Disambiguating Publication Venue Titles using Association Rules

Denilson Alves Pereira Eduardo Emanuel Braga da Silva Ahmed A. A. Esmin

Department of Computer Science
Federal University of Lavras
Lavras, Brazil
denilsonpereira@dcc.ufla.br
maneul2000@gmail.com
ahmed@dcc.ufla.br

ABSTRACT

Research agencies in several countries evaluate the impact of scientific publications of researcher groups to define their investments, and one of the main used metrics is the quality of the publication venues where their works were published. Several bibliometric indexes have been formulated by measuring the quality of a publication venue. However, given a set of citations extracted, for example, from curricula vitae of a researcher group, to effectively use bibliometric indexes to evaluate their quality it is necessary to identify correctly the publication venue title of each citation. This task is not easy, since there are not unique identifiers for publication venues. Frequently, citations contain abbreviated forms and acronyms, publication venues share similar titles, sometimes they change their titles, divide or merge, creating new ones. Traditional digital libraries deal with this problem by creating Authority Files. In this work, we present a twofold contribution: (i) the creation of a Computer Science publication venue authority file and (ii) the proposal of a method that uses association rules to disambiguate publication venue titles originated from citations. The disambiguator is a supervised learning method that uses the authority file to train a classifier, whose generated model is a set of association rules to identify publication venues. Experiments show that our method obtains better results than three state of art baselines.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

General Terms

Algorithms, Experimentation

Keywords

Citation, Publication Venue, Association Rules, Authority File, Entity Resolution

1. INTRODUCTION

We are developing a project that aims at comparing research groups based on the quality of the publication venue of their scientific publications. Several bibliometric indexes have been formulated by measuring the quality of a publication venue, such as the Journal Impact Factor (JIF)[1], the SCImago Journal & Country Rank (SJR)[2], and the Qualis Capes[3] (Brazilian system). However, given a set of citations (understand citation as a bibliographic record containing features about a particular publication, such as author names, work title, and publication venue title) extracted from curricula vitae of researchers, to effectively use such bibliometric indexes it is necessary to identify correctly the publication venue title of each citation. This task is not easy.

Consider, for example, the publication venue title reference strings in Table 1. Lines 1–4 present variant forms of referring to the VLDB Conference, and Lines 5–6 present variant forms of referring to the VLDB Journal, two distinct publication venues. However, traditional methods that use string similarity metrics [1], such as Cosine Similarity [2], Jaccard Coefficient Similarity [3], or Edit Distance [4], do not work well for clustering the strings of this example. Such methods tend to consider similar the strings of Lines 2 and 6, from two distinct publication venues, and not similar the strings of Lines 1 and 2, from the same publication venue. Furthermore, it is difficult to adjust a threshold level to consider two strings similar.

1	VLDB Conference
2	International Conference on Very Large Data Bases
3	International Conference on Very Large Databases
4	Int. Conf. on Very Large Data Bases (VLDB)
5	The VLDB Journal
6	The International Journal on Very Large Data Bases

Table 1: Examples of publication venue title reference strings

[1]http://wokinfo.com/products_tools/analytical/jcr/
[2]http://www.scimagojr.com/
[3]http://qualis.capes.gov.br/webqualis/principal.seam

Traditional digital libraries deal with the problem of identifying variant forms of referring to a same entity by creating Authority Files. According to Auld [5], an authority file maintains the correspondence among variant strings used to refer to an entity in a particular bibliographic field. One of the main initiatives in this area is the Virtual International Authority File (VIAF) project [6], which combines the authority files from several of the world's national bibliographic agencies into a single authority service, available on the Web.

In Computer Science area, the main digital libraries (DBLP[4], ACM Digital Library[5], IEEE Computer Science Digital Library[6], and CiteSeerX[7]) do not maintain explicit authority files. Regarding with publication venues, information about them are not available in a structured form. DBLP, for example, maintain a web page for each publication venue, which contains historical information of each edition of the publication venue. However, since data are not structured it is not easy to automatically extract the variant forms of referring to a same publication venue. Also, there is no explicit information about changing in publication venue titles, fusions and divisions of publication venues, which are specially common in conferences [7].

Our hypothesis is that if we have a well constructed publication venue authority file we are able to develop a method to disambiguate publication venue titles originated from citations, which identifies sets of keywords that discriminate different publication venues. That is, sets of keywords that occur in variant strings of only one publication venue. For example, considering the strings of Table 1 the sets of keywords {VLDB, Conference} and {Conference, Data, Bases} can distinguish the VLDB Conference from the VLDB Journal, and the sets of keywords {VLDB, Journal} and {Journal, Very, Large} can also distinguish the VLDB Journal from the VLDB Conference.

In this work, we propose a method to disambiguate publication venue titles from citations that uses association rules [8] to find sets of keywords that distinguish each publication venue in an authority file. It is a supervised learning method that uses an authority file to train a classifier, whose generated model is a set of association rules to identify publication venues. The association rules are of the form $\mathcal{X} \rightarrow pv_i$, where $\mathcal{X}$ is a set of keywords and pv_i is a publication venue (e.g.,$\{VLDB, Journal\} \rightarrow pv_2$). In the test phase, we generate sets of keywords from the string to be tested, and try to match them with the antecedents of the rules in model generated in the training phase. Experiments show that our method obtains better results than our baselines.

