ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0060

Web Crawler System for Distinct Author Identification in Bibliographic Databases

Nancy Dau *Marcial Russo* *Eric Bouwsema* *Tansel Özyer* *Reda Alhajj*

Department of Computer Science
University of Calgary
Calgary, Alberta, Canada

Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey

***ABSTRACT*- A person's name is regularly used to uniquely identify himself/herself from others; unfortunately names are in no way unique and this leads to serious problems. For instance, when trying to retrieve papers from academic database repositories, it can be difficult to distinguish one author from another if the individuals in question have the exact same name. An author can also assume another name, for instance by using the full name. Thus, being able to differentiate which person a specific name is referring to can be tricky. In this paper, we propose a method to solve this ambiguity problem by gathering information from bibliographic databases and using this information to create a social network tree. Based on the relationships created among co-authors it is possible to disambiguate authors with a high-level of accuracy.**

Keywords: Namesakes, social network, name ambiguity, academic databases.

1. INTRODUCTION

Names are not unique to a single person. For instance, the most common last name in North America is Smith [1]. Every year there are new lists of top 100 baby names for boys and girls. If one would take a look at these lists, one would notice that the top names do not vary much from year to year.

Name ambiguity occurs at an early age and examples of this can be traced back to kindergarten. If a class has students with the same first name, such as John, the first initial of the student's last name may be included in order for teachers and students alike to differentiate between which John they are referring to: so John Smith would be called John S.

Similarly, if the names extend past just the given name, teachers have to become more creative. The problem with having two people with the exact same name is called a namesake. In order to overcome this issue, the teacher may say that one John Smith would just be called John and the other John Smith would be called Johnny.

This same problem becomes serious when it is transferred to a professional environment, e.g., when it occurs in academic publications that exist in databases repositories like DBLP. Researchers have the problem of namesakes because as we noted before, names are not unique identifiers. The issue with this is that unlike in a classroom where there is a restricted number of students. The ambiguity of names in the academia is a global issue where the same name may exist within the same domain of research or different research interests and within the same university or in different universities, though the latter case is more common. Unfortunately publications are cited by name and hence it is difficult to identify and separate the exact publications of a given researcher in order to avoid giving the credit where it does not belong.

Digital libraries, e.g., DBLP, IEEE, ACM, Springer, Scopus, etc are all available on the World Wide Web (WWW). It should come to no surprise then that the issue of namesakes and name ambiguity can occur on these sites. Searches can be conducted by looking up authors, but as Figure 1 shows this can lead to confusion, as these websites do not have the ability to filter or identify a single individual from authors with the same name.

In the results of a search for papers written by a common name, say Ken Barker in 2009 alone, we can see the occurrence of namesakes. In Figure 1, we see that DBLP has correctly returned papers written by Ken Barker. The highlighted papers outlined by red and green show that these publications are actually referring to two different persons with the exact same name. When clicking on the bibliographic information on these two authors, we see different information returned. This can be seen in Figure 2.

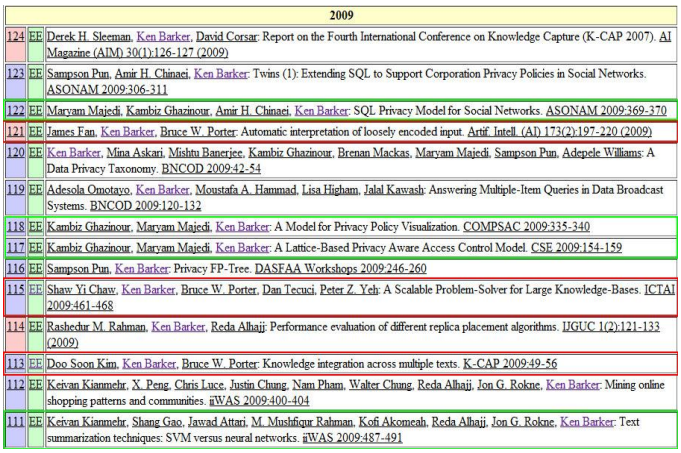
These academic websites have no way of differentiating between these two authors, so the search results of authors are not accurate. Knowing merely that the author of interest is Ken Barker; these sites will search their database and return any results that have Ken Barker as an author.

While investigating this issue of name ambiguity, we noticed a common trend: researchers throughout their career will meet peers and most of the time will mostly continue to collaborate with them, regardless whether they switch academic institutions. This relationship between authors and

