Evaluating the Use of Social Networks in Author Name Disambiguation in Digital Libraries

Felipe Hoppe Levin, Carlos A. Heuser

Universidade Federal do Rio Grande do Sul, Brazil
{fhlevin, heuser}@inf.ufrgs.br

Abstract. Digital libraries have become an important source of information for scientific communities. However, by gathering data from different sources, the problem of duplicate and ambiguous information about author names arises. Traditional methods of name disambiguation use syntactic attribute information. However, recently the use of relationship networks has been studied in data deduplication. This article presents a study of the impact of adding social network analysis to traditional methods in the name disambiguation problem in digital libraries. Through experiments using subsets of real libraries, we show that the use of social network analysis significantly improves the quality of results.

Categories and Subject Descriptors: H. Information Systems [**H.m. Miscellaneous**]: Databases

Keywords: digital libraries, name disambiguation, relationship analysis, social networks

1. INTRODUCTION

Digital Libraries (DLs) are information systems for creating, searching and using an online collection of information. Apart from the content of the collection itself a DL usually stores metadata that describes the content (e.g., author, owner, publisher) as well as relationships between data. This data is constructed, collected and organized with the goal of supporting the information needs of a specific community [Borgman 1999].

In the scientific community, DLs have become an important source of information by presenting a centralized interface for searching and browsing publications. By grouping publications by metadata such as author, topic and publishing venue, users may employ the content of DLs for distinct analysis. An institution may use the information contained in a digital library to evaluate a researcher's production and make a hiring decision, for instance.

However, when evaluating authors using digital libraries, users often assume the content is free of errors and ambiguities, which is rarely the case. Digital Libraries gather data from different sources which often use different standards and abbreviations, leading to ambiguities. Of these ambiguities, name ambiguity is one of the most important cases and has been object of many studies and research. For example, two authors, Mark Jones and Matthew Jones, may have their names abbreviated to M. Jones in some publications. A search for M. Jones would present these publications as if belonging to the same author, leading to a problem known as mixed citation [Lee et al. 2005]. However, while some of Mark Jones's production is under the name M. Jones, other publications could be found at the author's full name and a search of Mark Jones would not present the author's complete production, leading to split citation [Lee et al. 2005]. Therefore, problems such as assignment of improper authorship or splitting an author's production due to representation by multiple names may occur due to name ambiguity.

Copyright©2010 Permission to copy without fee all or part of the material printed in JIDM is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

The task of solving the name ambiguity problem is known in literature as name disambiguation. Traditional disambiguation methods compare syntactic attribute information between ambiguous objects and, by the use of similarity functions, determine if these objects represent the same real entity. In author name disambiguation, name syntax is compared. In some methods, such as [Cota et al. 2007], other attributes, such as venue and publication title are compared by syntax similarity and are used as evidence that two author's are (or aren't) the same real author. This paper argues that additional semantic information, specifically the relationships between authors, may be used in association with these syntax based methods to improve the quality of results.

In this paper we use social network analysis as an evidence that two authors of two different papers in a dataset are more likely to be the same author in the real world. A social network is a collection of people - or actors - where each actor is tied to a subset of the others [Newman 2001]. In scientific collaboration networks, actors are authors which are tied when they have already collaborated on the production of some work. Collaboration between two authors implies an affinity between them: they may be interested in the same area or be affiliated to the same organization [Menezes et al. 2009]. If the distance between these two authors in the network is small, they have a greater chance of having the same interests and being affiliated to the same institution. Distance is an important measure because of a phenomenon first identified in [Milgram 1967] and known as the "small world" effect, which states that any two individuals are separated by a path of 6 to 7 individuals, on average. As our experiments demonstrate, paths with more than three individuals start to lose importance in the name disambiguation problem. Additionally, if there are many paths connecting two authors, it means they are more strongly related. Therefore, when two authors in a dataset are strongly related and have some degree of syntactic similarity, we can assume with high precision that they are duplicates of the same real author, as we will demonstrate in our experiments.

The main contributions of this article are the following:

(1) Presenting a set of relationship metrics to assess the importance of connections between two authors in the author social network,
(2) presenting a set of match functions that combine these relationship metrics with traditional syntax similarity functions, using more strict thresholds for unrelated authors and more relaxed thresholds for strongly related authors,
(3) evaluating the impact of adding social network analysis to traditional syntax based methods by testing the presented match functions over three real-world datasets,
(4) showing through experimental results that just connections with a distance of two ties need to be considered in order to improve quality in the name disambiguation task.

This paper is organized as follows. Section 2 describes the concept of author social network and its use in name disambiguation. Section 3 presents the match functions used for determining duplicates. Section 4 describes the experiments performed in order to evaluate our approach. Section 5 covers related work. The paper is finished in Section 6 with some conclusions and the description of future work.

2. AUTHOR SOCIAL NETWORK

In Figure 1 we have a list of records, representing papers with two to three authors each. The corresponding author social network is represented as a graph shown in Figure 2. In this graph, nodes represent authors (square boxes) and papers (rounded boxes). There are two kinds of edges. An edge represented by a straight line links a paper to one of its authors. Notice that a specific person may author several papers and therefore will be represented multiple times in the graph. Another type of edge represented by a dotted line links two authors with exactly the same name. Authors linked by a dotted line potentially represent the same person.

```
<P1; Robert Walker; Ben Goldman; Carl Parker>
<P2; Carl Parker; Robert D. Walker>
<P3; Ben Goldman; Ruth Adams>
<P4; Rob Walker; Ruth Adams; George Brown>
<P5; R. Walker; J. Smith; D. Johnson>
```

