Mining Information for Instance Unification

Niraj Aswani, Kalina Bontcheva, and Hamish Cunningham**

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
niraj,kalina,hamish@dcs.shef.ac.uk

Abstract. Instance unification determines whether two instances in an ontology refer to the same object in the real world. More specifically, this paper addresses the instance unification problem for person names. The approach combines the use of citation information (i.e., abstract, initials, titles and co-authorship information) with web mining, in order to gather additional evidence for the instance unification algorithm. The method is evaluated on two datasets – one from the BT digital library and one used in previous work on name disambiguation. The results show that the information mined from the web contributes substantially towards the successful handling of highly ambiguous cases which lowered the performance of previous methods.

1 Introduction

Many Semantic Web (SW) and knowledge management applications need to populate their ontologies[1] from structured, semi-structured, or unstructured data sources. Frequently the same name (e.g., a person or a company name) would appear in more than one source (e.g. database records) and the system then needs to decide whether these names refer to the same real-world object or not. This problem is known as *instance unification* [2], i.e., given two instances in an ontology one needs to determine whether or not they refer to the same object. A typical example in applications such as Google scholar is the need to determine whether the authors "N.J. Davies" and "J. Davies" of two different papers are actually the same person. Or even, whether there are two different individuals both called J. Davies and therefore it is wrong to assume that two papers whose author is "J. Davies" are authored by the same person.

In this paper we address the instance unification problem for person names. The work is carried out in the context of the British Telecom digital library, as part of the SEKT project[2], which aims to build the next generation of knowledge management technology. The digital library consists of metadata about

** This work is partially supported by the EU-funded SEKT project (http://www.sekt-project.com)

[1] For the purposes of this paper an ontology is defined as the datamodel that describes classes (a.k.a. concepts), instances (a.k.a. individuals), attributes (a.k.a. properties) and relations (i.e. ways that objects can be related to one another).

[2] For further details see http://www.sekt-project.com

papers, including paper authors (initials and surname), title, place and date of publication, abstract, and, optionally, author affiliation. Some of the records also provide a link to the full text of the paper, however, we decided to not use it in the current experiment as only 30% of all papers have full text available. In addition, we wanted to develop a method that can work using only information from the ontology, without access to the original data sources.

Due to name variations, identical names and spelling mistakes, disambiguating person names is difficult. Researchers have been exploring various ways to address this problem. Perhaps the closest in spirit is work on Ontocopi [1] and name disambiguation in author citations [8]. Ontocopi exploits relations in the ontology in order to calculate the similarity between two instances, based on the overlap between their properties. The overlap is calculated based on string similarity and the approach was deployed in the context of disambiguating authors and project members. Similarly, the work on name disambiguation in author citations [8] exploits overlap in the co-authors, paper titles, and place of publication. The main shortcoming of these approaches is that they have difficulty distinguishing between authors with the same name, who work in the same area, and where the number of citations is not sufficient to build a good co-authorship model as is the case with our data.

This paper presents a fully automatic web-based approach for instance unification in ontologies containing publications, titles, authors, abstracts, etc., where different instances of these are created from bibliography records. In other words, the ontology population algorithm has assumed that all authors of all publications are different and a corresponding instance is created in the ontology for each of them. Then the instance unification task addressed here is to determine how many authors are there in the real world and insert the required "sameIndividualAs" statements in the ontology.

The approach is evaluated on two datasets – one from the BT digital library and one used in previous work on name disambiguation. The results show that the information mined from the web contributes substantially towards the successful handling of highly ambiguous cases which lowered the performance of previous methods.

A major part of the work focused on identifying which features lead to the best performance on the author disambiguation task and, consequently, these features are specific to this problem. Nevertheless, the algorithms discussed here (normalising names, identifying an author's publication page, identifying an author's full name) and the evaluation methodology can be applied to the more generic problem of instance unification.

The paper is structured as follows. Section 2 discussed related work and identifies outstanding problems. Next Section 3 presents the ontology used in these experiments. The web-based instance disambiguation algorithm is presented in Section 4. Several issues, such as normalising names, identifying author's publication page, identifying author's full name, calculating similarities based on the collected features and making the overall decision are discussed in this section.

