RESEARCH ARTICLE

Author name disambiguation of bibliometric data: A comparison of several unsupervised approaches[1]

Alexander Tekles[1,2] and **Lutz Bornmann**[1]

[1]Division for Science and Innovation Studies, Administrative Headquarters of the Max Planck Society, Hofgartenstr. 8, 80539 Munich, Germany

[2]Ludwig-Maximilians-Universität Munich, Department of Sociology, Konradstr. 6, 80801 Munich, Germany

Keywords: author name disambiguation, bibliometrics, unsupervised disambiguation approach

ABSTRACT

Adequately disambiguating author names in bibliometric databases is a precondition for conducting reliable analyses at the author level. In the case of bibliometric studies that include many researchers, it is not possible to disambiguate each single researcher manually. Several approaches have been proposed for author name disambiguation, but there has not yet been a comparison of them under controlled conditions. In this study, we compare a set of unsupervised disambiguation approaches. Unsupervised approaches specify a model to assess the similarity of author mentions *a priori* instead of training a model with labeled data. To evaluate the approaches, we applied them to a set of author mentions annotated with a ResearcherID, this being an author identifier maintained by the researchers themselves. Apart from comparing the overall performance, we take a more detailed look at the role of the parametrization of the approaches and analyze the dependence of the results on the complexity of the disambiguation task. Furthermore, we examine which effects the differences in the set of metadata considered by the different approaches have on the disambiguation results. In the context of this study, the approach proposed by Caron and van Eck (2014) produced the best results.

1. INTRODUCTION

Bibliometric analyses of individuals require adequate authorship identification. For example, Clarivate Analytics annually publishes the names of highly cited researchers who have published the most papers belonging to the 1% most highly cited in their subject categories (see https://clarivate.com/webofsciencegroup/solutions/researcher-recognition/). The reliable attribution of papers to corresponding researchers is an absolute necessity for publishing this list of researchers. Empirical studies also showed that poorly disambiguated data may distort the results of analyses referring to the author level (Kim, 2019; Kim & Diesner, 2016). Some identifiers that uniquely represent authors are available in bibliometric databases. These are maintained by the researchers themselves (e.g., ResearcherID, ORCID)—implying a low coverage—or are based on an undisclosed automatic assignment (e.g., Scopus Author ID)—which does not allow an assessment of the quality of the algorithm (the algorithm is not publicly available). Publicly

[1] This is an extended version of a study which has been presented at the 17th International Society of Scientometrics and Informetrics Conference (ISSI), September 2–5, 2019, in Rome.

an open access journal

Check for updates

Citation: Tekles, A., & Bornmann, L. (2020). Author name disambiguation of bibliometric data: A comparison of several unsupervised approaches. *Quantitative Science Studies*, *1*(4), 1510–1528. https://doi.org/10.1162/qss_a_00081

DOI: https://doi.org/10.1162/qss_a_00081

Received: 19 June 2019
Accepted: 28 May 2020

Corresponding Author:
Alexander Tekles
alexander.tekles.extern@gv.mpg.de

Handling Editor:
Vincent Larivière

The MIT Press

Table 1. Examples of homonyms and synonyms in bibliometric databases

Publication title	Author name	Author ID
Social theory and social structure	R. Merton	1
The Matthew effect in science	Robert Merton	1
Allocating Shareholder Capital to Pension Plans	Robert Merton	2

available approaches that try to solve the task of disambiguating author names have thus been proposed in bibliometrics. This task presents a nontrivial challenge, as different authors may have the same name (homonyms) and one author may publish under different names (synonyms).

Table 1 shows the titles, the author names and an author identifier for three publications, including both homonyms and synonyms. The author names of the first two publications are synonyms because they refer to the same person but differ in terms of the name. The author names of the last two publications are an example of homonyms because they refer to different persons but share the same name.

Although different disambiguation approaches have been developed and implemented in local bibliometric databases (e.g., Caron & van Eck, 2014), there is hardly any comparison of the approaches. However, this comparison is necessary to gain knowledge of which approaches perform best and the conditions on which the performance of the approaches depends. In this study, we compare four unsupervised disambiguation approaches. To evaluate the approaches, we applied them to a set of author mentions annotated with a ResearcherID, this being an author identifier maintained by the researchers themselves. Apart from comparing the overall performance, we take a more detailed look at the role of the parametrization of the approaches and analyze the dependence of the results on the complexity of the disambiguation task.