Fig. 1. A List of Papers

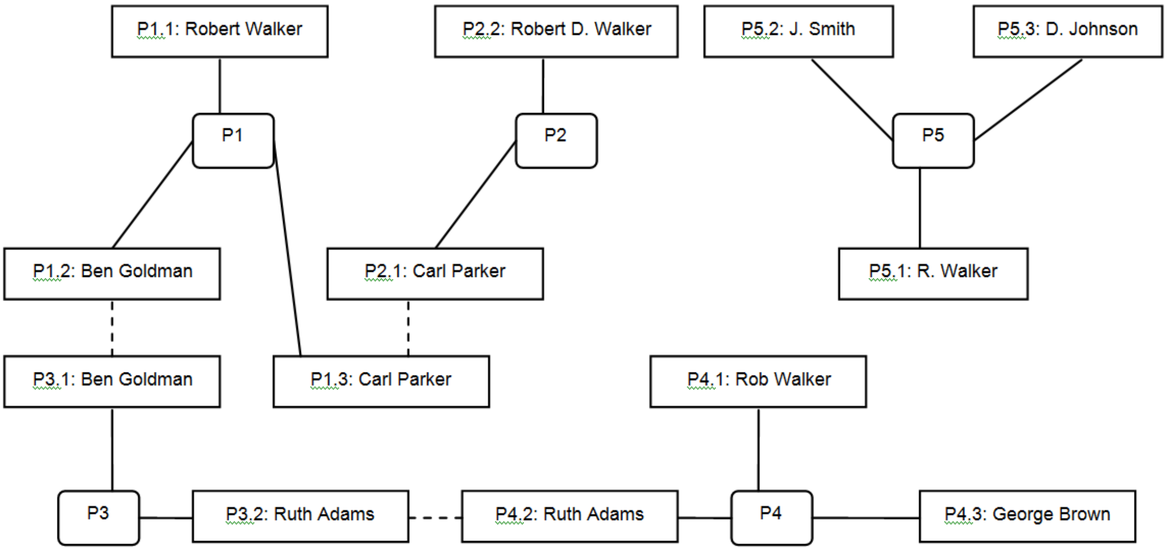

Fig. 2. Author Social Network

When comparing two authors, we use the social network to assess if two authors are the same person. For example, when comparing authors P1.1 and P2.2, we see that, additionally from having a great syntactic similarity, there is a path linking them through P1 and P2, and, therefore, they are socially related. This evidence, as we will demonstrate in our experiments, means they are much more likely to be the same person than if they had only syntactic similarity but no relationship between them. Authors P1.1 and P4.1 also have similar names, but they are more distant in the graph, as the path between them goes through 3 papers, so they are less likely to be the same person. Authors P1.1 and P5.1, despite having similar names, are even less likely to be the same person, for they have no relationship at all.

Formally the relationship between two authors is represented by the Relationship Distance (RD). In social networks, the distance between two actors is the length of the shortest path between them [Newman 2003]. In our approach, the path length between two authors is the number of papers in the path linking them. We can define RD as follows:

Relationship Distance (RD). Let a1, a2 be two authors being compared. Then, RD(a1, a2) is the length of the shortest path between them, returning 0 if no path exists.

To illustrate this, we change the graph of Figure 2, by adding the node P2.3, representing author reference George Brown, to P2 and linking this new node to the node P4.3, which is shown in Figure 3. When comparing nodes P1.1 and P2.2 we now have two paths between them, one going through

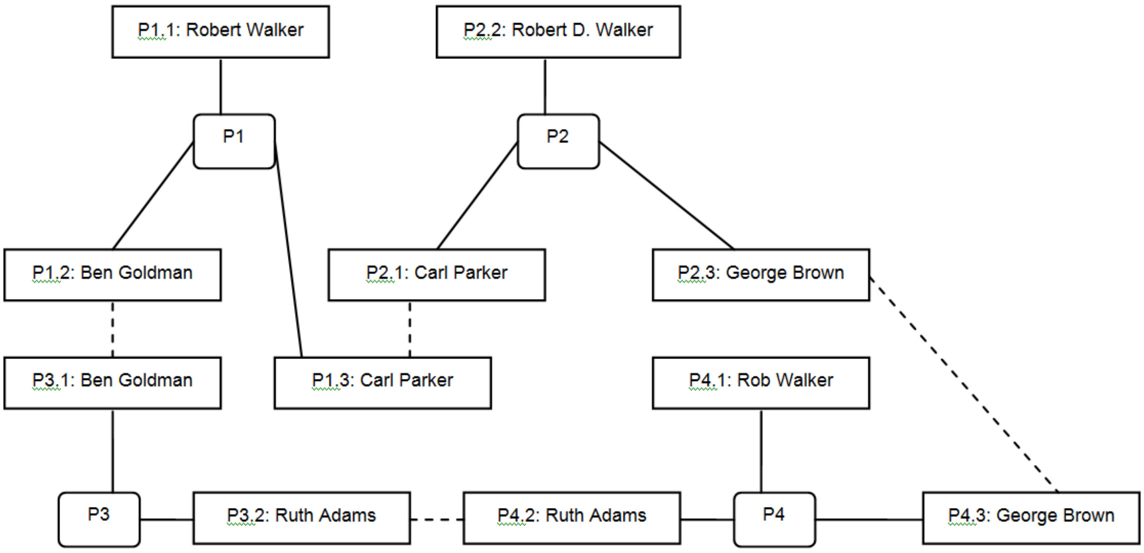

Fig. 3. Modified Author Social Network

P1-P2 and another going through P1-P3-P4-P2. Therefore, we have two paths of lengths two and four, respectively. RD between these two authors is two, the shortest path length between them.

Our approach uses RD as a way to measure the importance of the relationship, as lower distance means that two authors are more closely related and therefore have a greater chance of being the same person.

Another concept is the Relationship Existence (RE). RE returns true if there is relationship between two authors at a maximum distance d, and false otherwise. For example, RE between P1.1 and P2.2 at distance 2 is true in Figure 3, for they have RD = 2. Authors P1.1 and P4.1 have RD = 3, so RE at distance 2 or lower is false, but RE at distance 3 or greater is true. Formally, Relationship Existence is defined as follows:

Relationship Existence (RE). Let a1, a2 be two authors being compared and d an integer. Then, RE(a1, a2, d) is true if $1 \leq$ RD(a1, a2) $\leq$ d, and false otherwise.

Relationship Quantity (RQ) is the number of authors related to a specific author, considering some Relationship Distance. In Figure 3, author P4.1 is related to authors P4.2 and P4.3 at RD=1. Thus, its RQ at distance one is two. At distance two, it is related to P2.1, P2.2 and P3.1. Therefore, its RQ at distance two is three, and its RQ at distances one and two is five. Relationship Quantity is used to measure how likely the duplicates of an author will be related to it. If an object has a low RQ in the dataset, duplicates are likely not related to it, since it is not related to many objects, and we may consider authors not related to it as duplicate candidates, increasing recall. But if an object has a high RQ, duplicates are most likely related to it and we may only consider socially related authors as duplicate candidates, increasing precision.