doi:10.15849/icit.2015.0060

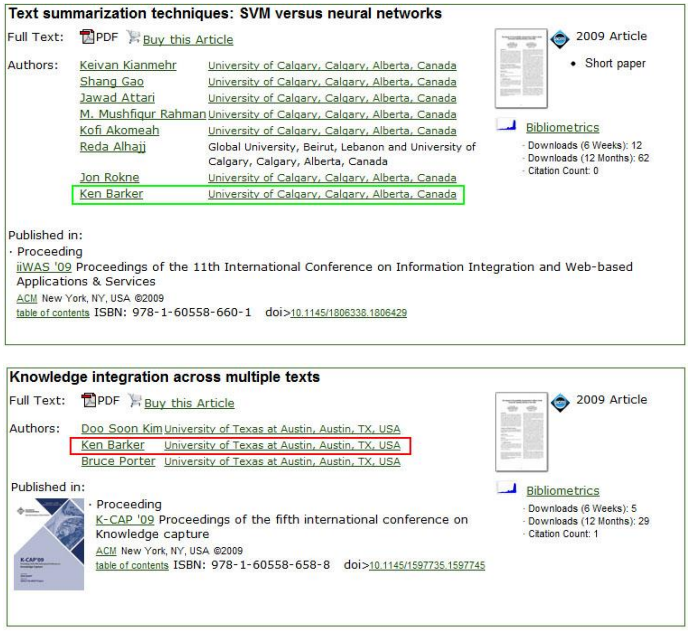
co-authors can help us distinguish one researcher from another.

In this paper, we propose a method that relies on the relationship made between authors of a paper as well as other information taken from bibliographic databases. We have created a web crawler that will go through ACM, DBLP and IEEE libraries. From these sites we collect papers of specific authors as well as information on these researchers.



2009		
124	EE	Derek H. Sleeman, Ken Barker, David Corsar: Report on the Fourth International Conference on Knowledge Capture (K-CAP 2007). AI Magazine (AIM) 30(1):126-127 (2009)
123	EE	Sampson Pun, Amir H. Chinaei, Ken Barker: Twins (1): Extending SQL to Support Corporation Privacy Policies in Social Networks. ASONAM 2009:306-311
122	EE	Maryam Majedi, Kambiz Ghazinour, Amir H. Chinaei, Ken Barker: SQL Privacy Model for Social Networks. ASONAM 2009:369-370
121	EE	James Fan, Ken Barker, Bruce W. Porter: Automatic interpretation of loosely encoded input. Artif. Intell. (AI) 173(2):197-220 (2009)
120	EE	Ken Barker, Mina Askari, Mishtu Banerjee, Kambiz Ghazinour, Brenan Mackas, Maryam Majedi, Sampson Pun, Adepele Williams: A Data Privacy Taxonomy. BNCOD 2009:42-54
119	EE	Adesola Omotayo, Ken Barker, Moustafa A. Hammad, Lisa Higham, Jalal Kawash: Answering Multiple-Item Queries in Data Broadcast Systems. BNCOD 2009:120-132
118	EE	Kambiz Ghazinour, Maryam Majedi, Ken Barker: A Model for Privacy Policy Visualization. COMPSAC 2009:335-340
117	EE	Kambiz Ghazinour, Maryam Majedi, Ken Barker: A Lattice-Based Privacy Aware Access Control Model. CSE 2009:154-159
116	EE	Sampson Pun, Ken Barker: Privacy FP-Tree. DASFAA Workshops 2009:246-260
115	EE	Shaw Yi Chaw, Ken Barker, Bruce W. Porter, Dan Tecuci, Peter Z. Yeh: A Scalable Problem-Solver for Large Knowledge-Bases. ICTAI 2009:461-468
114	EE	Rashedur M. Rahman, Ken Barker, Reda Alhajj: Performance evaluation of different replica placement algorithms. IJGUC 1(2):121-133 (2009)
113	EE	Doo Soon Kim, Ken Barker, Bruce W. Porter: Knowledge integration across multiple texts. K-CAP 2009:49-56
112	EE	Keivan Kianmehr, X. Peng, Chris Luce, Justin Chung, Nam Pham, Walter Chung, Reda Alhajj, Jon G. Rokne, Ken Barker: Mining online shopping patterns and communities. iiWAS 2009:400-404
111	EE	Keivan Kianmehr, Shang Gao, Jawad Attari, M. Mushfiqur Rahman, Kofi Akomeah, Reda Alhajj, Jon G. Rokne, Ken Barker: Text summarization techniques: SVM versus neural networks. iiWAS 2009:487-491

Figure 1. Namesakes on DBLP when searching for Ken Barker



Text summarization techniques: SVM versus neural networks

Full Text: PDF Buy this Article

Authors:
Keivan Kianmehr — University of Calgary, Calgary, Alberta, Canada
Shang Gao — University of Calgary, Calgary, Alberta, Canada
Jawad Attari — University of Calgary, Calgary, Alberta, Canada
M. Mushfiqur Rahman — University of Calgary, Calgary, Alberta, Canada
Kofi Akomeah — University of Calgary, Calgary, Alberta, Canada
Reda Alhajj — Global University, Beirut, Lebanon and University of Calgary, Calgary, Alberta, Canada
Jon Rokne — University of Calgary, Calgary, Alberta, Canada
Ken Barker — University of Calgary, Calgary, Alberta, Canada

2009 Article
• Short paper

Bibliometrics
· Downloads (6 Weeks): 12
· Downloads (12 Months): 62
· Citation Count: 0