The main contributions of this work are:

- the creation of a Computer Science publication venue authority file;
- the proposal of a method that uses association rules to disambiguate publication venue titles originated from citations;
- a set of experiments demonstrating the effectiveness of our method;
- a tool available online to search our authority file and disambiguate publication venue titles.

The remaining of this paper is organized as follows: In Section 2, we discuss related works on authority file and methods for entity disambiguation. In Section 3, we discuss the creation of our Computer Science publication venue authority file. In Section 4, we present our method that uses association rules to disambiguate publication venue titles originated from citations, and in Section 5, we discuss its computational complexity. In Section 6, we describe our experiments, evaluation metrics, and results. Finally, in Section 7, we present our conclusions and future works.

2. RELATED WORK

The Virtual International Authority File (VIAF) project [6] is the main initiative of digital libraries to maintain authority files. It combines multiple name authority files into a single authority service, available on the Web. In [9], the authors describe the initial creation of VIAF and the details of its automated name matching algorithms. Our proposal is not create an authority file such as VIAF, we only create a specific Computer Science publication venue authority file, and used it to disambiguate bibliographic citations. VIAF do not provide detailed records on Computer Science publication venues, specially on conferences. Our work can also contribute to enhance the VIAF name matching algorithms.

Initiatives like VIAF tend to be limited to National Libraries, leaving out many other organizations that are responsible for large amount of bibliographic data such as specialized digital libraries. In [10], the authors present a proposal in direction of overcoming such limitations, creating a tool that generates authority files using the context of the semantic web. Our work also contributes in the direction of specialized solutions for creating authority files and for disambiguating bibliographic data. In [11], the authors present an application that harvests digital repositories data, and enables citation management and configurable custom reporting. Similar to our work, they also created an authority file for publication venues. Their authority file also was produced semi-automatically through clustering of publication venue titles and subsequent manual correction of errors.

To create our authority file, we used approximate string matching and clustering techniques, such as in [1]. Another similar approach was investigated in [12], which used a web search engine to find additional information to help cluster publication venues. Their ideas were extended in a more generic framework, described in [13]. To create our authority file, we did not use a web search engine because our data sources were already wrapped from the main web sources.

Other works with distinct objectives but related to ours are found in the literature. The works [14] and [15] proposed several algorithms for matching citations from different sources based on edit distance, word matching, phrase matching, and attribute extraction. The work [16] discusses the problem of data quality in DBLP, presenting the difficulties to maintain an authority control on entity names (journal, conference, person etc.) in order to guarantee that they are always represented by the same character string and that distinct entities do not share the same representation. Related to author name disambiguation, [17] presents a brief survey and proposes a taxonomy for characterizing the current methods. Out of the context of bibliographic

[4]http://www.informatik.uni-trier.de/~ley/db/
[5]http://dl.acm.org/
[6]http://www.computer.org/csdl
[7]http://citeseerx.ist.psu.edu/

data, [18] uses data mining and various types of evidence to disambiguate names of composers, lyricists, and arrangers in the Levy Sheet Music Collection, and in [19], the authors describe a system that locates the occurrences of named entities within a text, when given *a priori* a set of related names included in authority lists.

Disambiguating publication venues is a specific case of entity resolution, where the attribute publication venue title of a bibliographic citation is the entity. Given a set of entity references, such as publication venue titles, entity resolution is the process of identifying which of them correspond to the same real-world entity [20]. In a recent survey on entity resolution (or entity matching), [21] presents an implementation of a framework for evaluating entity matching systems through a systematic generation of synthetic test cases. Other surveys and tutorials on entity resolution can be found in [22], [23], [24], and [25].

To disambiguate publication venues in bibliographic citations, our work uses association rules [8], a data mining technique that can find out the relationship among item sets in a dataset. Association rules were also used in the work [26] to disambiguate author names in bibliographic citations. They proposed supervised learning methods where tokens in coauthor names, work title, publication venue titles are used as features to train classifiers that exploit rules associating citation features to specific authors. Association rules were also used to classify documents in [27].

3. CREATION OF A COMPUTER SCIENCE PUBLICATION VENUE AUTHORITY FILE

The creation of a Computer Science publication venue authority file is a contribution of this work. An authority file is an index of authority records, where each record representing a publication venue is composed of a heading to be used as the publication venue label and a list of variant labels also used to refer to the publication venue, called cross references. The headings and the cross references can be used by a search system to answer queries related to a publication venue.

Our authority file contains information about journals, conferences, and workshops. For each publication venue, the authority file stores the variant forms of writing the current title and acronym, title and acronym formerly, when the publication venue changed its name, and merge of, when the publication venue was originated from the fusion of two or more other publication venues. It also stores some extra information, such as issn, publisher, language, subject, web site, and bibliometric indexes (JIF and Qualis Capes). Figure 2 shows an example of part of a record in the publication venue authority file.

The authority file was created collecting data from the main digital libraries and institutions that organize publication venue rankings of quality in the computer science area. The sources of data were: DBLP[8], ACM Digital Library[9], IEEE Computer Science Digital Library[10], Wikipedia[11,12], Qualis Capes[13], and Web of Knowledge (Journal Citation Report)[14]. All sources contain data about conferences and journals, except the last, that contains only journals.