Relationship Quantity (RQ). Let a be an author, A the set of authors in the dataset and d an integer. Then, RQ(a, d) = |B|, where for all b $\in$ B, $1 \leq$ RD(a, b) $\leq$ d and B $\subset$ A.

Relationship Strength (RS) is the number of paths between two authors. As stated earlier, in the example, there are two paths between P1.1 and P2.2, and therefore, the Relationship Strength between

P1.1 and P2.2 equals two. The greater the RS between two authors the stronger the relationship between them, increasing the possibility of duplicity.

Relationship Strength (RS). Let a1, a2 be two distinct authors and d an integer. Then, RS(a1, a2, d) is the number of paths between a1 and a2 with length d or lower in the social network graph.

3. DETERMINING DUPLICATES

Duplicate authors are determined using a match function. A match function is defined in [Benjelloun et al. 2006] as a function that takes two objects as input, returning true if they represent the same entity or false otherwise. In this article we compare different match functions to show that functions using both name syntax and co-author relationships are more effective than functions that consider only name syntax.

When comparing name syntax, a string similarity function is used. Two similarity functions will be used in experiments in this paper: the Levenshtein edit distance [Levenshtein 1966] and the trigram similarity function [Elmagarmid et al. 2007]. The Levenshtein distance is the number of character transformations (insertions, deletions and replacements) needed to transform string a into string b, normalized by the largest string size. Trigram similarity is the number of equal trigrams (sequences of 3 characters) normalized by the total number of trigrams. Both functions return a value from 0 to 1 representing the level of similarity between the strings, 1 meaning they are equal.

To determine duplicates using a similarity function we must establish a threshold. When similarity between two names is greater than the given threshold, they are considered duplicates.

In our experiments, we combine a string similarity function with the relationship measures defined in section 2 and evaluate the impact produced by the relationship measures in the quality of results. There are many ways to combine these measures with syntactic similarity. A match function composed only by a syntactic function can be defined as follows:

Syntactic Match Function (SynM). If Sim(a1, a2) > k then SynM(a1, a2, k) = true else SynM(a1, a2, k) = false, where Sim(a1, a2) is the similarity between two authors and k a given threshold.

We can combine the RE measure with the function Sim(a1, a2), creating a new match function as follows:

Relationship Match Function 1 (RelM1). If Sim(a1, a2) > k and RE(a1, a2, d) then RelM1(a1, a2, k, d) = true else RelM1(a1, a2, k, d) = false

This function will only consider as matches authors with similarity greater than k and with at least one relationship at distance d or lower, increasing precision.

To allow one threshold for related authors and a different threshold for unrelated authors, we will define a different function as follows:

Relationship Match Function 2 (RelM2). If Sim(a1, a2) > k or (Sim(a1, a2) > l and RE(a1, a2, d)) then RelM2(a1, a2, k, d) = true else RelM2(a1, a2, k, d) = false

By allowing one threshold for unrelated authors and another for related authors, we may use a lower threshold for related authors, increasing the recall of SynM while minimizing precision loss.

To use the RQ measure, we define a third function:

Relationship Match Function 3 (RelM3). If (Sim(a1, a2) $>$ k and (RQ(a1, d) $<$ q or RQ(a2, d) $<$ q)) or (Sim(a1,a2) $>$ l and RE(a1, a2, d) and RQ(a1,d) $\geq$ q and RQ(a2,d) $\geq$ q) then RelM3(a1, a2, k, l, d, q) = true else RelM3(a1, a2, k, l, d, q) = false

This function will use RQ to determine which threshold, k or l, will be used. If RQ of one of the authors is less than q, meaning this author has low connectivity in the social network, then they don't have to be related, but they need to have a greater similarity (assuming k greater than l). If both authors have RQ greater or equal to q, meaning a good connectivity, they need to have a lower similarity, but they must be related.

The last function is a variant of RelM3 and uses RS as a way to increase recall:

Relationship Match Function 4 (RelM4). If (Sim(a1, a2) $>$ k and (RQ(a1, d) $<$ q or RQ(a2, d) $<$ q)) or (Sim(a1,a2) $>$ l and RE(a1, a2, d) and RQ(a1,d) $\geq$ q and RQ(a2,d) $\geq$ q) or (Sim(a1, a2) $>$ m and RS(a1, a2, d) $\geq$ s) then RelM4(a1, a2, k, l, m, d, q, s) = true else RelM4(a1, a2, k, l, m, d, q, s) = false

This function adds another threshold to RelM3, which will be used if the authors share a relationship strength of s or greater. This means that, even if they have a low connectivity, if their relationship is strong enough we can relax the similarity threshold, increasing recall.

4. EXPERIMENTAL RESULTS

4.1 Datasets

In order to evaluate the improvement of relationship analysis in name disambiguation over real data, we used three different datasets. The Cora dataset was created by Andrew McCallum [McCallum et al. 2000], consisting in 1878 citations to real papers. This dataset has been hand-clustered into groups referring to the same paper and is available at the author's web page. Since our goal is to find duplicate authors, not papers, we hand-clustered the authors in the dataset using information available on the authors' web pages and other sources on the web, resulting in 178 different authors and 1341 duplicate pairs.

The second dataset is a subset of the **BDBComp**[1] digital library and has been used in [Oliveira et al. 2005]. The subset was made available to us by the authors of that paper. It is made up of 361 papers first authored by people with the most frequent last names in BDBComp, having 674 duplicate author pairs. In our experiments, only the first author of each paper was compared in the disambiguation process, since only first authors were hand-clustered by the authors of [Oliveira et al. 2005]. However, all co-authors were used to link these first authors in the Author Social Network.