Evaluation results are discussed in Section 5. The paper concludes by outlining future work.

2 Related Work

The author disambiguation problem bears similarities to citation matching, which typically applies machine learning in order to identify whether two citations actually refer to the same publication, by using string similarity and frequency-based features (e.g., [10]). However, citation matching is different from the problem of resolving person name ambiguities, because it is only concerned with paper references and does not disambiguate the authors in them.

The research most relevant to our is on name disambiguation. A survey carried out in the United States showed that names can be very ambiguous as over 90,000 names are being shared by 100 million people in the United States alone [6]. However, name disambiguation is particularly difficult when there is limited contextual data. Such problem arises in the domain of citations, or in bibliographies, where no additional information other than the citation itself is available. Various approaches have been tried, some directly linked to the problem of disambiguating authors in citations (e.g., [6], [8]) and others to disambiguation of person names (e.g., [9]).

One such recent approach for author name disambiguation uses a K-means clustering algorithm based on an extensible Naive Bayes probability model [7]. The algorithm is based on three features collected from citations: co-author names, the title of the paper and the title of the journal or proceedings. The work is based on the assumption that a researcher usually has research areas that are stable over a period and tends to co-author papers with a particular group of people during that period. The disambiguation system, given an author name, clusters the citations of different similar named entities. However, their method uses manually collected publications pages, where the correct publication pages are identified manually among the results returned by Google with a query consisting of the author name and "publication" as a keyword.

The approach is evaluated on two names "J Anderson"(6) and "J Smith"(9) with accuracy of 70.6% and 73.6% respectively. The work was improved further by using information about aliases and name invariants from a database [8]. Co-author names were identified as the most robust attribute for name disambiguation. They also show that using journal titles gives better performance than using words from the paper title. The reported results are more than 90% accurate in disambiguating the two names "J Anderson" and "J Smith". This paper demonstrates how these results can be improved further by mining information from the web.

Another method [6] is semi-automatic and uses user feedback where people are asked to provide some contextual information to help identify the author unambiguously. Examples include *Location*, *Contact* such as email or phone, *Organization*, *Relation to other person(s)*, etc. While the goal of their work is different from ours, they use co-occurrence of the given person name and the contextual information as disambiguation evidence, which bears similarities to the way we identify the person's full name (see Section 4.1).

Fietelson [5] discusses disambiguating first names using lexical means. In his approach, elements of a name, the first name and the last name, are identified using self-citations among other features. Afterwards, the names are normalised into lower-case and foreign accents and special characters are replaced. In our approach we employ part of the described technique in order to normalise author names. In addition, [5] demonstrated that full names lead to better results than initials and surname information. Consequently, given an abbreviated name of an author, we first search the web and try to identify their full name.

As our approach mines the web for people's publication pages as part of the instance unification process, therefore work on finding such pages is also relevant. Perhaps the most similar in spirit is the Armadillo system [3], which discovers who works for a given department and their home pages. The system identifies automatically person names and checks them against DBLP, then relies on HomePageSearch[3] to identify the author's home page. Alternatively, the given department web site is searched for the home page. However, this approach is not applicable in our case for two reasons. Firstly, Armadillo assumes that the homepage is located within a specified website, whereas in the general case (e.g., a digital library) the system does not have such information. Secondly, the algorithm for checking the person name is dependent on the existence of an external domain-specific resource, which means that the system needs to be tailored specifically for each domain.

In the SW context, instance unification in ontologies is important for interoperability among ontologies and for cross ontology reasoning. Two general means of detecting whether two instances refer to the same real-world object have been identified [2]. One of them is the exact case, where the instances are unifiable and the another one is the probabilistic case, where each pair of two instances is assigned some probability (between 0 and 1). A threshold is used to decide if the instances are same. The aim of our work is precisely to identify the features which are important for the instance disambiguation task. Therefore, we experiment with various combinations of features and collect probability measures for each of these combinations of features. Having obtained these measures, one can use machine learning methods to learn a threshold and unify or disambiguate instances automatically.