We developed wrappers to collect data from each data source. A wrapper [28] is a specialized program that identifies data of interest and map them to some suitable format, in our case, XML. The data sources are not structured, some of them have hierarchical structures, the information is incomplete, and sometimes, incorrect. In order to identify variant forms of writing a publication venue title we need to extract the title of each edition of the publication venue. In the ACM Digital Library, for example, there is a link to browse the proceedings, and they are organized by edition. However, the publication venue titles are not explicit, sometimes they are in the middle of other texts containing volume, demos sections and other data. In DBLP, there is a web page for each publication venue. The main title is in the top of the page, but the other variant forms to refer to the publication venue are not explicit. In this page, there is a historical information about each edition of the publication venue, but the title of each one is implicit, in the middle of other data, such as editor names, local, date, and workshops.

In our strategy, data from each data source were extracted by a wrapper and stored in a XML file, in a semi-structured format. In a cleaning process, stopwords (articles, conjunctions, prepositions), and some words such as "proceedings", "international", ordinal numbers, names of months, and years were removed, and some misspelling were fixed. A clustering algorithm, based on the K-Nearest Neighbors (Knn) method [29], was applied, in order to identify record replicas. The algorithm used the Jaccard similarity coefficient [3] as a metric to identify similar strings. After that, a human specialist checked the consistency of the result and manually solved the problems.

When data of each data source was consistent, then all of them were integrated. They were fused in pairs to form a single database. We followed the same principle as before, applying a clustering algorithm to fuse each pair of data source and having a human specialist checking the result of each fusion. As result, we created an authority file containing $11,592$ references to $5,524$ distinct publication venues ($1,937$ journals, $2,227$ conferences, $1,344$ workshops, and 16 magazines). A web search interface is available to query the authority file and disambiguate publication venue titles[15].

4. OUR METHOD TO DISAMBIGUATE PUBLICATION VENUE TITLES

4.1 Problem Formulation

The publication venue title disambiguation task may be formulated as a classification problem as follows. Let $C = \{c_1, c_2, ..., c_k\}$ be a set of citations. Each citation c_i has a list of attributes such as author names, work title, and publication venue title. Let $PV = \{pv_1, pv_2, ..., pv_l\}$ be a set of L classes with their respective labels, in this case, a set of publication venues. The objective is to produce a

[8]http://www.informatik.uni-trier.de/~ley/db/
[9]http://dl.acm.org/
[10]http://www.computer.org/csdl
[11]http://en.wikipedia.org/wiki/List_of_computer_science_conferences
[12]http://en.wikipedia.org/wiki/Category:Computer_science_journals
[13]http://qualis.capes.gov.br/webqualis/principal.seam
[14]http://wokinfo.com/products_tools/analytical/jcr/
[15]http://pvaf.dcc.ufla.br

pv.head	ECML PKDD - European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
pv.mergeof	ECML - European Conference on Machine Learning
pv.mergeof	PKDD - European Conference on Principles and Practice of Knowledge Discovery in Databases
pv.titleformerly	EWSL - European Working Session on Learning
pv.titleformerly	PKDD - European Symposium on Principles of Data Mining and Knowledge Discovery from Databases
pv.qualis	A2

Table 2: Example of a record in the publication venue authority file. The attribute *pv.head* is the heading, the following attributes contain the cross references of the authority record, and the *pv.qualis* attribute shows the Qualis Capes bibliometric index. The heading includes the acronym and the canonical title of the publication venue.

disambiguation function that maps each citation c_i into one of the predefined classes of the set PV.

Our proposal to solve the classification problem uses a supervised machine learning technique. In this case, we are given an input dataset, called the *training data* and denoted as $\mathcal{D}$, which consists of examples of publication venue reference strings for which the correct publication venue is known. Each example is composed of a set of m features $(f_1, f_2, ..., f_m)$ and a class label $pv_i \in PV$, which uniquely identifies a publication venue. The features are words in the publication venue reference string. The training data is used to produce a learning model that relates the features in the training data to the correct publication venue. The *test data*, denoted as $\mathcal{T}$, for the classification problem consists of a set of citations for which the features are known while the correct publication venue is unknown. From these citations, we use only the publication venue title attribute. The learning model, which is a function that maps $\{f_1, f_2, ..., f_m\}$ to $\{pv_1, pv_2, ..., pv_l\}$, is used to predict the correct publication venue of citations in the test set.

The learning function uses an associative disambiguator to exploit associations among words in the publication venue reference strings that uniquely identify the publication venues. Such associations are uncovered using rules of the form $\mathcal{X} \rightarrow pv_i$, where $\mathcal{X} \subseteq \{f_1, f_2, ..., f_m\}$ and $pv_i \in PV$. For example, $\{Conference, Data, Bases\} \rightarrow pv_1$ and $\{VLDB, Conference\} \rightarrow pv_1$ are two association rules indicating that the set of words $\{Conference, Data, Bases\}$ and the set $\{VLDB, Conference\}$, both, uniquely identify the publication venue pv_1 (VLDB - International Conference on Very Large Data Bases), while $\{VLDB, Journal\} \rightarrow pv_2$ is an association rule identifying the publication venue pv_2 (The VLDB Journal).