The last dataset is a subset of the **DBLP**[2] digital library and was extracted and evaluated by us. We selected papers from the database authored by people whose names start with letter 'a' and which contain the string 'silva', the most common Brazilian surname in the library, resulting in 371 papers with 773 duplicate pairs of authors. The names with the string 'silva' were considered as the ambiguous ones, being subject to the disambiguation process, and all related authors on DBLP were used to create the Author Social Network, using a maximum relationship depth of 4. The author clusters have been manually generated using information available on the author's web pages and additional information found on the web.

[1] BDBComp: http://www.lbd.dcc.ufmg.br/bdbcomp/

[2] DBLP: http://dblp.uni-trier.de/

Table I. Hypothetical Test Results

Threshold	Recall	Precision
0.9	21.3%	100.0%
0.7	54.5%	89.2%
0.5	73.1%	62.3%
0.3	92.3%	31.2%
0.1	100.0%	1.7%

Table II. Interpolated Results Using 5 Recall Points

Recall	Precision
0%	100.0%
25%	89.2%
50%	89.2%
75%	31.2%
100%	1.7%

4.2 Evaluation Measures

For the experiments performed in this paper, the quality of results is measured using typical information retrieval metrics: recall, precision and F-measure [Salton and McGill 1983]. Traditional F-measure was used, attributing the same weight to precision and recall.

In Information Retrieval, similarity functions are often evaluated using recall/precision curves [Manning et al. 2008]. For such evaluations, first a query object (an author name, for example) is compared to all the objects in the dataset using the similarity function and then results are ranked by similarity. Starting at the top of the rank, we compute precision at specific recall points (usually eleven points, from 0% to 100% with 10% intervals). This process is made for several queries and average precision is calculated at each recall point. Then, results are interpolated, meaning if a recall point has a lower precision value than recall point b and recall b is also a higher recall point, a assumes the same precision value as b. Finally, the recall/precision curve is constructed for the similarity function and is compared to curves from other similarity functions. This way, it is possible to compare precision at several recall points, presenting an evaluation which is independent of thresholds chosen when comparing two functions.

Because the match functions defined in section 3 return a Boolean value instead of a real number, it is impossible to rank the results. To compare match functions using recall/precision curves we used a different method to create them. We tested several thresholds for each match function, starting with very strict thresholds, aiming at high precision, and ending with very relaxed thresholds for high recall. Every test was made using the entire dataset. The recall/precision curve was created plotting the obtained precision values for every recall point. The data has been interpolated, meaning that even if we don't get a recall value of 10% in our tests, if our smallest recall value is 21.3%, with 100% precision, recall points 10% and 20% will assume 100% precision. To illustrate this, Table I shows hypothetical test results while evaluating some similarity function and Table II shows the interpolated results in 5 recall points. We used 11 recall points in our experiments.

4.3 Experiments

In the first experiment, we evaluated the match functions defined in section 3 in the three datasets. We tested several thresholds, from 0 to 1, with intervals of 0.25. Distance d=2 was used for all Author Social Network based match functions. Table III shows the best F-measure results of experiments using Levenshtein similarity, while Table IV shows the same comparison using trigram similarity.

Table III. Match Functions using Levenshtein Similarity

	Cora			BDBComp			DBLP		
	Prec.	Recall	F-meas.	Prec.	Recall	F-meas.	Prec.	Recall	F-meas.
SynM	78.04%	70.25%	73.94%	70.97%	52.23%	60.17%	82.20%	75.29%	78.60%
RelM1	99.63%	20.28%	33.71%	93.85%	43.03%	59.00%	100.00%	38.03%	55.11%
RelM2	78.30%	71.59%	74.80%	69.47%	78.34%	73.64%	97.39%	77.10%	86.06%
RelM3	78.13%	70.83%	74.31%	76.65%	75.96%	76.30%	97.35%	75.94%	85.32%
RelM4	78.33%	71.96%	75.01%	76.83%	81.16%	78.93%	97.60%	89.39%	93.32%

Table IV. Match Functions using Trigram Similarity

	Cora			BDBComp			DBLP		
	Prec.	Recall	F-meas.	Prec.	Recall	F-meas.	Prec.	Recall	F-meas.
SynM	99.17%	88.67%	93.62%	57.88%	73.00%	64.57%	96.81%	74.64%	84.30%
RelM1	100.00%	31.02%	47.35%	76.25%	54.30%	63.43%	99.38%	41.27%	58.32%
RelM2	93.63%	89.86%	91.70%	72.07%	83.83%	77.50%	95.35%	95.47%	95.41%
RelM3	99.11%	91.13%	94.95%	81.41%	77.39%	79.30%	96.35%	91.98%	94.11%
RelM4	90.83%	91.57%	91.20%	81.24%	83.53%	82.37%	95.72%	95.47%	95.60%

As we can see in both tables and in all datasets, when comparing RelM1 to the syntactic match function SynM, there is a significant improvement in precision, which confirms our hypothesis that related authors are much more likely to be duplicates than unrelated ones. In BDBComp using Levenshtein, precision was improved by 23% and in DBLP by 18%, reaching 100% precision. However, when using RelM1, which limits matches to related authors only, recall is drastically reduced, resulting in a reduced F-measure when comparing RelM1 to SynM. This shows that, although related authors are more likely to be duplicates, there are many duplicate unrelated authors that are ignored by RelM1.

RelM2 tries to fix this problem by establishing one threshold for any pair of authors and a lower, more relaxed threshold for related authors. As our results show, this significantly improves recall with little reduction - and sometimes with gain - of precision, improving F-measure in all scenarios, except for Cora dataset with trigrams, were we had a 2% reduction. We had a gain in recall in all scenarios, and 13% F-measure improvement for BDBComp with Levenshtein and 11% improvement for DBLP with trigrams.

With match function RelM3, as explained earlier, the relationship quantity (RQ) is used. In our experiments, we used a higher value for threshold k and a lower value for threshold l and thus authors with high RQ have to be related but need a lower name similarity, while authors with low RQ don't have to be related but need a higher similarity. We used q = 3 for DBLP and BDBComp datasets and for the Cora dataset, which has duplicate papers and, by consequence, much more connections per author, we used q = 25. Our results show that RelM3 improves recall and F-measure in all scenarios when comparing it to SynM. Precision is also improved in 4 out of 6 scenarios.