In the section below we describe our work on instance disambiguation and present different experiments.

3 The Ontology and the Author Instance Disambiguation Problem

The ontology used in these experiments is Proton[4], a basic upper-level ontology developed in the SEKT project which contains about 300 classes and 100 prop-

[3] http://hpsearch.uni-trier.de/

[4] http://proton.semanticweb.org/

ID	Author Name	Co-authors	Publication Title
1	Davies, J	Merali, Y	Knowledge capture and utilization in virtual communities
2	Davies, J	Chaomei, C	Integrating spatial, semantic, and social structures for knowledge management
3	Davies, B.J.	Shuliang Li	Key issues in using information systems for strategic marketing decisions
4	Davies, N.J.	Krohn, U Weeks, R.	Concept lattices for knowledge management
5	Davies, J	Mabin, V.J.	Knowledge management and the framing of information: a contribution to OR/MS practice and pedagogy
6	Davies, N. J.	Crossley, M. McGrath, A.J. Rejman-Green, M.A.Z.	The knowledge garden

Table 1. An example dataset for the name "J. Davies"

erties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval of documents.

The metadata from the digital library is automatically inserted as instances in the ontology. The total number of papers in the library is 5 million and our test set contains 4429 instances of papers in the area of knowledge management with 9065 author names.

Table 1 shows an example dataset for the author "J. Davies" giving information on his publications (author name, co-author names, and publication titles)[5].

As discussed in Section 2, previous work has used a number of features to disambiguate author names: compatibility between initials and first names, overlap in paper titles, co-authorship, the name of conference or journal where the paper is published, etc. The disambiguation problem is made harder on our dataset, as the papers were chosen from within the same field (knowledge management), where different authors would publish at the same set of conferences and journals and have similar words in the paper titles. In addition, the data only provides the surname and initials of the authors. In case of "B.J. Davies" and "N.J. Davies", where the first name initials are also available, one can easily distinguish them by simply referring to their names. On the other hand, it is difficult to identify whether the first "J. Davies" is same as any other "Davies" in the table. There is a very little overlap in the names of co-authors of different "J. Davies" (Table 1). Similar to "J. Davies", we could not find any overlap in the names of co-authors of "Smith" (21 instances).

Consequently, it is difficult to disambiguate author names by computing similarities only on the basis of the citation details. However, the information available on the web can be exploited to perform instance disambiguation. An approach specifically tailored to mining computer science department web sites

[5] Abstract details are excluded from the table due to space limitations.

was discussed earlier in Section 2. In the following section, we describe a more general method for web-based instance disambiguation.

4 Web-Assisted Instance Disambiguation

Given the ontology and a surname, the first step is to retrieve all publications authored by authors with the given surname. For each citation information such as co-authors, title of the paper and abstract is collected.

After collecting all citations of authors with the given surname, the task is to exploit these features and identify which author names refer to the same real persons and how many real persons have authored each of the papers in our dataset. Below we describe an application, which, step-by-step, carries out various operations to disambiguate instances of different authors in the ontology.

It is assumed that each author with the same surname has a different instance ID and therefore the task is to identify which two IDs (i.e., instances) refer to the same author. For each pair of author IDs we calculate a number of similarity measures based on features such as the following:

- whether the authors have the same full names as identified from the web (Section 4.1)
- whether the authors share the same publication page (Section 4.2)
- title similarity (Section 4.3)
- abstract similarity (Section 4.3)
- name initials similarity (Section 4.3)
- co-author similarity (Section 4.4)

Based on the collected individual similarity measures, the overall similarity is calculated for each author pair and a binary equivalence decision is made. Next we explain the method of calculating similarity for each of the features.

4.1 Finding Authors' Full Names

As explained in [5], people write their names in different forms, so as a first step we try to calculate the similarity in authors' names. In our case, however most of the names in citations remain ambiguous due to the use of initials or incomplete names. For example "D. Jones" can refer to either "David Jones" or "Daniel Jones" or maybe to some other author whose first name starts with "D". Consequently if the authors' full names are discovered, then the ambiguity problem can be reduced substantially.