2. RELATED WORK

To find sets of publications corresponding to real-world authors, approaches for disambiguating author names try to assess the similarity between author mentions by exploiting metadata such as coauthors, subject categories, and journal. To reduce runtime complexity and exclude a high number of obvious false links between author mentions, most approaches reduce the search space by blocking the data in a first step (On, Lee, et al., 2005). The idea is to generate disjunctive blocks so that author mentions in different blocks are very likely to refer to different identities, and therefore the comparisons can be limited to pairs of author mentions within the same block (Levin, Krawczyk, et al., 2012; Newcombe, 1967). A widely used blocking strategy for disambiguating author names in bibliometric databases is to group together all author mentions with an identical canonical representation of the author name, consisting of the first name initial and the surname (On et al., 2005; see also section 4.1).

The algorithms to disambiguate author names that have been proposed up to now differ in several respects (Ferreira, Gonçalves, & Laender, 2012). One way to distinguish between different approaches is to classify them as either unsupervised or supervised (Smalheiser & Torvik, 2009). Supervised approaches try to train the parameters of a specified model with the help of certain training data (e.g., Ferreira, Veloso, et al., 2010, 2014; Levin et al., 2012; Torvik & Smalheiser, 2009). The training data contains explicit information as to which author mentions belong to the same identity and which do not. The model trained on the basis of this data is then used to

detect relevant patterns in the rest of the data. Unsupervised approaches, on the other hand, try to assess the similarity of author mentions by explicitly specifying a similarity function based on the author mentions' attributes. Supervised approaches entail several problems, especially the challenge of providing adequate, reliable, and representative training data (Smalheiser & Torvik, 2009). Therefore, we focus on unsupervised approaches in the following.

The unsupervised approaches for disambiguating author names that have been proposed so far vary in several ways. First, every approach specifies a set of attributes and how these are combined to provide a similarity measure between author mentions. Second, to determine which similarities are high enough to consider two author mentions or two groups of author mentions as referring to the same author, some form of threshold for the similarity measure is necessary. This threshold can be determined globally for all pairs of author mentions being compared, or it can vary depending on the number of author mentions within a block that refer to a single name representation. Block-size-dependent thresholds try to reduce the problem of an increasing number of false links for a higher number of comparisons between author mentions; that is, for larger name blocks (Backes, 2018a; Caron & van Eck, 2014).

Third, the approaches differ with regard to the clustering strategy that is applied, that is, how similar author mentions are grouped together. All clustering strategies used so far in the context of author name disambiguation can be regarded as agglomerative clustering algorithms (Ferreira et al., 2012), especially in the form of single-link or average-link clustering. More specifically, single-link approaches define the similarity of two clusters of author mentions as the maximum similarity of all pairs of author mentions belonging to the different clusters. The idea behind this technique is that each of an author's publications is similar to at least one of his or her other publications. In average-link approaches, on the other hand, the two clusters with the highest overall cohesion are merged in each step; that is, all objects in the clusters are considered (in contrast to just one from each cluster in single-link approaches). This rests on the assumption that an author's publications form a cohesive entity. As a consequence, it is easier to distinguish between two authors with slightly different oeuvres compared to single-link approaches, but heterogeneous oeuvres by a single author are more likely to be split.

Previous author name disambiguation approaches have usually been evaluated in terms of their quality. This evaluation is always based on measuring how pure the detected clusters are with respect to real-world authors (precision) and how well the author mentions of real-world authors are merged in the detected clusters (recall). However, different metrics have been applied when assessing these properties. Furthermore, different data sets have been used to evaluate author name disambiguation approaches (Kim, 2018). It is therefore difficult to compare the different approaches based on the existing evaluations.

3. APPROACHES COMPARED

We focused on unsupervised disambiguation approaches in our analyses (see above). As these approaches require no training data to be provided *a priori*, they are more convenient for use with real-world applications. We investigated four elaborated approaches in addition to two naïve approaches, which only consider the author names (a) in the form of the canonical representation of author names used for the initial blocking of author mentions (first initial of the first name and the surname; see also section 4.1), and (b) in the form of all first name initials and the surname. These approaches were selected to cover a wide variety of features that characterize unsupervised approaches for disambiguating author names. We applied the approaches to data from the Web of Science (WoS, Clarivate Analytics) that had already been preprocessed according to a blocking strategy, as described in section 4.1.

3.1. Implementation of the Four Selected Disambiguation Approaches

In the following, the four disambiguation approaches that we investigated in this study are explained.