Published in:
· Proceeding
iiWAS '09 Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services
ACM New York, NY, USA ©2009
table of contents ISBN: 978-1-60558-660-1 doi>10.1145/1806338.1806429

Knowledge integration across multiple texts

Full Text: PDF Buy this Article

Authors:
Doo Soon Kim — University of Texas at Austin, Austin, TX, USA
Ken Barker — University of Texas at Austin, Austin, TX, USA
Bruce Porter — University of Texas at Austin, Austin, TX, USA

2009 Article

Bibliometrics
· Downloads (6 Weeks): 5
· Downloads (12 Months): 29
· Citation Count: 1

Published in:
· Proceeding
K-CAP '09 Proceedings of the fifth international conference on Knowledge capture
ACM New York, NY, USA ©2009
table of contents ISBN: 978-1-60558-658-8 doi>10.1145/1597735.1597745

Figure 2. From the same search results as before we see works from two Ken Barkers, one from the University of Calgary and the other from the University of Texas.

With this information in turn, we can create a network graph/tree. The nodes represent the researchers and within the nodes we have more information related researchers, such as their academic institutions. With these nodes we will begin to create associations between authors and co-authors. This approach will slowly create a network of clusters among researchers and allow the user to see namesakes or aliases of researchers.

2. RELATED WORKS

As stated in the introduction, we noted that the idea of name ambiguity and namesakes is not a new problem. This topic has been widely investigated and there are several varieties of techniques described in the literature. Looking at the previous research done we see that the idea of duplicating records in large data files was investigated in 1983 [2]. There was research done by Hernandez et. al who gathered large commercial databases and merged data from multiple sources, this he defined as the merge/purge problem and become efficient but costly [3].

Branting has done a comparative study just on name-matching algorithms [4]. Name-matching has been studied and in a study done by Top et al. they showed just how complex name-matching can be with various different situations just based on the name and different alias' a person can have, intentional or not [5]. Not only do names differ in spelling, but researchers such as Ji et al. are interested in the way that phonetics can help with name-matching; just another way researchers are thinking outside the box to accomplish the task of matching names to the appropriate persons [6].

Recently, there are researchers who have used multi-layer clustering to try and detect name ambiguity [7]. Jiang et al. used a combination of package-merge algorithm, pattern-matching techniques and fuzzy logic rules in their research.

A study was also carried out by Wu et al. and to solve name ambiguity, they also worked with obtaining more information on the authors, such as workplace and co-author relationships. Wu and his team used this information and applied the association rule and a pre-set threshold to differentiate between name distinctions [8].

Research done in 2005 by Han used the method of K-way spectral clustering, relying on subsequent information given, such as co-authors, paper titles and publication venues [9]. With the clusters, they are able to differentiate groups and decide which groups had which members included in them. Wei et al. on the other hand, concentrated primarily on a biomedical academic website when creating an algorithm for name ambiguity [10]. By using EntrezIDs, he would match the EntrezID information with the information of authors. This allowed them to have a unique ID for each author. Even with this unique identifier and their smaller size database they gained about 75.1% precision when dealing with name ambiguity.

Shin et al. used social networks to resolve the issue of name ambiguity [11]. This research resembles ours described in

doi:10.15849/icit.2015.0060

this paper, as it focuses on the social networks created by authors and co-authors. They constructed their own namesake and name ambiguity algorithms in order to create their social networks. This research has impacted ours, but it is important to note that they have concentrated on DBLP as their main source of papers; DBLP will connect with other academic sites, but if there is no submission to DBLP then the paper will be left out. Since our research is based on the cumulative returns of the chosen academic research websites, with each having its own crawler, we can gather more information to create our network graph and hopefully also be able to gather more information if one website has a better biographical database.

There are a variety of other techniques used for identification purposes. There are a lot of creative combinations of techniques in just these few sample papers. All of these techniques are used to attempt to solve the problem of personal identification or name ambiguity. We can see that the issue of name ambiguity is a tricky one and all the papers attempted to solve this issue required many steps and more than one algorithm in order to come up with acceptable results.

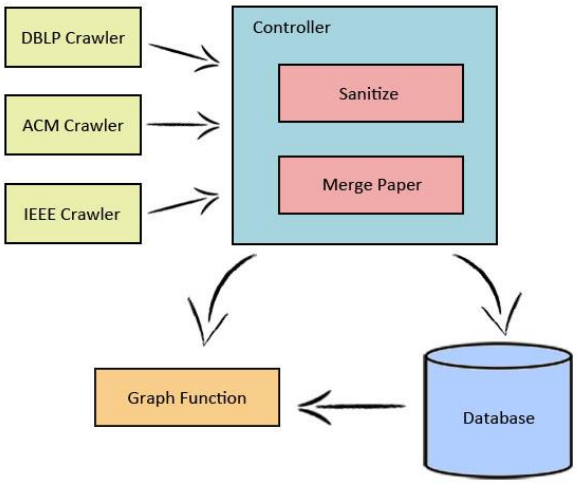


Figure 3. Proposed system architecture.