Therefore we implemented a method which from a surname and a publication tries to retrieve the author's full name from the web—based on the assumption that a web page may exist that contains the author's full name and the given publication. The method first tries to locate such a page and, if successful, verifies that the name is indeed a full name according to the following orthographig constraints[6].

[6] The algorithm assumes that the first and the middle names are one token each.

1. If the name consists of two words:
   (a) the first letters of both words must be in uppercase
   (b) if one of the words is identical to the surname, and if the length of the other word is two characters, they must not be in upper case. If they are, they are considered to be the initials of the first and second names.
2. If the name consists of three words:
   (a) the first letters of all three words must be in upper case
   (b) if the first word is identical to the surname, the second word must contain at least two letters. In this case the last word is considered to be the middle name and can have a single upper case initial.
   (c) if the last word is identical to the surname, the first word must contain at least two letters. In this case, the middle word is considered to be the middle name.

The top five pages that contain the author surname and the publication are considered as candidates for retrieval of the full name. Using the above heuristics, names are retrieved from each of these pages and the distance between the full name and the publication in terms of number of characters is calculated. The name that is nearest to the publication title is deemed to be the full name of the author under consideration. Having obtained as many full names as possible, for each pair of author IDs we calculate a full-name similarity matrix as follows: a value of 1 is given to authors having identical full names and 0 otherwise (including cases in which full names were not found for either or both of the authors).

4.2 Identifying Authors' Publication Pages

For each pair of author IDs and their associated publications, Google or Yahoo is queried in an attempt to find a page that contains the author surname and the titles of the two publications. This search is based on the assumption that if the author IDs refer to the same real person, the relevant papers will most likely appear together on his publication page.

Digital libraries such as ACM and CiteSeer are the most likely and obvious source of bibliographies. Since they use various approaches to index citations (e.g. conservative or normalizing names), when queried, they are the most likely hits. As a result, they show the entire bibliography page that contains both the titles and the surname specified. Since such bibliography pages are the results of pure text search, they do not help in disambiguating names but add more complexity to the problem, so such digital libraries are excluded from this search. The Google query is prepared with the following elements:

- The keyword "publication" or "papers"
- Author Surname
- Title of the publication of the first author
- Title of the publication of the second author
- `-site:`<sitesToExclude> digital libraries such as acm.org, sigmond.ord, ist.psu.edu and informatik.uni-trier.de

The query is then sent to a search engine. An empty result set is interpreted as an indication *against* considering the two author IDs as references to the same person. However, the final decision on whether these IDs should be unified is not based on this criterion alone, as there can be other explanations for the lack of matching pages. (For instance, the author's publications page may not be up to date or he may not have one.)

Although we exclude some digital libraries from the engine query, this does not guarantee that the results will not contain any bibliography pages, e.g., a bibliography of knowledge management publications. These need to be filtered out as they are not single-person publication pages (and therefore not evidence that the two papers were written by the same person).

After a careful analysis of several bibliography web pages, we developed a filtering module that removes a whole web page from the search results if it contains the word "bibliography" in any of the following contexts:

- title
- headers (i.e. h1, h2, h3, h4, h5 and h6)
- **boldface** tag
- *italic* tag
- head
- meta
- centered

The top five pages in the result set after filtering out bibliographies are processed further in order to identify the author's publication page (assuming that indeed both publications have been authored by the same person).

The formulation of the query means that all matched pages will contain the publication titles and the author's surname and, if it is indeed a publication page, the author's name would appear in it with a higher frequency than any other person name. Therefore each page is processed with the ANNIE named entity recognition system [4] in order to identify sentence boundaries and locate person names.

The final step is to determine which of several returned pages is actually the given author's publication page. Analysis of the matching pages showed that some would be the author's publication page but others would be more complex (e.g., CVs). The contents of such complex web pages tend to be divided into several sections, such as personal interests, work history, names of supervised students, recommended readings, publications, etc. Consequently, straightforward counting of the frequency of author names cannot reliably distinguish the publication page from other pages. Instead, the algorithm assigns the highest score to the page which contains the highest percentage of author names and references over its total length.