In order to produce association rules that uniquely identify each publication venue, the associative disambiguator only learns rules that have 100% of confidence. According to [8], a rule $\mathcal{X} \rightarrow \mathcal{Y}$ holds in the dataset $\mathcal{D}$ with confidence c if c% of instances in $\mathcal{D}$ that contain $\mathcal{X}$ also contain $\mathcal{Y}$. Then, the generated model does not contain rules $\mathcal{X} \rightarrow pv_i$ and $\mathcal{X} \rightarrow pv_j$ such that $i \neq j$. This strategy is not perfect since it may not produce rules for all publication venues. Such situation occurs when the sets of words in all reference strings of a publication venue are contained in some set of words of reference strings of a distinct publication venue. In this case, no rule is generated for the publication venue that has the subsets of words. To solve such situation, in the test phase, to predict the publication venue for a string for which no rule is found in the learning model, our method uses similarity string comparison as it will be explained latter.

4.2 Training Phase

The training data $\mathcal{D}$ can be provided by a data source such as an authority file, as described in Section 3. Each input instance in $\mathcal{D}$ is a reference string r_{ji} for the publication venue pv_i, and there are one or more instances for each distinct publication venue. Let $\mathcal{R}_{pv_i}^{r_{ji}}$ be the set of rules $\mathcal{X} \rightarrow pv_i$ where $\mathcal{X} \subseteq r_{ji}$ (i.e., r_{ji} contains all features in $\mathcal{X}$). That is, $\mathcal{R}_{pv_i}^{r_{ji}}$ is composed of rules predicting publication venue pv_i originated from reference string r_{ji}. Let $\mathcal{R}_{pv_i}$ be the set of rules predicting the publication venue pv_i, and let $\mathcal{R}$ be the set of all rules generated by the learning model. Then, $\mathcal{R}_{pv_i}^{r_{ji}} \subseteq \mathcal{R}_{pv_i} \subseteq \mathcal{R}$.

Algorithm 1 shows the steps of the training phase. It receives a set of reference strings for which the publication venues are known, and returns a set of associative rules that have 100% of confidence to predict these publication venues. The first step of the algorithm (Lines 1–6) inserts the distinct words from the reference strings into an inverted index structure [2]. This structure is formed by pairs key-value, where the key is a word and the value is an occurrence list of this word, containing, in each position, the publication venue i and its specific reference string identification j. Such operation is executed by the InsertInvertedIndex function. The Tokenize function split a reference string into words.

The second step of the algorithm (Lines 8–21) is an iterative process to create associative rules. The GenItemSets function generate sets of items (words) of size k combining the items of size $k-1$ obtained in the previous iteration. In [8], the authors present an algorithm to combine item sets. Each k-itemset is searched in the inverted index, and if it occurs in only one publication venue, a rule is created containing the k-itemset as antecedent and the publication venue id as consequent (Lines 12–16). The SizeOccurrenceList function returns the number of distinct publication venues in which a k-itemset appears in a same reference string. This function searches in the inverted index using each word in a k-itemset as key and retrieve its occurrence list. When $k > 1$, it makes an intersect operation of occurrence lists of each word to find the result. If size is equal to 1 then a new rule is inserted in the set of rules by the InsertRule function.

If a k-itemset forms a rule then any l-itemset, $l > k$, that includes this k-itemset also forms. Then, the algorithm uses a pruning strategy, avoiding combine this k-itemset in the next iteration (RemoveItemSet function in Line 17). We do not use pruning by support, we consider the minimum support equals to 1, since a single occurrence of a combination of words is important to identify a publication venue. Support of a rule $\mathcal{X} \rightarrow \mathcal{Y}$ is defined as the number of instances in the dataset $\mathcal{D}$ that contain $\mathcal{X} \cup \mathcal{Y}$.

Algorithm 1 Training Phase

Require: Examples for training $\mathcal{D}$
Ensure: The set of rules $\mathcal{R}$

for each reference string $r_{ji} \in \mathcal{D}$ **do**
  $C_1 \leftarrow$ Tokenize(r_{ji})
  for each 1-itemset $it \subset C_1$ **do**
    InsertInvertedIndex(it, j, pv_i)
  end for
end for
$\mathcal{R} \leftarrow \emptyset$
for each reference string $r_{ji} \in \mathcal{D}$ **do**
  $C_0 \leftarrow$ Tokenize(r_{ji})
  $m \leftarrow$ Length(C_0)
  for ($k \leftarrow 1; k \leq m; k++$) **do**
    $C_k \leftarrow$ GenItemSets(k, C_{k-1})
    for each k-itemset $it \subset C_k$ **do**
      if SizeOccurrenceList(it) = 1 **then**
        // 100% confidence rule
        InsertRule($it \rightarrow pv_i$, $\mathcal{R}$)
        RemoveItemSet(it, C_k)
      end if
    end for
  end for
end for
return $\mathcal{R}$

Example 1. Table 3 illustrates an example of training data and the rules generated by their reference strings. Notice that there is no rule for the pv_2 class, since its tokens form a subset of the tokens of the pv_1 class. Also, rules such as $\{csa, security\} \rightarrow pv_5$ is not generated, since it is a superrule of $\{csa\} \rightarrow pv_5$. We considered the token "symposium" = "conference".