RelM4 is similar to RelM3, but uses Relationship Strength (RS) to add a third threshold, which will be used for pairs with a minimum RS between them. We used s = 2 for all datasets. Results show RelM4 had the best F-measure performance for 5 out of 6 scenarios when comparing all match functions in the three datasets using both trigrams and Levenshtein distance.

The chart in Figure 4 shows F-measure results in each dataset by match function using Levenshtein similarity, while Figure 5 shows the same comparison for experiments using trigram similarity. Our experiments show that by using the Author Social Network the performance of both similarity functions has been significantly improved in the DBLP and BDBComp datasets, with RelM4 having the best performance.

In the Cora dataset, the improvement was not as significant, and match functions RelM2 and

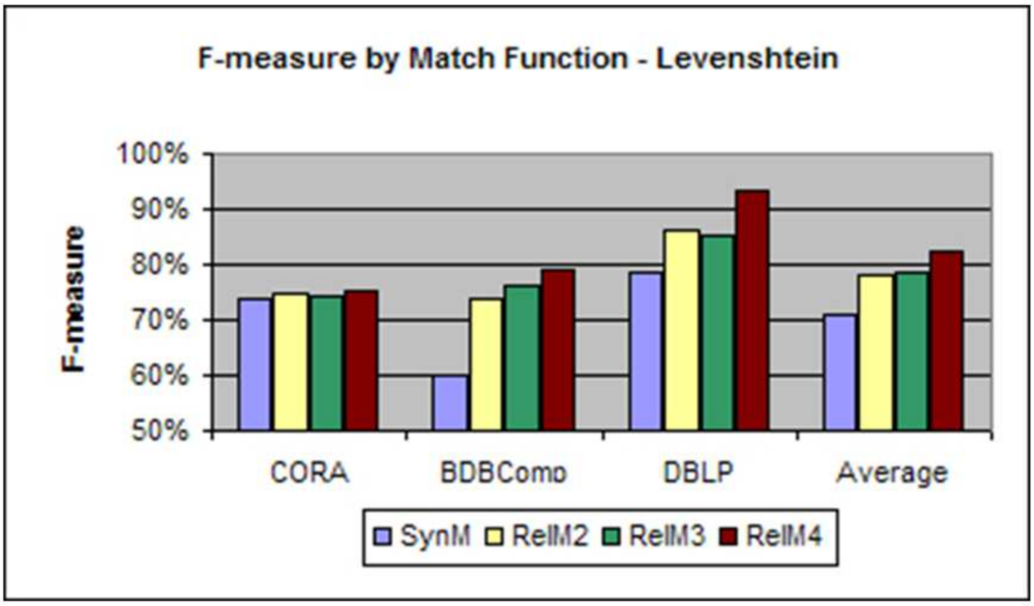

Fig. 4. F-measure by Match Function using Levenshtein Similarity

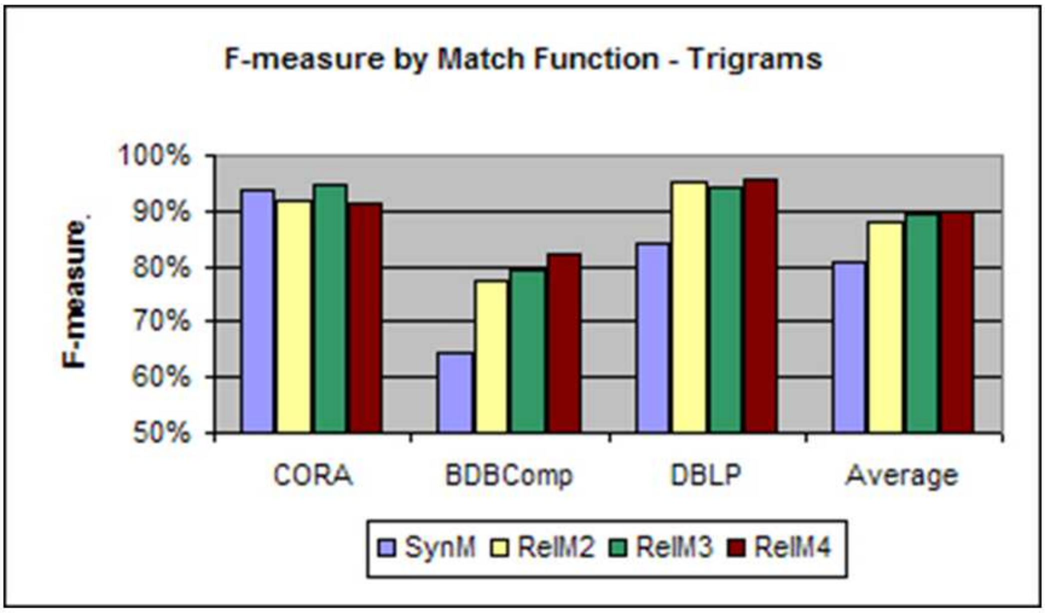

Fig. 5. F-measure by Match Function using Trigram Similarity

RelM4 performed worse than the original SynM when using trigram similarity. One of the reasons for this is because SynM with trigram similarity performed very well and there wasn't much to be improved. Also, some author names in the Cora dataset contain typos and, as there are duplicate papers using different abbreviation standards, names are abbreviated in many different ways resulting in more name variations than in the DBLP and BDBComp datasets, which negatively affects the Author Social Network. This happens as direct links are created between authors with the exact same name in the graph and by having several variations of the same name, many of these links are