Another assumption is that it is likely to find more of a given author's publications on his own publication page than on any other webpage and therefore the page that contains, for example, 5 publications by that author out of 10 references in total is deemed less relevant than the page that contains 10 publications by the given author out of 20 or 25 references. In other words, preference is given to the page that contains the most publications by the given author.

Each pair of author IDs for which a page is successfully identified is given the score 1 to indicate a possible match. When no page is located, the score 0 is assigned instead. It is possible that the search engine does not respond to some queries and in such case the score of -1 is given to indicate that the results should not be taken into account. The identified page is re-used later to find other titles of other authors under consideration. If the match is located for any other author name, the author name is considered to be the same as the other two names for which originally the page was identified.

4.3 Use of Titles, Abstracts and Initials

Before computing overlap in titles and abstracts, stop words such as articles and prepositions are removed and the remaining content words (e.g. nouns, proper names, adjectives and verbs) are stemmed so that their lemmas can be compared. Word order is not important for comparing titles and abstracts, but it plays a very important role when comparing initials and surnames. For example, given a pair of author IDs and titles (or abstracts), the similarity measure is calculated as follows:

$$S_{(e_1,e_2)} = \frac{2n}{L_1 + L_2} \tag{1}$$

where

$$\begin{aligned}
S &= \text{similarity} \\
e_1 &= \text{instanceidofthefirstauthor} \\
e_2 &= \text{instanceidofthesecondauthor} \\
n &= \text{numberofidenticaltokensinthetitle(orabstract)featureof}e_1\text{and}e_2 \\
L_1 &= \text{totalnumberoftokensinthetitle(orabstract)featureof}e_1 \\
L_2 &= \text{totalnumberoftokensinthetitle(orabstract)featureof}e_2
\end{aligned}$$

The same formula is used for titles and abstracts. When there are co-authors, the number of identical co-authors is taken into account. As pointed out before, the order of tokens is very important when comparing initials of two authors: for example the initials “N.D.” would mean different from the initials “D.N.”. Similarly, the initials “N.D.” can have some similarity with the initial “N.” but not with the initial “D.”. In the former case, it is possible that the first name of both authors is same and hence the initials. One can not exclude a possibility of people using their middle name as first name, but considering it as a first initial is more likely to introduce more errors so this comparison is not used in our algorithm.

4.4 Co-authorship Information

In the case of co-authorship information, the overlap among the co-authors of each pair of publications is calculated. Consider Table 2, which presents co-authorship information for various instances referring to the **same** author.

In this case, co-authors of each instance are compared with co-authors of other instances. The third column shows the similarity figures. In this case,

ID	Author Name	Co-authors	similarities
1	Y. Wilks	N. Webb, H. Hardy, M. Ursu, T. Strzalkowski	id:2=0.33, id:3=0, id:4=0
2	Y. Wilks	N. Webb, M. Hepple	id:1=0.33, id:3=0, id:4=0
3	Y. Wilks	N. Ide	id:1=0, id:2=0, id:4 =0
4	Y. Wilks	-	id:1=0, id:2=0, id:3=0

Table 2. Co-authorship information for the name "Y. Wilks"

the first two instances do share at least one co-author but none of the rest have any common co-authors. The results show some probability for the first two instances referring to the same author, but it will be unfair to comment anything for the third and the fourth instances. If the instances are identified as referring to different authors, just because they do not share any co-author, the disambiguation would be incorrect—at least for the given example where all instances do refer to the same author. The same is true for the earlier example of "J. Davies" (see Table 1), where actually the first, second, fourth and the sixth instances in the table are referring to the same author and none of them share any co-author. Thus, in our dataset, the co-authorship does not give us much evidence in some cases.

5 Overall Similarity and Results

After independently obtaining similarity measures for the various features, the overall similarity needs to be calculated for each pair of author IDs. Because the features vary in importance, each feature is assigned a weight and the overall similarity for a given pair of author IDs (e_1 and e_2) is computed as the sum of each individual similarity measure multiplied by its weight. Equation 2 is used for obtaining the overall similarity for the given pair of author IDs (e_1 and e_2). Finally, we specify a minimum similarity threshold for for a pair of author IDs to be deemed torefer to the same author.