3. SYSTEM ARCHITECTURE

We need to rely on many parts of the system in order to properly handle name ambiguity. First, it is important to start off with a quick overview of all the parts.

In Figure 3, we see the existence of web crawlers. The current implementation of our web crawlers will comb through the online databases and retrieve papers and authors whose names are attached to these papers. These web crawlers will be scheduled to run or will be manually run by the user. For our current research, we will concentrate on three academic websites: namely DBLP, ACM and IEEE, though others may be easily added if needed.

From the output of the web crawlers we will be able to analyze our results with our controller. It will have two main purposes. The first is to sanitize the information to make sure it is unified and the second is to merge papers that are the same from the different crawlers.

In order to easily retrieve and update the information returned from the web crawler, we will create a database that will store the information.

The graph function that we developed will be our main algorithm to build our graph and the relationships, with which we will be able to deal with namesakes and resolve name ambiguity.

3.1 Web Crawlers

In order to efficiently keep the database up to date, we have created automated web crawlers that will search through our three academic websites. Since all three websites have different bibliographic structures we require different crawlers to return specified information from the underlying database as shown in Table 1.

Table 1. Information returned from web crawlers.

Web Crawler	List of Papers	List of Author(s)	Author Information
DBLP	Yes	Yes	No
ACM	No	No	Yes
IEEE	No	No	Yes

Since DBLP contains all of the author's papers, it will be our main source to get all authors and all the paper's information. Unfortunately DBLP doesn't have any of the author information such as Affiliation or email. This information is useful to identify authors with same names, but in this paper we will concentrate mainly on comparing an Author's co-authors to distinguish them.

With the list of papers and authors returned, the system will be able to utilize the information to help us create a network graph.

3.2 Controller

In the second step of our system architecture, we have the controller. It has two main functions: to merge papers returned from the crawlers that are the same. The crawlers will be pulling from DBLP and information from ACM/IEEE; there is possibility that researcher groups could have submitted their papers to individual academic websites. The paper could have been approved for more than one submission. If there are multiples of the same paper the controller will identify this and merge them together along with the information. The second role of the controller is to be able to sanitize the information. What is meant by sanitize, is to clean up the results so that our algorithm will

doi:10.15849/icit.2015.0060

be able to effectively analyze the information without any issues, such as issues which may arise when the results include special characters, these can be seen in languages that use accents such as French or Spanish.

3.3 Data Store

In the database we will be storing two sets of information taken from our controller.

The first will be the names of the authors. We will keep a list of names, which will be taken from the results of the DBLP crawler. The names will be added one by one into the database. In the beginning, we will have only one author; from there we will search the co-authors of a given paper. Through this iterating process, we will eventually be searching those previously added authors to see if there is a name that currently does not exist in our list. We can see an example of how this will be stored in Table 2.

Table 2. Table kept by the database with the list of authors currently seen.

Name
Ken Barker
Reda Alhajj
Bruce Porter

This will be used by both the controllers and by the graph function in order to create relationships. When we come across a new author we will add him or her to the current table of names and then in a next iteration of the author relationship table we will look for new relationships amongst them. This will create a domino effect and spread through the full database, returning to us a complete result. An example of how this is done is given in Table 3.

Table 3. Relationships between authors and co-authors

Relationship
Ken Barker – Reda Alhajj
Ken Barker – Bruce Porter

Looking at the tables, we can see how this relationship among co-authors will help us building our network graph. We see from the tables that there is a relationship between Ken Barker and Reda Alhajj, and another relationship exists between Ken Barker and Bruce Porter. Also note, that there is currently no relationship between Reda Alhajj and Bruce Porter in our database. This lack of a relationship is just as important as the relationships that exist because they will help us create an accurate network graph moving forward. Indeed the lack of relationship may be a good indicator that the two occurrences of Ken Barker are not the same person.

The second information that the database will keep track of is the list of paper titles. Once again, the system will be merging those papers that are exact duplicates of one another. Unlike other research papers, this one is based on multiple academic databases so the occurrence of duplicate papers is likely since these are separate websites. We will have to be able to properly identify and merge papers that are similar.

3.4 Graph Function

Using our knowledge of the current system, we now move on to the graph function. This part of the system is to be considered the heart of our overall architecture. The role of the graph function is to take the knowledge from the database and turn it into a network graph of clusters. From these clusters we will be able to differentiate namesakes as well as be able to eventually merge alias names.