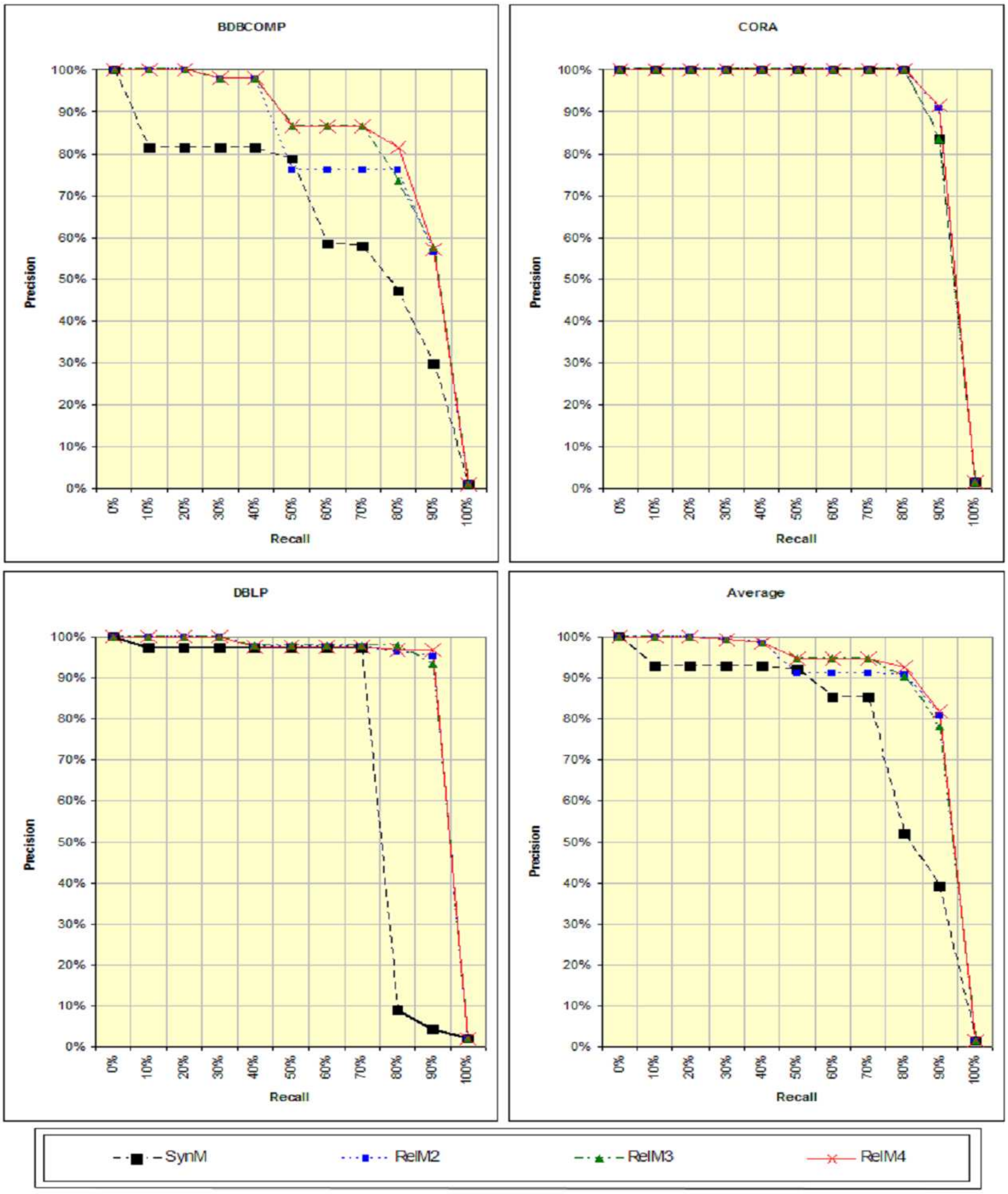

Fig. 6. Recall/Precision Curves for Match Functions using Trigram Similarity

not created. In DBLP and BDBComp, there less name variations for each author and therefore one author's connections will not be as split as in Cora. However, even with these problems, using the Author Social Network in Cora didn't decrease quality significantly and even improved results in most cases.

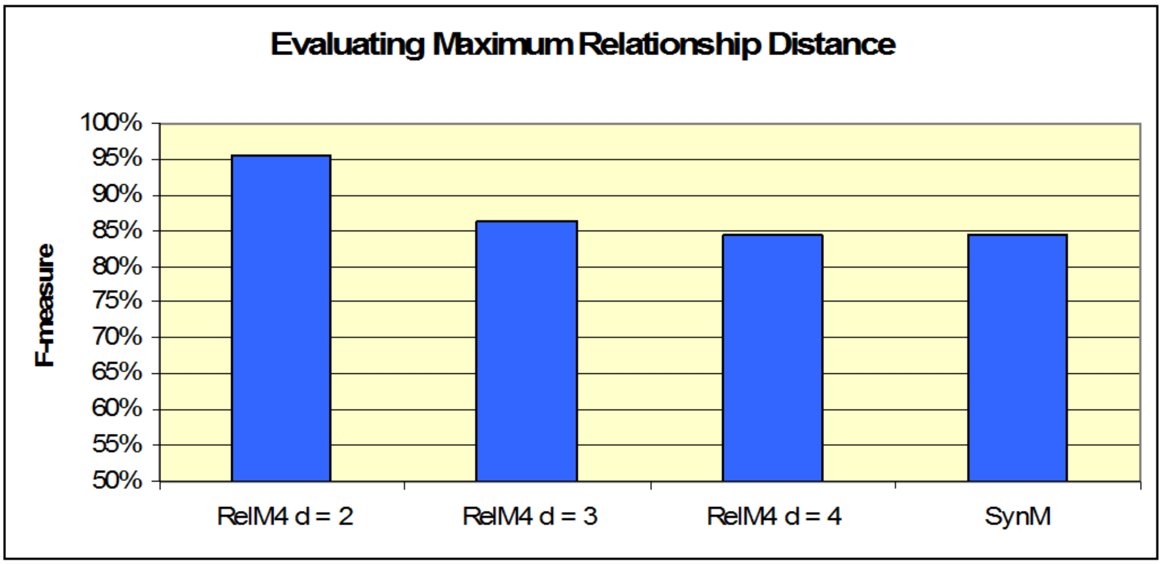

Fig. 7. Evaluation of Different Maximum Distances on DBLP Dataset

In Figure 6, match functions SynM, RelM2, RelM3 and RelM4 with trigram similarity are compared using recall/precision curves. For Cora dataset, behavior is very similar in all match functions, and even SynM maintains 100% precision until the 80% recall point. At 90% recall, RelM2 and RelM4 show better results when compared to SynM, demonstrating that even when the original similarity function performs well, using the Author Social Network can still improve results. On DBLP and BDBComp datasets, however, improvements are higher. Until the point of 70% recall results show a very high precision in all match functions on DBLP, but after the 80% recall point, there is a sudden drop in precision in SynM, while the social network based functions continue to have high precision. On BDBComp, the social network based functions have a better performance when compared to SynM through all recall points, with the only exception of the 50% recall point, when SynM performs slightly better than RelM2.