Cota, Gonçalves, and Laender (2007) proposed a two-step approach that considers the names of coauthors, publication titles, and journal titles. In a first step, all pairs of author mentions that share a coauthor name are linked. The linked author mentions are then clustered by finding the connected components with regard to this matching. The second step iteratively merges these clusters if they are sufficiently similar with respect to their publication or journal titles. The similarity of two clusters (one for publication titles, one for journal titles) is defined as the cosine similarity of the two term frequency-inverse document frequencies (TF-IDFs) for the clusters' publication titles (or journal titles). Two clusters are merged if one of their similarities (with either regard to publication or to journal titles) exceeds a predefined threshold. This process continues until there are no more sufficiently similar clusters to merge, or until all author mentions are merged into one cluster.

Schulz, Mazloumian, et al. (2014) proposed a three-step approach based on the following metric for the similarity s_{ij} between two author mentions i and j:

$$s_{ij} = \alpha_A\left(\frac{|A_i \cap A_j|}{\min(|A_i|, |A_j|)}\right) + \alpha_S(|p_i \cap R_j| + |p_j \cap R_i|) + \alpha_R(|R_i \cap R_j|) + \alpha_C\left(\frac{|C_i \cap C_j|}{\min(|C_i|, |C_j|)}\right) \quad (1)$$

Here, A_i denotes the coauthor list of paper i, R_i its reference list, and C_i its set of citing papers. The first step links all pairs of author mentions with a similarity (determined by Eq. 1) exceeding a threshold β_1. A set of clusters is determined by finding the corresponding connected components. In the second step, these clusters are merged in a very similar way as in the first step. To determine the similarity $S_{\gamma\kappa}$ of two clusters γ and κ, the similarities between author mentions within these clusters are combined by means of the following formula:

$$S_{\gamma\kappa} = \sum_{i\in\gamma j\in\kappa} \frac{s_{ij}\Theta(s_{ij})}{|\gamma||\kappa|}, \quad \Theta(s_{ij}) = \begin{cases} 1 & if \ \ s_{ij} > \beta_2 \\ 0 & if \ \ s_{ij} \le \beta_2 \end{cases} \quad (2)$$

Here, $|\gamma|$ denotes the number of author mentions in cluster γ (similarly for cluster κ). As the formula shows, only those similarities between author mentions that exceed a threshold β_2 are considered when calculating the similarity between two clusters. As in the first step, this cluster similarity is used to link clusters if they exceed another threshold β_3 to find the corresponding connected components. The third step of this approach finally adds single author mentions that have not been merged to a cluster in either of the first two steps, provided its similarity with one of the cluster's author mentions exceeds a threshold β_4.

Caron and van Eck (2014) proposed measuring the similarity between two author mentions based on a set of rules that rely on several paper-level and author-level attributes. More precisely, a score is specified for each rule, and all of the scores for matching rules are added up to an overall similarity score for the two author mentions (see Table 2). If two author mentions are sufficiently similar with regard to this similarity score, they are linked and the corresponding connected components are considered oeuvres of real-world authors. The threshold for determining whether two author mentions are sufficiently similar depends on the size of the corresponding name block. The idea behind this approach is to take into account the higher

Table 2. Rules for rule-based scoring proposed by Caron and van Eck (2014)

Field	Criterion	Score
Email	exact match	100
Number of shared initials	2 / > 2 / conflicting initials	5 / 10 / −10
Shared first name	general name / nongeneral name	3 / 6
Address (linked to author)	matching country and city	4
Number of shared coauthors	1 / 2 / > 2	4 / 7 / 10
Grant number	at least one shared grant number	10
Address (linked to publication, but not linked to author)	matching country and city	2
Subject category	matching subject category	3
Journal	matching journal	6
Self-citation	one publication citing the other	10
Bibliographic coupling: number of shared cited references	1 / 2 / 3 / 4 / > 4	2 / 4 / 6 / 8 / 10
Co-citation: number of shared citing papers	1 / 2 / 3 / 4 / > 4	2 / 3 / 4 / 5 / 6

risk of false links in larger blocks. Higher thresholds are therefore used for larger blocks to reduce the risk of incorrectly linked author mentions.