4.3 Test Phase

The test data $\mathcal{T}$ is composed of a set of citations. From each test citation c_i, we use only the publication venue title attribute, pvc_i. First, pvc_i, of size m, is tokenized and the k-itemsets, $1 \leq k \leq m$, are generated. Second, each itemset is matched against the antecedents of the rules $\mathcal{R}$ in the learning model. All rules whose antecedents match with any itemset form the set of candidate rules, R_{pvc_i}. Third, using a vote schema, the consequent of each rule in R_{pvc_i} are counted, and the publication venue $pv_i \in PV$ with the highest counting is chosen as the class of the citation c_i. In case of tie in the voting, the reference strings used in the training phase for the pv_i's with the same counting are compared with the string pvc_i using a similarity metric (Jaccard or Cosine, for example), and the pv_i correspondent to the reference string with the highest similarity is chosen as the class of the citation c_i.

In case of no rule is found in $\mathcal{R}$ whose antecedent matches with the itemsets from pvc_i, our method uses Jaccard similarity metric (although other metrics can also be used) to choose the class of c_i. In this case, each reference string used in training phase is compared with the string pvc_i, and the pv_i corresponding to the reference string with the highest similarity is chosen as the class of the citation c_i.

Algorithm 2 shows the details of the test phase to predict the class of the publication venue pvc of a citation in the test data.

Training reference Strings	**Class**
InfoVis IEEE Information Visualization Conference	pv_1
Information Visualization	pv_2
International Conference on Communication Systems and Applications	pv_3
International Conference on Optical Communication Systems	pv_4
CSA Cloud Security alliance	pv_5
SSNDS IEEE International Symposium on Security in Networks and Distributed Systems	pv_6
Training Rules	
$\{infovis\} \rightarrow pv_1$ $\{ieee, information\} \rightarrow pv_1$ $\{ieee, visualization\} \rightarrow pv_1$ $\{conference, visualization\} \rightarrow pv_1$ $\{conference, information\} \rightarrow pv_1$	pv_1
	pv_2
$\{applications\} \rightarrow pv_3$	pv_3
$\{optical\} \rightarrow pv_4$	pv_4
$\{alliance\} \rightarrow pv_5$ $\{csa\} \rightarrow pv_5$ $\{cloud\} \rightarrow pv_5$	pv_5
$\{ssnds\} \rightarrow pv_6$ $\{networks\} \rightarrow pv_6$ $\{distributed\} \rightarrow pv_6$ $\{ieee, security\} \rightarrow pv_6$ $\{ieee, systems\} \rightarrow pv_6$ $\{security, systems\} \rightarrow pv_6$ $\{conference, security\} \rightarrow pv_6$	pv_6

Table 3: An example of training data and their generated rules

Algorithm 2 Test Phase

Require: $\mathcal{R}$, $\mathcal{D}$, $pvc \in \mathcal{T}$
Ensure: The predicted publication venue of pvc

$C_0 \leftarrow$ Tokenize(pvc)
$m \leftarrow$ Length(C_0)
$\mathcal{R}_{pvc} \leftarrow \emptyset$
for ($k \leftarrow 1; k \leq m; k++$) **do**
  $C_k \leftarrow$ GenItemSets(k, C_{k-1})
  for each k-itemset $it \subset C_k$ **do**
    for each $\mathcal{X} \rightarrow pv$ such that $it = \mathcal{X}$ **do**
      $\mathcal{R}_{pvc} \leftarrow \mathcal{R}_{pvc} \cup \mathcal{X} \rightarrow pv$
    end for
  end for
end for
if $\mathcal{R}_{pvc} = \emptyset$ **then**
  $ppv \leftarrow$ PredictBySimilarity($\mathcal{D}$, pvc)
  return ppv // the predicted publication venue of pvc
end if
for each $r \in \mathcal{R}_{pvc}$ in the form $\mathcal{X} \rightarrow pv$ **do**
  $pv.count++$
end for
$ppv \leftarrow pv_i$ such that $pv_i.count > pv_j.count \; \forall j \neq i$
if there are more than one pv_i with the same highest counting **then**
  $ppv \leftarrow$ PredictBySimilarity(all $r \in \mathcal{R}_{pvc}$ containing pv_i with the same highest counting, $\mathcal{D}$, pvc)
end if
return ppv // the predicted publication venue of pvc

Example 2. Table 4 illustrates an example of three reference strings in test data and the rules in the training data of Table 3 that match with the itemsets generated by the test reference strings. For the first test string, there are three rules in the training model indicating pv_6 as the correct class, and for voting it is the chosen class (Line 19 of Algorithm 2). For the second test string, occurs a tie between the pv_3 and pv_5 classes. In this case, the Jaccard similarity metric chooses pv_3 as the correct class (Lines 20–22 of Algorithm 2). And for the last test string, as there is no rule in the training model that matches with this string then the decision is also made by a similarity metric (Lines 12–15 of Algorithm 2).