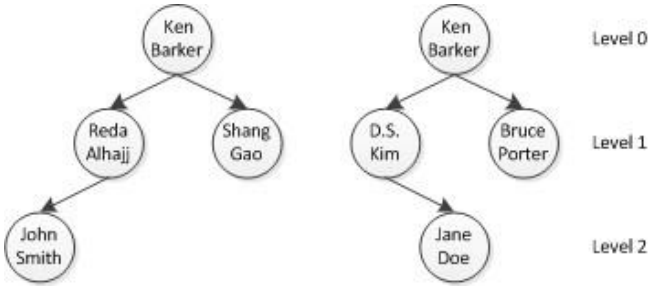


Figure 4. Visualization of the network graph

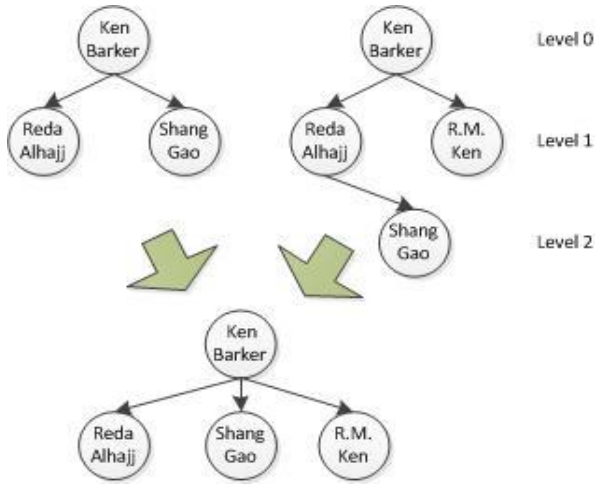


Figure 5. Merging of two clusters in the case of Namesake.

3.4.1 Namesake

We can see from Figure 4 how this graph function will be developed. A node will represent the authors and the relationships are based on co-authors. So, if a node or author has collaborated with another node, we call this a relationship and connect those two nodes together. This will basically become a forest of trees. In Figure 4, we see that

doi:10.15849/icit.2015.0060

two Ken Barkers exist. They both have the exact same name, but their relationships are distinct from each other. This will be the easiest case to confirm that these authors are indeed separate authors and that there is no need to merge them.

If for example in Figure 5 there is a case where two Ken Barkers exist with the same name and the co-authors are relatively similar, we will have to look at what degree of similarity these two clusters have. This will depend on the co-authors and the levels that these similarities occur on. If the similarities threshold is met, we will combine the two clusters.

3.4.2 Six Degrees of Separation

Another case study that needs to be explored is the idea of having a namesake within a namesake. In our tree, we note levels on the right. The deeper we move downward into the tree, the less value we give to that relationship. This rule is to prevent a phenomenon known as the Six Degrees of Separation [12] (or in pop-culture, the Seven Degrees of Kevin Bacon). This idea is well known: if one picks a person in the world, usually a celebrity, it will take six people or less in order to "know" this person. This will create a lot of "friend of a friend" instances, but in the end one will somehow be socially related to that chosen person.

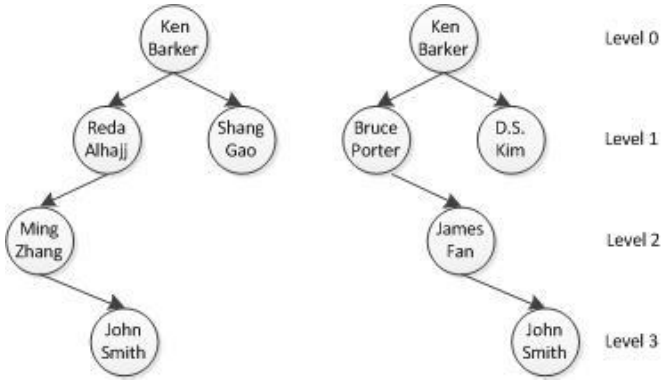


Figure 6. Theory of degrees of separation applied.

Unlike the idea of Six degrees of separation, we do not have a database that includes all of society and because of this we must put a limit on how much of a degree of separation we can allow before our analysis will no longer be optimal. In Figure 6, we added another level of our network graph. Note that on level 3 we have two nodes in separate clusters with the name John Smith. Looking at the two clusters, we see there is noticeable difference in the other co-authors that Ken Barker has worked with. So, even though Ken Barker has associations with John Smith, this will have little effect since it occurs lower on the tree and the co-authors before him do not meet the threshold of similarities. We want to restrain our tree from adapting to the idea of six degrees of separation, so we have reduced the number into half; we will look at 3 degrees and put less weight on each level.

3.4.3 Multiple Names (Alias)

Another example to explore is the idea of an author using more than one name when publishing papers. Examples of this can be seen when a person uses his or her middle name to avoid the issue of namesakes. Or if the individual moves to another institution and the spelling of the name uses special characters. When first iterating through the tree, we may have two separate nodes for one author.

In the case of two nodes that actually represent one author occurring as shown Figure 7, the system will have to be able to first identify and then merge the clusters. If the two clusters have a high enough agreeability rate, the root of the smaller cluster will have to be sanitized to conform to the larger cluster, which we will be merging with. This sanitization process is the only difference between a regular merge versus this specific type of merge. The clusters will be combined by integrating them together. In order to keep the databases as similar as possible, we will also have to reassign the papers that were previously assigned under the old author's name to the new author's name. This creates a cycle between the controller, the database and the graph algorithm. We can see the example of a merge occurring presented visually in Figure 7.