Table 3 shows the name disambiguation results for the author "J. Davies". The instance pairs in **bold** refer to the same person and consequently the instance unification algorithm should consider them the same. The overall similarity measures in **bold** indicate a correct result, whereas those in *italics* indicate an incorrect result. The first six columns show the individual similarity measures for the features (shared publication page, identical full name, etc.). Columns C1 to C6 then show the overall similarity measure for the given pair of IDs, when a given set of features is taken into account. C1 corresponds to only using titles, initials, and abstracts for disambiguation; whereas C2 uses the co-authorship information as well. Therefore, C2 uses the features suggested in previous name disambiguation work, as discussed in Section 2.

$$f = \sum_{i=1}^{6} w_i S_{i(e_1,e_2)} \tag{2}$$

ID1	**ID2**	**P**	**F**	**A**	**I**	**T**	**C**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
threshold								**0.4**	**0.26**	**0.4**	**0.4**	**0.35**	**0.385**
14Davies,N.J.	**65Davies,J.**	1	1	0.12	0.67	0.22	0	0.34	0.25	**0.45**	**0.40**	**0.60**	**0.50**
14Davies,N.J.	**68Davies,N.J.**	0	1	0.28	1	0.33	0	**0.54**	**0.40**	**0.58**	**0.52**	**0.52**	**0.44**
14Davies,N.J.	89Davies,J.	-1	0	0.13	0.67	0.18	0	0.33	0.25	0.20	0.20	0.25	0.20
14Davies,N.J.	**30Davies,J.**	1	1	0.27	0.67	0.33	0	**0.42**	**0.32**	**0.49**	**0.45**	**0.65**	**0.54**
14Davies,N.J.	98Davies,B.J.	-1	0	0.09	0	0	0	0.03	0.02	0.02	0.02	0.02	0.02
65Davies,J.	**68Davies,N.J.**	0	1	0.18	0.67	0.36	0	**0.40**	**0.30**	**0.47**	**0.44**	**0.44**	0.37
65Davies,J.	89Davies,J.	-1	0	0.08	1	0.25	0	*0.44*	*0.33*	0.28	0.27	0.33	0.27
65Davies,J.	**30Davies,J.**	1	1	0.16	1	0.18	0	**0.45**	**0.33**	**0.54**	**0.47**	**0.67**	**0.56**
65Davies,J.	98Davies,B.J.	-1	0	0.10	0.67	0	0	0.25	0.19	0.19	0.15	0.19	0.15
68Davies,N.J.	89Davies,J.	-1	0	0.18	0.67	0.31	0	0.38	0.29	0.22	0.23	0.29	0.23
68Davies,N.J.	**30Davies,J.**	-1	1	0.20	0.67	0.25	0	*0.37*	*0.28*	**0.47**	**0.42**	**0.53**	**0.42**
68Davies,N.J.	98Davies,B.J.	-1	0	0.12	0	0	0	0.04	0.03	0.03	0.02	0.03	0.02
89Davies,J.	30Davies,J.	-1	0	0.16	1	0.15	0	*0.44*	*0.33*	0.29	0.26	0.33	0.26
89Davies,J.	98Davies,B.J.	-1	0	0.20	0.67	0.12	0	0.33	0.25	0.22	0.20	0.25	0.20
30Davies,J.	98Davies,B.J.	-1	0	0.30	0.67	0	0	0.32	0.24	0.24	0.19	0.24	0.19
Accuracy								**73.33**	**73.33**	**100**	**100**	**100**	**100**

KEY:P=Sharing Publication Page, F=Identical Full Name, A=Abstract Similarity
I=Initials Similarity, T=Title Similarity, C=Co-author Similarity
C1=AIT, C2=AITC, C3=FAIT, C4=FAITC, C5=PFAIT, C6=PFAITC

