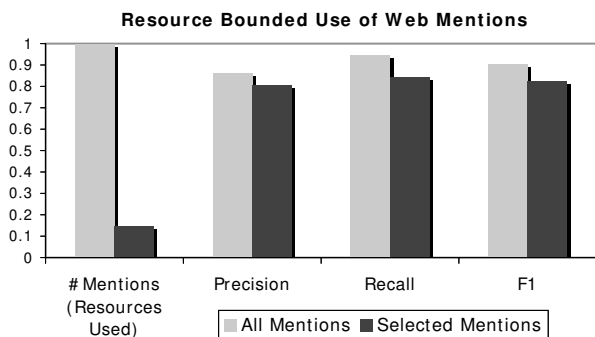
Resource Bounded Use of Web Mentions

Figure 3: Using the web and selectively obtaining new mentions.

We would like to include the word "Resource used" under # Mentions, if possible

very large datasets.

Both the criteria work better than uncertainty-based and random, except an occasional spike. All four methods are sensitive to the noise in data labeling, result of the web queries and sampling in stochastic graph partitioning, as reflected by the spikes in the curves. However, these results show that expected entropy approach is the best way to achieve maximum returns on investment and proves to be a promising approach to solve this class of problems, in general.

6.5 Resource Bounded Querying for Additional Web Mentions

We now present the results for efficiently querying the web and adding new mentions to the graph. We start with initial partitioning of citations for data sets in the DBLP corpus. We then pick up to two or three tightly connected citations in each cluster, clean their titles and remove stop words to form a query. Fig. 3 shows the comparison of the result of these queries with the result of performing all pairwise queries. By adding only 14.86% of the nodes to the graph, we can achieve 91.25% of the original F1. In other words, we gain most of the benefit by using a small fraction of the queries. Note that the resulting graph is smaller and hence is faster to process.

7   Conclusions and Future Work

We have formulated a new class of problems: resource bounded information gathering from the web in the context of correlational clustering, and have proposed several methods to achieve this goal in the domain of entity resolution. Our current approach yields positive results and can be applied for coreference of other object types, e.g. automatic product categorization. We believe that this problem setting has the potential to bring together ideas from the areas of active learning, relational learning, decision theory and graph theory, and apply them in a real world domain.

In future work we will explore alternative queries, (including input from more than two citations), as well as various new ways of efficiently selecting candidate queries. We are interested in investigating more sophisticated querying criteria in the case of web-as-a-mention. Additional theoretical work in the form of new formulations and bound proofs for these methods are also anticipated.

8   Acknowledgments

References

[Bansal *et al.*, 2002] N. Bansal, S. Chawla, and A. Blum. Correlation clustering. *In The 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 238–247, 2002.

[Demaine and Immorlica, 2003] Erik D. Demaine and Nicole Immorlica. Correlation clustering with partial information. In *RANDOM-APPROX*, page 1, 2003.

[Dong *et al.*, 2004] Xinyi Dong, Alon Y. Halevy, Ema Nemes, Stefan B. Sigurdsson, and Pedro Domingos. Semex: Toward on-the-fly personal information integration. In *Workshop on Information Integration on the Web (IIWEB)*, 2004.

[Etzioni *et al.*, 2004] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Web-scale information extraction in knowitall. In *Proceedings of the International WWW Conference, New York*. ACM, May 2004.

[Han *et al.*, 2005] Hui Han, Hongyuan Zha, and Lee Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL 2005)*, 2005.

[Kapoor and Greiner, 2005] Aloak Kapoor and Russell Greiner. Learning and classifying under hard budgets. In *ECML*, pages 170–181, 2005.

[McCallum and Li, 2003] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL*, pages 188–191, 2003.

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

[McCallum and Wellner, 2004] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing (NIPS)*, 2004.

[Roy and McCallum, 2001] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann, 2001.

[Thompson *et al.*, 1999] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *ICML*, page 406, 1999.

[Zilberstein and Lesser, 1996] Shlomo Zilberstein and Victor Lesser. Intelligent information gathering using decision models. *Technical Report 96-35, Computer Science Department University of Massachusetts at Amherst*, 1996.

[Zilberstein, 1996] Shlomo Zilberstein. Resource-bounded reasoning in intelligent systems. *ACM Comput. Surv*, 28, 1996.