Test reference Strings	Rules
Security in Networks and Distributed Systems	$\{networks\} \rightarrow pv_6$ $\{distributed\} \rightarrow pv_6$ $\{security, systems\} \rightarrow pv_6$
CSA Communication Systems and Applications	$\{csa\} \rightarrow pv_5$ $\{applications\} \rightarrow pv_3$
Information Visualization	

Table 4: An example of test data and the rules of Table 3 that match with the itemsets generated by the test reference strings

5. COMPUTATIONAL COMPLEXITY

The computational complexity time of our method in the training phase is dominated by the number of reference strings to be trained and the number of tokens in these strings. All tokens in each string need to be combined to form the itemsets. Let n be the number of reference strings and let m be the average number of tokens in theses strings. Then, the computational complexity time of the method is $O(n * 2^m)$.

However, the number of tokens per string, m, is not so high, and it is not dependent of the size of the input, n. It depends only of the characteristics of the reference strings. In our experiments, using the computer science publication venue authority file, the number of tokens varies between 1 and 15, removing stopwords, with average equals to 4.34. In the test datasets (see the description of the test datasets in Section 6.1), the number of tokens varies between 1 and 11, with average equals to 4.41 in the Google dataset, and between 1 and 15, with average equals to 4.83 in the Microsoft dataset. Then, m can be considered a constant, and the computational complexity time of the method becomes linear, $O(n)$.

The same reasoning can be done for the test phase of the algorithm. Notice that the search in the set of rules, $\mathcal{R}$, in the learning model can be done in $O(1)$ using a hash table to keep the rules.

6. EXPERIMENTAL RESULTS

In this section we describe our experiments to evaluate the usefulness of association rules to disambiguate publication venue titles using an authority file as training dataset.

6.1 Datasets

To train our classifier we use the Computer Science Publication Venue Authority File presented in Section 3. It contains 11,592 reference strings to 5,524 distinct publication venues (journals, conferences, and workshops).

To test our classifier we use two datasets. The first one is a real dataset obtained by querying Google Scholar. This dataset was used in the works [12] and [13]. It contains 1,110 distinct reference strings to 55 distinct publication venues, having on an average 20.2 strings per class, the largest class has 81 strings, and there are only 7 classes with only one string. In the following sections we call this dataset as Google dataset.

The second dataset is composed of the 3,113 distinct publication venue title strings collected from Microsoft Academic Search[16]. We collected all conference and journal title strings from the Computer Science Rank and manually matched them against our authority file in order to identify the class of each string. All non-matched strings were removed, as well as the strings that contained only abbreviated forms. In the following sections we call this dataset as Microsoft dataset.

6.2 Evaluation Metrics

To evaluate the quality of our classifier, we used the metrics accuracy, micro-average, and macro-average F_1. Accuracy is the fraction of the test strings assigned to their correct classes by the classifier. The F_1 measure is defined as:

$$F_1 = \frac{2rp}{r+p}$$

where p is the precision of the classifier and r is its recall.

Micro-average F_1 corresponds to a global F_1 value obtained by computing precision and recall over all classes. Macro-average F_1 is computed simply by averaging F_1 across all classes [2].

6.3 Baselines

We compared our method with three baselines: a Jaccard, a Cosine, and a SVM based method. In the Jaccard based method, the Jaccard Coefficient Similarity [3] is used to compare each reference string in the test dataset against each reference string in the training dataset. The publication venue in the training dataset associated with the reference string with the highest similarity is chosen to be the class of the tested string.

In the Cosine based method, we used the same strategy of the Jaccard method, except that the Cosine Similarity is used to compare the strings. Each string is represented as a vector of token weights using the distinct tokens in the training dataset. The token weights are computed as the inverse-document-frequency (IDF) [2].

In the SVM based method, each publication venue is associated with a class and a SVM classifier [30] for that class is trained. Each reference string is represented as a feature vector with each token corresponding to a feature, and its IDF value being the feature weight. For the experiments, we used the package LibLinear[17]. We use a grid search to find the best parameter values for the training step, i.e., before applying the method to the test data.

[16] http://academic.research.microsoft.com
[17] http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

Method	*Google dataset* (%)			*Microsoft dataset* (%)		
	Accuracy	MicroF1	MacroF1	Accuracy	MicroF1	MacroF1
Our method	92.88	96.18	92.97	97.75	98.45	98.14
Jaccard	88.74	93.72	89.85	94.44	95.64	94.43
Cosine	91.98	95.42	91.43	95.12	96.36	95.04
SVM	91.53	95.18	88.99	96.21	97.07	96.39

Table 5: Results comparing our method against the baselines.

	Google dataset			*Microsoft dataset*		
	Correct	Incorrect	Total	Correct	Incorrect	Total
Our method	757	21	778	2,766	7	2,773
Jaccard	717	61	778	2,664	109	2,773
Cosine	756	22	778	2,683	90	2,773
SVM	756	22	778	2,734	39	2,773

Table 6: Number of instances classified correctly and incorrectly for the strings predicted by rule in our method.

6.4 Results and Discussions

The first experiment aims at comparing our method against the baselines to disambiguate publication venues reference strings. Table 5 presents the results on Google and Microsoft datasets, using accuracy, micro F1, and macro F1 metrics. Our method obtains the best numbers in both datasets and in all metrics. The Cosine based classifier was the best baseline in the Google Dataset, and the SVM based method was the best baseline in the Microsoft dataset. Statistically, considering a 95% confidence level, we can state that the our method is superior to the baselines on the Microsoft dataset, and statistically tied with Cosine and SVM on the Google dataset.