Table 3. Instance unification results for a particular person called "J. Davies" (the author IDs in bold refer to this person)

where

$$f = \text{overallsimilarity}$$
$$w_i = \text{weightassignedtothe}i^{th}\text{feature}$$
$$i = \begin{cases} 1 \ \text{sharingpublication} \\ 2 \ \text{identicalfullname} \\ 3 \ \text{abstractsimilarity} \\ 4 \ \text{initialssimilarity} \\ 5 \ \text{titlesimilarity} \\ 6 \ \text{co} - \text{authorsimilarity} \end{cases}$$

$$S_i = \text{similarityforthe}i^{th}\text{feature}$$
where,
$$S_1 = \begin{cases} 1 & \text{authorssharepublicationpage} \\ 0 & \text{authorsdonotshareanypublicationpage} \\ -1 & \text{searchenginedoesnotrespond} \end{cases}$$
$$S_2 = \begin{cases} 1 & \text{authorshavesamefullname} \\ 0 & \text{authorshavedifferentfullname} \\ -1 & \text{searchenginedoesnotrespond} \end{cases}$$
$$S_{i\in\{3,4,5,6\}} = \text{(seeequation 1)}$$

For the initial experiments, all the features were given equal weight. Table 4 shows the name disambiguation results for the authors "D. Smith", "J. Davies", "Cooper", "Williams", "Brown", and "Jones", using different combinations of features. As discussed earlier, the similarity threshold for each different combination of features needs to be determined empirically. Therefore, we chose the values that yielded the maximum accuracy for the given combination of features on the first two authors "D. Smith" and "J. Davies", and used these threshold values to evaluate the algorithm's performance on the remaining authors.

To enable comparison between our approach to name disambiguation and previous work, we re-created the evaluation sets used in [8] by manually collecting the publications of the six authors named J. Anderson and seven named J. Smith. The original evaluation used eleven J. Smith authors, but we had to

Name	**AIT**	**AITC**	**FAIT**	**FAITC**	**PFAIT**	**PFAITC**
threshold	**0.4**	**0.26**	**0.4**	**0.4**	**0.35**	**0.385**
D. Smith(7)	95.24	95.24	85.71	85.71	80.95	90.48
J. Davies(6)	73.33	73.33	100	100	100	100
Cooper(5)	90	90	90	90	90	90
Brown(10)	100	100	100	100	100	100
Jones(10)	93.28	93.28	99.16	99.16	98.32	99.16
J. Anderson(6)	97.01	97.01	77.61	88.06	85.07	97.01
J. Smith(7)	93.33	93.33	84.44	95.56	93.33	97.78
Mean	94.72	94.72	90.24	94.56	93.34	96.79

Table 4. Evaluation of instance disambiguation for various authors

exclude four of them whose publications we could not find on the web. In comparison to the best score of 90% for the six J. Anderson authors reported in [8], our approach obtained 97.01% accuracy using all features (i.e. including the mined information). In case of J. Smith, [8] obtained accuracy of about 90%, whereas the accuracy obtained by our algorithm (although only for 7 authors in comparison to their 11 J. Smith authors) is 97.78%.

Since the main goal of this work is to identify which features lead to the best performance, we carried out an analysis of the results and the most interesting findings are as follows:

1. In some cases (e.g. D. Smith, J. Anderson), the combination of basic features (such as abstract, initials and title similarities) performed better than any other combinations. There are two reasons: (1) in these cases there were many similar words in the paper titles and abstracts, thus leading to high similarity scores on these features; and (2) some of these authors do not maintain their own publication pages or the web mining algorithm was not able to find them.
2. Co-authorship information does not help in most cases in our dataset. Given 100 author IDs, each ID pair referring to the same author, the algorithm was able to find only 23 author IDs where there was some overlap in the co-authors. On the other hand, surprisingly, we could find only 1 overlap in the names of co-authors among 300+ author ID pairs, where the authors were not identical. The first and the second columns in Table 4 show that there is no change in the results after co-authorship information is added.
3. Although though the algorithm for identifying authors' publication pages is very efficient, due to various limitations of the Google API (such as communication problems with the main Google server), results[7] are not guaranteed every time a query is issued. On the other hand, the Yahoo search engine's ranking algorithm has a poorer performance than Google's, so there is often a trade-off in using them.