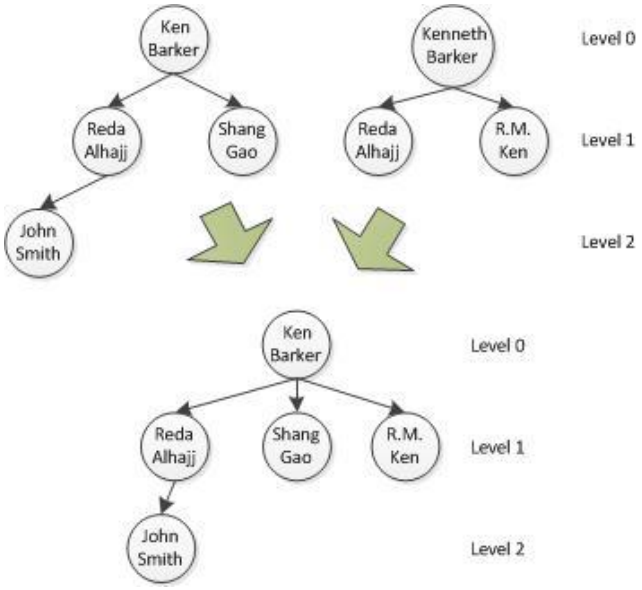


Figure 7. Merging of two separate clusters when authors use alias.

With the current system this situation is not handled. The issue with our current implementation is that if an author were to move to another academic institution, and no longer write with their former peers, the system then has no way of connecting the two entities. A solution to this issue is to be able to go through the paper using a PDF reader.

doi:10.15849/icit.2015.0060

Unfortunately, the implementation for the PDF reader and the current system has not yet been incorporated in this running system. This is discussed more in future work.

4. EXPERIMENTAL RESULTS

4.1 Setup

The results are highly dependent on the system's ability to obtain all of the available papers from the specified author, using the three implemented web crawlers. The system must also have the ability to distinguish different authors with the same name. DBLP has standardized their data, thus obtaining authors and their papers do not require a check to be made for name abbreviation, or modifications; hence all records can be obtained by using our DBLP crawler to search the database.

The first step in the system is to declare a set of authors to act as our seeds. The crawlers will then be used by using the previously declared set to perform a DBLP search. Since DBLP returns a list of papers with their authors, we break them into two objects. The paper: this object consists of the paper's title, year, authors, topic and where it can be found, the hyperlink; this is shown in Figure 8.

The second object that is created is the author: this consists of the authors name, affiliations, email, coauthors, and papers that he or she has participated in writing; this is shown in Figure 8.

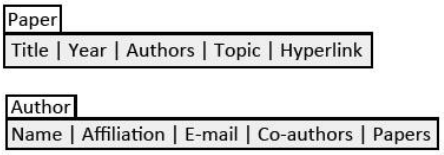


Figure 8. Representation of the two objects created.

Once all the available information of an author has been collected, the program enters a cyclic stage, the co-authors now become seeds and the cycle starts again. Eventually when all seeds have being added to the main set, we have finished collecting information and can now move to building our network.

The program then moves on to the next stage: Identifying authors with the same name and merging them into one, as shown in Figure 5. If the situation occurs that we have a namesake within a namesake, the system will have to perform the most complicated step of the identification algorithm by using up to three levels of our created network to correctly merge authors and networks, as shown in Figure 6. If two existing networks are separated and no affiliation is made between them up until three levels, we do not merge. If there does exist a co-author within three levels we will assume this is the same person and merge.

Table 4. Results for seed Reda Alhajj

	Distinct Papers	Distinct Co-Authors
Initial Results	291	698
Merge based on Names & Papers	24	197
Path Merge	13	197

4.2 Results & Analysis

Our system was able to successfully reduce the number of potential distinct authors for two test names (Reda Alhajj and Ken Barker) drastically. In the case of Reda Alhajj we have a single author who is prolific and has numerous co-authors.

From our initial results returned from DBLP we were able to group first into 24 distinct authors, and then through path operations we were able to reduce down to 13 distinct authors. This drastically reduces the number of potential candidates to match using additional heuristics. (Table 4.)

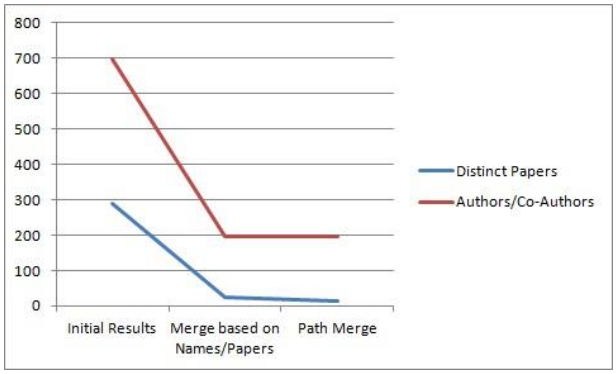


Figure 8. Results for seed Reda Alhajj

Table 5. Results for seed Ken Barker

	Distinct Papers	Distinct Co-Authors
Initial Results	138	137
Merge based on Names & Papers	23	135
Path Merge	22	135