As the average of the three datasets demonstrates, RelM4 had the best results in our experiments through all levels of recall. RelM2 had the second best precision when working with high recall while RelM3 had the second best precision when working with low recall.

To demonstrate our approach's improvements to the original SynM are statistically significant, we performed a Wilcoxon Test [Wilcoxon 1945], comparing the best results from RelM2, RelM3 and RelM4 to the best results of SynM on DBLP using trigram similarity, previously shown on Table IV. The Wilcoxon Test is an alternative to Student's T-Test when the samples are not normally distributed, which is the case here. We used 1000 samples for each test and p values were lower than 0.0001 in all tests. Since this values are lower than the statistical significance threshold of 0.01, this demonstrates that all three functions have statistically superior performance when compared to the original SynM.

Table V. Percentage of Connected Author Pairs on DBLP Dataset

Maximum Distance	% Connected Pairs
2	2.05%
3	18.15%
4	84.35%

Finally, in the last experiment we compared the match functions using different d values, which represents the distance between two authors. As stated earlier, in our approach the distance between two authors is the minimum path length between them, and path length is calculated as the number of publications in the path. We used d values from 2 to 4, as distance 1 would only compare an author to its co-authors, which would not improve results unless an author name appeared twice in a publication. Results have shown that as maximum distance increases beyond 2, recall increases but precision falls drastically. To increase precision, thresholds also have to be increased, which makes recall decrease. In the end, $d = 2$ provided the best F-measure results, as is demonstrated in Figure 7, which shows results of RelM4 on the DBLP dataset. F-measure decreases as maximum distance increases and, at maximum distance 4, RelM4 performance becomes equal to SynM. This happens because at $d = 4$ every author is connected, on average, to 84.35% of the other actors in the DBLP dataset, as it is demonstrated on Table V, which indicates the presence of the "small world" effect.

4.4 Examples of Failure Cases

In this section, we show a few examples where our approach failed, giving reasons for this. All examples are from the BDBComp dataset, using the RelM4 match function with trigram similarity. In the first example, author references "Ana Cristina B. da Silva" and "A. C. da Silva", which represent the author Ana Cristina Barbosa da Silva, were not considered duplicates. This happened because both author references were not connected to any other authors in the dataset, having $RQ = 0$, so they were compared to a higher, more strict threshold and the match function failed.

In another example, author references "Antonio Alberto Fernandes de Oliveira" and "A. Oliveira", which represent the same real author, were not matched. Although they have RQ of 5 and 11, respectively, and are connected by the Author Social Network, their trigram similarity is lower than all thresholds established.

The last example shows a case where two author references where considered duplicates by the match function when they are actually not. Both were referenced as "A. Oliveira", but one of them is from the author Antonio Alberto Fernandes de Oliveira and the other from Arnaldo Oliveira. Since similarity between both author references is 100%, even though they are not related, the match function returned true.

5. RELATED WORK

Among the large research effort on the area of name disambiguation, which recently has received considerable attention due to the authorship assignment problem in digital libraries, there have been many pieces of research that are related to the use of graphs and co-authorship relations. In this section we present a short review of some pieces, comparing them to our research.

In [Nin et al. 2007], co-authorship networks are used to reduce the number of comparisons, and therefore increase computational performance, by semantic blocking. Blocking is a method used in data disambiguation in which objects are clustered by some function that is less computationally expensive than the actual match function, which will be used to compare only objects within the same cluster. Their technique creates blocks by clustering objects that are connected in the co-authorship network within a maximum distance d and then uses a syntactic similarity function to compare objects inside the block. The problem with this approach is that, as demonstrated in section 4.3, in typical digital libraries, there are many duplicate authors which are not connected in the network, and only comparing connected authors will decrease recall significantly.

A generic approach has been presented in [Kalashnikov and Mehrotra 2006], using the entity-relationship graph to solve the reference disambiguation problem, which is very similar to our problem. The difference is that there is a reference of real entities and the problem consists in linking the entities in the dataset to the real entities. Although generic, the example given in [Kalashnikov and Mehrotra 2006] is the author name disambiguation problem. For every author in the dataset, their approach uses a syntactic similarity function in a first step to disambiguate authors. The second step uses co-authorship relations to disambiguate only those references that could not be disambiguated in the first step. One of the differences in our approach is that we use both syntactic and semantic information in the same step of disambiguation, increasing the importance of the co-authorship information.

The method presented in [Cota et al. 2007] uses, along with the author name, evidences such as paper title, paper venue and co-author list to disambiguate authors. The difference in our approach is that, although not using paper title and venue evidences, we create an Author Social Network from the co-authors lists, using more elaborate social network evidences, while in [Cota et al. 2007] co-author lists are only compared syntactically.

In [Bhattacharya and Getoor 2007], authors are compared collectively, as clusters, instead of individually. As in our approach, their method uses co-authorship relations as evidence that author names represent the same real author. To use this evidence, however, [Bhattacharya and Getoor 2007] uses a neighborhood similarity value, in which author names need to have a similar set of co-authors to be considered the same author. Sets of co-authors are also compared on [Kang et al. 2009], which uses searches on the web to obtain these sets. In [Malin 2005], a Social Network similarity is calculated as the probability from author a to reach author b and this similarity is used as evidence to match author names. And in [On et al. 2006], a context graph is constructed for each entity, using co-authorship relations for example, and similarity between context graphs is measured. The main difference between these approaches and ours is that in our approach, instead of calculating the similarity of the relationship networks, author names need only to be linked and the strength or even the existence of this link will define if the attribute similarity needs to be strict or relaxed.

There are also many approaches which aim to improve author name disambiguation that do not use social networks and graphs. The methods presented in [Han et al. 2005], [Huang et al. 2006] and [Treeratpituk and Giles 2009], for example, are based on Machine Learning techniques and [Pereira et al. 2009] uses information extracted from the web as evidence for matching author names.