[7] A result is a valid response from the Google server (i.e. it may return a set of documents, or no documents). By the term communication problems, we mean that the server encounters some errors and does not respond correctly.

4. As explained earlier, if the authors' full names are known, the names themselves can be used as the first disambiguation step (e.g. see results for "Jones" in Table 4). But in some cases (e.g. J. Anderson), where all the names in the dataset have the same first name "James", the similarity for each such pair will be equal to 1 (given that the middle name can not be identified or it is the same). Also, the initials similarities will be nearing 1. In such circumstances, the features such as full name similarity and initials similarity do not contribute much and should not be used on their own.
5. Last but not least, it must be noted that the evaluation experiments reported here are somewhat limited by the lack of bigger human-annotated datasets.

6 Conclusion and Future work

This paper addresses the instance unification problem and presents a fully automatic method which, given an ontology and an author name (either surname or initials and surname), retrieves the author IDs (instances) and relevant publications for the given name. It then tries to unify all instances which refer to the same individual in the real world. Citation information typically used in citation matching and author name disambiguation work is used as a basis (i.e., abstract, initials, titles and co-authorship information). The novel aspect is in the use of web mining in order to retrieve the full name of a given author and to find a publication page which contains the publications corresponding to the author IDs being considered for unification.

The approach is evaluated in a number of experiments carried out over some of the ambiguous author names in our ontology (i.e. "D. Smith", "J. Smith", "J. Davies", "J. Anderson" etc.). Since the aim of this work is to identify a set of relevant features that can be used for the instance disambiguation task, we perform an analysis over the results. In addition, we demonstrate that the information mined from the web leads to a substantial performance improvement on previous name disambiguation work using the J. Anderson and J. Smith dataset.

In our approach the two values weight and threshold are very important in deciding whether the two author IDs refer to the same person. For the experiments shown in this paper, equal weight was assigned to all features and the threshold was determined from the results of two authors "J. Davies" and "D. Smith".

As part of our future work, we will assign different weights to the features based on their importance and contribution in the overall result. Most of the previous work on instance disambiguation is based on Machine Learning (ML) algorithms. Having identified the correct combinations of relevant features, the next task will be to use these features and train some ML model (e.g., SVM or Naive Bayes). The threshold value, which helps in transforming probabilistic results into the exact results, will be derived for different combinations of features. According to the results "sameIndividualAs" statements will be added to the ontology.

References

1. H. Alani, S. Dasmahapatra, N. Gibbins, H. Glaser, S. Harris, Y. Kalfoglou, K. O'Hara, and N. Shadbolt. Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 317–334, Siguenza, Spain, 2002.
2. J. Bruijn and A. Polleres. Towards An Ontology Mapping Specification Language For the Semantic Web. Technical report, Digital Enterprise Research Institute, 2004.
3. F. Ciravegna, S. Chapman, A. Dingli, and Y. Wilks. Learning to Harvest Information for the Semantic Web. In *Proceedings of the 1st European Semantic Web Symposium*, Heraklion, Greece, May 2004.
4. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
5. D. G. Feitelson. On identifying name equivalences in digital libraries. *Information Research*, 9(4), 2004.
6. R. V. Guha and A. Garg. Disambiguating People in Search. In *Proceedings of the 13th World Wide Web Conference (WWW 2004), ACM Press*, 2004.
7. H. Han, C. L. Giles, and H. Zha. A model-based k-means algorithm for name disambiguation. In *Proceedings of the 2nd International Semantic Web Technologies for Searching and Retrieving Scientific Data*, Florida, USA, 2003.
8. H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'04)*, 2004.
9. G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In W. Daelemans and M. Osborne, editors, *Proceedings of the 7^{th} Conference on Natural Language Learning (CoNLL-2003)*, pages 33–40. Edmonton, Canada, May 2003.
10. B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 593 – 601, Banff, Canada, 2004.