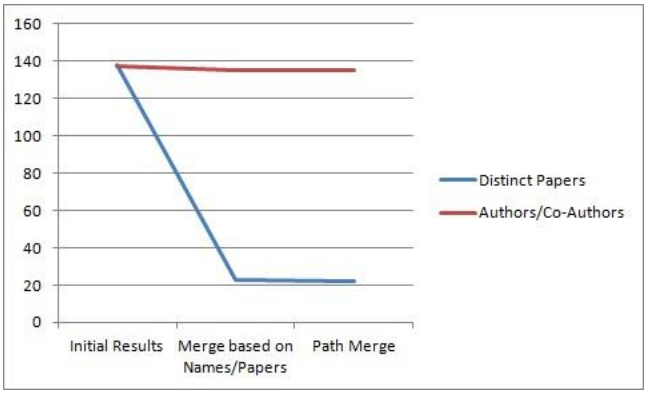


Figure 9. Results for seed Ken Barker

doi:10.15849/icit.2015.0060

Our second author, Ken Barker, consisted of two distinct authors who have similar names. Our initial results returned 138 potential authors and we were able to reduce it to 23 potential candidates. The path procedure did not provide drastically improved results, only reducing the number of potential candidates to 22. (See Table 5.)

When examining the results for Ken Barker we can see that there are a number of distinct clusters who consist of most of the collaborative work done by both individual authors. For Ken Barker currently at the University of Calgary you can see a number of past and current students and staff who are the co-authors. For Ken Barker from University of Texas this also holds.

Table 6. Clusters for authors 'Ken Barker'

Ken Barker **Univ. of Calgary/Univ. of Manitoba/Univ. of Alberta**	**Co-Authors**	**Papers**
Brenan Mackas, Jawad Attari, Philip W. L. Fong, Kofi Akomeah, Angela Cristina Duta, Walter Chung, M. Sheelagh T. Carpendale, Justin Chung, Nelson C. N. Chu, Chenen Liang, M. Mushfiqur Rahman, Rosa Karimi Adl, George Shi, X. Peng, Adepele Williams, Christoph W. Sensen, Chunyan Wang, Janaki Gopalan, Jamal Jida, Leanne Wu, Nancy Situ, Maryam Majedi, Kambiz Ghazino, Reda Alhajj….	54	79
Moustafa A. Hammad, Jalal Kawash, Adesola Omotayo, Lisa Higham	4	4
Joseph Osuji, Faith-Michael E. Uzoka, Okure U. Obot	3	1
Dina Said, Peter Federolf, Lisa Stirling	3	1
Randal J. Peters, Coimbatore Rajagopal Saravanan	2	2
Peter C. J. Graham	1	1
Ahmad R. Hadaegh	1	2
John Aycock	1	1
C. I. Ezeife	1	2
Wendy Osborn	1	2
M. Tamer Özsu	1	2
Ramon Lawrence	1	3
Subhrajyoti Bhar	1	2
Amin Y. Noaman	1	1
Sergio Camorlinga	1	2
Sylvanus A. Ehikioya	1	3
Md. Moniruzzaman	1	1
Michael Zapp	1	1

Ken Barker **Univ. of Texas/Univ. of Ottawa**	**Co-Authors**	**Papers**
Pedro Romero, Mark Greaves, Daniel Hansch, Rutu Mulkar-Mehta, Michael Eriksen, Andrés Rodríguez, David Gunning, Bhalchandra Agashe, Blake Shepard, Michael Glass, Moritz Weiten, David D. McDonald, Nancy Salay, Gavin Matthews, Jing Tien, Bonnie E. John, Benjamin N. Grosof, Paul G. Allen, Eduard H. Hovy, Sourabh Patwardhan, Jérôme Thoméré, Doo Soon Kim…	50	21
Sylvain Delisle, Terry Copeck, Stan Szpakowicz	3	4
David Corsar, Derek H. Sleeman	2	2
Nadia Cornacchia	1	1

Interestingly enough for both Ken Barkers the system does not care that the author has moved universities, but rather groups based on collaborative efforts, where colleagues continue to work together even after moving to new institutions. (See Table 6.)

An issue we have noticed with this system though is when two authors write a paper, but one or the other of the authors does not collaborate with any other authors at the time of crawling the database. As you can see with Ken Barker (Univ. of Texas) and Nadia Comacchia, with only one paper and only one co-author (Nadia Comacchia only has one paper on DBLP) it is hard for us to cluster this author, and thus requires us to use other methods to determine which Ken Barker this is. For the purpose of Table 6 we were able to determine the proper author based on CV's and not using our system.

While we are not able to determine with 100% accuracy the clustering of each author, we have shown that we can drastically reduce the number of potential unique authors using our system. Building upon this work we should be able to determine with a high level of accuracy each distinct author.