The results for all methods were better on Microsoft dataset. This dataset contains only one reference string per class, and the strings represent publication venue acronyms and titles as they are known in Microsoft Academic Search. The Google dataset is more difficult to disambiguate, since it contains reference strings as they were automatically collected by an extractor software. Due to imperfect extraction, some strings contain part of work title, location or other noisy token along with the publication venue title. For example, "Operations Systems, Communications of the ACM, University of Waterloo" is a reference string to "Communications of the ACM". The results for macro F1 was lower than for micro F1 on Google dataset because the errors were concentrated in instances of a same class and in classes containing few instances.

As explained in Section 4, test phase of our method, there are some situations where no rule is found in the training model that matches with the itemsets generated by a test string. In Table 6, we analyze the quantity of test strings for which there are at least one rule in the training model, the hit rate, and how the baselines behave for the same test strings. For the Google dataset, 778 test strings, that is, 70.1% of $1,110$, were solved by rule in our method. The remaining test strings were solved by comparison, using the Jaccard similarity metric. For the Microsoft dataset, this number is higher, $2,773$, that is, 89.1% of $3,113$ test strings were solved by rule. For the Google dataset, our method hits only one string more than the baselines Cosine and SVM based methods. For the Microsoft dataset, the hit rate of our method is higher, it classifies incorrectly only 7 strings in $2,773$.

In another experiment, we analyzed the influence of a good authority file in the results of the disambiguation methods. We trained the classifier using only the main acronym and title of each publication venue in the authority file, removing other variant forms of referring to it. The resulting values dropped to 87.92, 93.37, and 85.48 for metrics accuracy, micro F1, and macro F1, respectively, on the Google dataset, and 95.05, 96.32, and 95.76 on the Microsoft dataset. Compared to the values in Table 5 (line our method), it was a substantial drop. Similar declines occurred for all baselines. This experiment shows that a well constructed authority file can contribute to improve disambiguation methods.

Difficulties, Problems and Limitations

In this section, we show some cases for which our method failed, discuss the reasons for these failures, and illustrate them with some examples. We also discuss some limitations of our method and some difficulties to create an authority file.

One of the failure cases of our method is due the existence of noisy tokens in test strings. For example, in the real test string "Energy Minimization Methods in Computer Vision and Pattern Recognition" the tokens "Energy" and "Methods", that are not part of the publication venue title, occurs in titles of other publication venues in our training dataset. Such tokens generated rules that point to this other publication venues, and caused failure in our method.

Some test strings contained forms completely abbreviated, such as "PROC. DES. AUTOM. CONF." abbreviation of "Proceedings of Design Automation Conference", which was another case of failure. Our authority file contains few abbreviated forms, only those collected from the Web of Knowledge for journals.

Some cases of failure can be solved by improving our authority file, by adding new variant forms of referring to the publication venues. Due to many similar titles, sometimes a token such as "ACM" or "IEEE" may make the difference between an error or hit.

Our method is limited to solve cases for which there is no rule in the training model. Such cases occurs when the tokens of a publication venue title is subset of another title. In this case, no rule for the smaller title will be generated. There are hard cases to solve, such as the existence of two journals with the same set of tokens: "Advances in Engineering Software", issn 0965-9978, and "Advances in Software

Engineering", issn 1687-8655.

One of the main difficulties we faced was in the construction of our authority file. The data sources (e.g. DBLP, ACM, IEEE) do not provide the publication venue titles and their variant forms in a structured way. The data are difficult to collect and there are many inconsistencies among them, which required a lot of manual effort to organize them.

7. CONCLUSIONS AND FUTURE WORK

In this work, we presented the strategies we used to create a Computer Science publication venue authority file. We developed wrappers and collected data from the main digital libraries and institutions that organizes publication venue rankings of quality in the computer science area. The result was the creation of an authority file containing 11,592 references to 5,524 distinct publication venues. The authority file stores variant forms of referring to each publication venue, besides information about formerly titles, changing in names, merging, bibliometric indexes and others.

We also proposed a method that uses association rules to disambiguate publication venue titles originated from citations. The experiments show that our method obtains better results than three state of art baselines. Experiments also show that using our authority file as a training dataset improve the results, including those of the baselines. Moreover, our method is simple, does not require adjust complex parameters, and has a good computational complexity.

For future works, the main challenges is to keep the authority file updated and to add new information to it. We are working on a mechanism to detect new publication venues and add them to the authority file. We are also developing new tools to extract from the Web additional information, such as sites and editions of each conference, publisher of each journal, tables of contents and others. This information will enrich our authority file and provide new features to disambiguate citations. After that, we will be able to provide more precise tools to compare research groups such as in [31], using more data sources.

Acknowledgements

This work was partially supported by the FAPEMIG grant CEX-APQ-01964-11 and individual scholarships from PIBIC FAPEMIG.

8. REFERENCES

[1] J. C. French, A. L. Powell, and E. Schulman, "Using clustering strategies for creating authority files," *J. Amer. Soc. Inform. Sci.*, vol. 51, no. 8, pp. 774–786, 2000.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval: The Concepts and Technology behind Search.* Addison-Wesley Professional, 2011.