6. CONCLUSIONS

In this paper, we have evaluated the use of social networks to solve the author name disambiguation problem in digital libraries. Also, we have presented a set of relationship metrics to establish the existence and measure the strength and importance of connections between authors in the Author Social Network. We introduced a set of match functions that combine these metrics with traditional similarity functions. Experimental results showed that the use of social networks significantly improves the performance of syntax based similarity functions.

In [Levin and Heuser 2010] we have complemented this study by evaluating the impact of adding the relationship metrics presented in this article to other evidences specific to the digital libraries domain, such as title similarity and venue similarity. We presented an algorithm that uses Genetic Programming to create match functions, combining a set of different evidences. Our experimental results have shown that when the set of evidences used to generate match functions included our relationship metrics, the resulting match functions achieved a significantly higher performance than match functions generated by a set of evidences that did not include the relationship metrics. This shows that our relationship metrics can be used to improve not only name similarity functions, but also complex match functions. The match functions generated by our Genetic Programming approach were able to compete with the state-of-the-art method presented in [Cota et al. 2007].

7. ACKNOWLEDGEMENTS

This work has been partially supported by the Brazilian National Institute of Science and Technology for the Web (CNPq Grant No. 573871/2008-6) and by CNPq project No. 550891/2007-2.

REFERENCES

Benjelloun, O., Garcia-Molina, H., Kawai, H., Larson, T. E., Menestrina, D., Su, Q., Thavisonboon, S., and Widom, J. Generic Entity Resolution in the SERF Project. *IEEE Data Engineering Bulletin* vol. 29, pp. 13–20, 2006.

Bhattacharya, I. and Getoor, L. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data* vol. 1, pp. 1–36, 2007.

Borgman, C. L. What are digital libraries? Competing Visions. *Information Processing and Management* vol. 35, pp. 227–243, 1999.

Cota, R., Gonçalves, M. A., and Laender, A. H. F. A Heuristic-based Hierarchical Clustering Method for Author Name Disambiguation in Digital Libraries. In *Proceedings of the Brazilian Symposium on Databases*. João Pessoa, Brazil, pp. 20–34, 2007.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering* vol. 19, pp. 1–16, 2007.

Han, H., Xu, W., Zha, H., and Giles, C. L. A hierarchical naive Bayes mixture model for name disambiguation in author citations. In *Proceedings of the ACM Symposium on Applied Computing*. Santa Fe, New Mexico, pp. 1065–1069, 2005.

Huang, J., Ertekin, S., and Giles, C. L. Efficient name disambiguation for large-scale databases. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*. Berlin, Germany, pp. 536–544, 2006.

Kalashnikov, D. and Mehrotra, S. Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph. *ACM Transactions on Database Systems* vol. 31, pp. 716–767, 2006.

Kang, I.-S., Na, S.-H., Lee, S., Jung, H., Kim, P., Sung, W.-K., and Lee, J.-H. On co-authorship for author disambiguation. *Information Processing and Management* vol. 45, pp. 84–97, 2009.

Lee, D., On, B.-W., and Kang, J. Effective and scalable solutions for mixed and split citation problems in digital libraries. In *Proceedings of the International Workshop on Information Quality in Information Systems*. Baltimore, Mariland, pp. 69–76, 2005.

Levenshtein, V. I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* vol. 10, pp. 707–710, 1966.

Levin, F. H. and Heuser, C. A. Using genetic programming to evaluate the impact of social network analysis in author name disambiguation. In *Proceedings of the Alberto Mendelzon Workshop on Foundations of Data Management*. Buenos Aires, Argentina, 2010.

Malin, B. Unsupervised name disambiguation via social network similarity. In *Proceedings of the Workshop on Link Analysis, Counterterrorism, and Security, in conjunction with the SIAM International Conference on Data Mining*. Newport Beach, California, pp. 93–102, 2005.

Manning, C., Raghavan, P., and Schutze, H. Chapter 8. In , *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England, 2008.

McCallum, A. K., Nigam, K., and Ungar, L. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston, Massachusetts, pp. 169–178, 2000.

Menezes, G. V., Ziviani, N., Laender, A. H. F., and Almeida, V. A. Geographical Analysis of Knowledge Production in Computer Science. In *Proceedings of the International World Wide Web Conferences*. Madrid, Spain, pp. 1041–1050, 2009.

Milgram, S. The small world problem. *Psychology Today* vol. 1, pp. 60–67, 1967.

Newman, M. E. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America* 98 (2): 404–409, 2001.

Newman, M. E. The structure and function of complex networks. *SIAM Review* 45 (2): 167–256, 2003.

Nin, J., Muntés-Mulero, V., Martinez-Bazan, N., and Larriba-Pey, J. On the Use of Semantic Blocking Techniques for Data Cleansing and Integration. In *International Database Engineering and Applications Symposium*. Banff, Canada, pp. 190–198, 2007.

Oliveira, J. W. A., Laender, A. H. F., and Gonçalves, M. A. RemoÃğÃčo de Ambiguidades na IdentificaÃğÃčo de Autoria de Objetos BibliogrÃąficos. In *Proceedings of the Brazilian Symposium on Databases*. Uberlândia, Brazil, pp. 206–219, 2005.

On, B.-W., Elmacioglu, E., Lee, D., Kang, J., and Pei, J. An effective approach to entity resolution problem using quasi-clique and its application to digital libraries. In *Proceedings of the ACM IEEE Joint Conference on Digital Libraries*. Chapel Hill, North Carolina, pp. 51–52, 2006.

Pereira, D. A., Ribeiro-Neto, B. A., Ziviani, N., Laender, A. H. F., Gonçalves, M. A., and Ferreira, A. A. Using web information for author name disambiguation. In *Proceedings of the ACM IEEE Joint Conference on Digital Libraries*. Austin, Texas, pp. 49–58, 2009.

Salton, G. and McGill, M. *Introduction to Modern Information Retrieval*. McGraw - Hill, 1983.

Treeratpituk, P. and Giles, C. L. Disambiguating authors in academic publications using random forests. In *Proceedings of the ACM IEEE Joint Conference on Digital Libraries*. Austin, Texas, pp. 34–48, 2009.

Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin* 1 (6): 80–83, 1945.