5. FUTURE WORK

A current issue that can be resolved with future work is that bibliographic information is not as consistent as it is needed to be among the websites. Since websites such as DBLP, IEEE and ACM may run independently of each other, the bibliographic information provided range from good bibliographic information, to basically no bibliographic information; this makes it more difficult to pull from websites as they are not all consistent with each other.

For future works what can be done with the current system to gain better accuracy would be to use a PDF reader to gain access to information such as email or the educational institution. This will allow future research to more accurately connect the authors with this information and thus return stronger results than what we've been able to do thus far. With ACM and other academic research websites currently enforcing stronger layouts of their submitted papers this means that such problems as namesakes can be better dealt with. For example, just as this paper, many other papers are forced to now include author information such as e-mail and the institution which the researchers are writing for. E-mail is a unique identifier as no two people can ever have the same e-mail under the same domain name. Furthermore a researcher is not allowed to be working for two or more academic locations at one time, from this we can safely say that this is also a unique identifier, as published works of an author can only be from that single institution. This would help us be able to deal with authors who may write under several different names. Also for better

doi:10.15849/icit.2015.0060

accuracy to distinguish authors a restriction to comparing common names could be placed. For example if both authors share a paper with "John Smith" since this is such a common name it should not be used to identify the uniqueness of the author.

The current system also runs a lot slower than originally anticipated. A suggestion for future work is to incorporate something such as Hadoop [13]. This open-source framework will allow the system to be reliable and scalable. Since it is utilizing distributed systems this will help speed up the system and will be able to get faster results.

6. CONCLUSION

In conclusion, we proposed a solution to solve the problem of name ambiguity. Through research we suggested using a method that relied on networking. Our data was taken from web crawlers that searched through academic websites, and extracting information specifically from the bibliographic pages that they supplied.

We considered such things as namesakes and proposed a theory for how to deal with authors who may use multiple names when publishing papers. Our main goal was to be able to distinguish papers by researchers who publish at the same time as someone else who has the same name. Our proposed system is able to make a network from co-authors providing us associations that we can use to properly distinguish these authors who have the issue of namesakes.

We have also provided future work that can be done, in order to improve our current system, this can be found in the future works section of our paper.

REFERENCES

[1] Wikipedia List of most common surnames in North America. http://en.wikipedia.org/wiki/List_of_most_common_surnames_in_North_America.

[2] Bitton D. and DeWitt, D., Duplicate Record Elimination in Large Data Files, ACM Transactions on Database Systems, pp. 255-265, 1983.

[3] Hernandez, M. and Stolfo, S. The merge/purge problem for large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 127-138, 1995.

[4] Branting, L.K., A comparative evaluation of name-matching algorithms. ICAIL '03 Proceedings of the 9th international conference on Artificial intelligence and law.

[5] Top, P., Dowla, F. and Gansemer, J. A Dynamic Programming Algorithm for Name Matching. Computational Intelligence and Data Mining, 2007. CIDM 2007. Page(s): 547-551.

[6] Ji, H., Grishman, R. and Wang, W. Phonetic name matching for cross-lingual Spoken Sentence Retrieval. Spoken Language Technology Workshop, 2008. SLT 2008. IEEE. Digital Object Identifier: 10.1109/SLT.2008.4777895. Publication Year: 2008, Page(s): 281-284.

[7] Jiang, W., Wang, A, Wu, C., Chen, J. and Yan J. Approach for Name Ambiguity Problem Using a Multiple-Layer Clustering. Computational Science and Engineering, 2009. CSE '09. Volume: 4, Page(s): 874-878.

[8] Wu, B., Cai, W. and Li, Y. Association analysis and case study framework based on the name distinction. Computer Application and System Modeling, 2010. Volume: 4, Pages V4-285 – V4-289.

[9] Han, H., Zha, H., and Giles, C.L. Name disambiguation in author citations using a K-way spectral clustering method. Digital Libraries. JDCL '05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Object Identifier. pp.334-343, 2005.

[10] Wei, C., Huang, I., Hsu, Y. and Kao, H. Normalizing Biomedical Name Entities by Similarity Based Inference Network and De-ambiguity Mining. Bioinformatics and Bioengineering. BIBE '09. Ninth IEEE International Conference on Digital Object Identifier. pp.461-466, 2009

[11] Shin, D., Kim, T., Jung, H. and Choi, J. Automatic Method for Author Name Disambiguation Using Social Networks. Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on Digital Object Identifier. Page(s): 1263-1270.

[12] The Guardian: Proof! Just six degrees of separation between us. http://www.guardian.co.uk/technology/2008/aug/03/internet.email.

[13] Welcome to Apache Hadoop! http://hadoop.apache.org/.

[14] Ferreira A. A., Gonçalves M. A. and Laender A. H. F., A Brief Survey of Automatic Methods for Author Name Disambiguation, SIGMOD Record, Vol. 41, No. 2, June 2012.