[3] P. Jaccard, "Étude comparative de la distribuition florale dans une portion des Alpes et des Jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901.

[4] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[5] L. Auld, "Authority control: An eight-year review," *Library Resources & Tech. Services*, vol. 26, pp. 319–330, 1982.

[6] "VIAF: The virtual international authority file," 2014, http://viaf.org/. Accessed in February, 2014.

[7] D. Lee et al., "Are your citations clean?" *Commun. ACM*, vol. 50, no. 12, pp. 33–38, December 2007.

[8] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. of the 20th Int. Conf. on Very Large Data Bases*, Santiago, Chile, 1994, pp. 487–499.

[9] R. Bennett et al., "VIAF (virtual international authority file): Linking die deutsche bibliothek and library of congress name authority files," in *Proc. of the World Library and Information Congr.: 72nd IFLA General Conf. and Council*, Seoul, Korea, August 2006.

[10] A. Leiva-Mederos et al., "AUTHORIS: a tool for authority control in the semantic web," *Library Hi Tech*, vol. 31, no. 3, pp. 536–553, 2013.

[11] N. Houssos et al., "Implementing citation management and report generation value-added services over oai-pmh compliant repositories," in *Proc. of the 5th Int. Conf. on Open Repositories*, Madrid, Spain, July 2010.

[12] D. A. Pereira et al., "Using web information for creating publication venue authority files," in *Proc. of the 8th ACM/IEEE-CS Joint Conf. on Digital Libraries.* Pittsburgh, USA: ACM New York, NY, USA, June 2008, pp. 295–304.

[13] D. A. Pereira et al., "A generic web-based entity resolution framework," *J. Amer. Soc. Inform. Sci. Tech.*, vol. 62, no. 5, pp. 919–932, May 2011.

[14] S. Lawrence et al., "Digital libraries and autonomous citation indexing," *IEEE Computer*, vol. 32, no. 6, pp. 67–71, 1999.

[15] S. Lawrence et al., "Autonomous citation matching," in *Proc. of the 3rd Annu. Conf. on Autonomous Agents.* Seattle, USA: ACM, New York, NY, USA, May 1999, pp. 392–393.

[16] M. Ley and P. Reuther, "Maintaining an online bibliographical database: The problem of data quality," in *Proc. of the Extraction et Gestion des Connaissances (EGC)*, Lille, France, 2006, pp. 5–10.

[17] A. A. Ferreira et al., "A brief survey of automatic methods for author name disambiguation," *SIGMOD Record*, vol. 41, no. 2, pp. 15–26, 2012.

[18] J. W. Warner and E. W. Brown, "Automated name authority control," in *Proc. of the 1st ACM/IEEE-CS Joint Conf. on Digital Libraries*, Roanoke, USA, June 2001, pp. 21–22.

[19] P. T. Davis et al., "Methods for precise named entity matching in digital collections," in *Proc. of the 3rd ACM/IEEE-CS Joint Conf. on Digital Libraries*, Houston, USA, May 2003, pp. 125–127.

[20] O. Benjelloun et al., "Swoosh: A generic approach to entity resolution," *The VLDB Journal*, vol. 18, no. 1, pp. 255–276, March 2009.

[21] E. Ioannou et al., "On Generating Benchmark Data for Entity Matching," *J. Data Semantics*, vol. 2, no. 1, pp. 37–56, 2013. [Online]. Available: http://dx.doi.org/10.1007/s13740-012-0015-8

[22] A. K. Elmagarmid et al., "Duplicate record detection: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 19,

no. 1, pp. 1–16, 2007.

[23] H. Köpcke and E. Rahm, "Frameworks for entity matching: A comparison," *Data & Knowl. Eng.*, vol. 69, no. 2, pp. 197–210, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2009.10.003

[24] N. Koudas et al., "Record linkage: Similarity measures and algorithms," in *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*. Chicago, USA: ACM New York, NY, USA, June 2006, pp. 802–803.

[25] L. Getoor and A. Machanavajjhala, "Entity resolution: Theory, practice and open challenges," *Proc. VLDB Endowment*, vol. 5, no. 12, pp. 2018–2019, 2012, tutorial available at http://www.cs.umd.edu/~getoor/Tutorials/ER_VLDB2012.pdf.

[26] A. Veloso et al., "Cost-effective on-demand associative author name disambiguation," *Inform. Process. Manage.*, vol. 48, no. 4, pp. 680–697, 2012.

[27] A. Veloso et al., "Multi-evidence, multi-criteria, lazy associative document classification," in *Proc. of the 15th ACM Int. Conf. on Information and Knowledge Management*. Arlington, Virginia, USA: ACM, New York, NY, USA, 20, pp. 218–227.

[28] A. H. F. Laender et al., "A brief survey of web data extraction tools," *SIGMOD Rec.*, vol. 31, no. 2, pp. 84–93, 2002.

[29] I. H. Witten et al., *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

[30] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, USA: Springer-Velag, 1995.

[31] A. H. F. Laender et al., "Assessing the research and education quality of the top brazilian computer science graduate programs," *ACM SIGCSE Bulletin*, vol. 4, no. 2, pp. 135–145, 2008.