Backes (2018a) proposed an approach that starts by considering each author mention as one cluster. An agglomerative clustering algorithm is then employed that iteratively merges clusters (starting with single author mentions as clusters, then merging clusters of several author mentions) if they are sufficiently similar; that is, two clusters are connected if their similarity exceeds a quality limit l. The similarity metric indicating how similar two clusters are takes into account the specificity of the author mentions' metadata. For example, if two author mentions share a very rare subject category this might be a strong indicator that the author mentions refer to the same author, while this is not true for a very common subject category. This strategy is applied to compute a similarity score for each attribute under consideration. The similarity score $p_a(C|\dot{C})$ for an attribute a and two clusters C, $\dot{C}$ is defined as

$$p_a(C|\dot{C}) = \sum\nolimits_{(x,\dot{x})\in C\times\dot{C}} p(x|\dot{x}) \cdot \frac{\#(\dot{x}) + \varepsilon}{\#(\dot{C}) + |C| \cdot \varepsilon} \tag{3}$$

with $p(x|\dot{x}) = \frac{1}{\#(\dot{x})+\varepsilon} \cdot \left(\sum_{f\in F} \frac{\#(f,x)\cdot\#(f,\dot{x})}{\#(f)} + \frac{\varepsilon}{|\dot{X}|}\right)$

F = set of all features for attribute a

$\#(f, x)$ = number of occurences of feature f for author mention x

$\#(x) = \sum_{f\in F} \#(f, x)$

$\#(C) = \sum_{x\in C} \#(x)$

$\#(f) = \sum_{x\in X} \#(f, x)$

$|X|$ = number of author mentions in the name block containing x and $\dot{x}$

ε = smoothing parameter to prevent division by zero.

When using this approach in our study, we considered the following attributes: titles, abstracts, affiliations, subject categories, keywords, coauthor names, author names of cited references, and email addresses. Backes (2018a) proposed several variants to combine these scores into a final similarity score of two clusters. In the variant implemented in this study, the scores are combined in the form of a linear combination with equal weights for all attributes' scores. This allows including attributes flexibly without the necessity to specify the corresponding weights separately. The results reported in Backes (2018a) suggest that using equal weights for all attributes produces good results. Each iteration of the clustering process merges all pairs of current clusters whose similarity exceeds l. The quality limit l is designed to have a linear dependence on the block size $|X|$, whereby the parameter λ specifies this relationship (see Eq. 4).

$$l = \lambda \cdot |X| \tag{4}$$

Several other unsupervised approaches for disambiguating author names have been proposed besides the four aforementioned approaches (e.g., Hussain & Asghar, 2018; Liu, Li, et al., 2015; H. Wu, Li, et al., 2014; J. Wu & Ding, 2013; Zhu, Wu, et al., 2017). Overviews of these approaches have been published by Ferreira et al. (2012) and Hussain and Asghar (2017). Our selection of the approaches aims at considering a wide range of strategies that can be applied for unsupervised author name disambiguation: using few versus using many attributes, using block-size-dependent versus using block-size-independent thresholds, and calculating similarity metrics based on various attributes versus merging author mentions based on one attribute at a time.

Besides the four approaches, we also included two naïve approaches that only use author names for the disambiguation. The first naïve approach uses the name blocks as the disambiguation result. This allows us to assess how much the elaborate approaches improve the disambiguation quality as compared to the blocking step alone. The second naïve approach only uses all initials of the first names and the surname for the disambiguation. This very simple approach has been widely used (Milojević, 2013) and seems to perform relatively well according to empirical analyses (Backes, 2018b). Including this approach in our analyses allows us to judge whether the additional effort associated with the more elaborate approaches is worthwhile with regard to the improvement in the disambiguation quality.

3.2. Parameter Specification

Some form of threshold (or a set of thresholds) must be specified for each of the four approaches. As such thresholds have not been proposed for all approaches by the authors, and some of the proposed thresholds produce poor results for our data set, we fitted them with regard to our data. This allows better comparability because the thresholds are matched to the particular data they are applied to. Our procedures for specifying the thresholds maximize the metrics $F1_{pair}$ and $F1_{best}$ (see below) that we used for the evaluation of the approaches. In our analyses, this is primarily a means for evaluating the approaches independently of the particular thresholds used, as the results reflect how good the approaches are instead of how well the thresholds are chosen. In practical applications, this would only be possible if a sufficiently large amount of the data is already reliably disambiguated (which is usually not the case though).

We specified a procedure for each of the approaches that allowed an efficient consideration of a wide range of thresholds. A set of thresholds uniformly distributed over the complete parameter space was chosen as a candidate set for the approach of Cota et al. (2007). We also specified the thresholds for the approach of Schulz et al. (2014) by evaluating a candidate set of parameters;

in this case, the candidate set of thresholds was chosen on the basis of the parameters proposed in the original paper. The parametrization of this approach was further optimized by fitting β_1, β_2, and β_3 independently from β_4. β_4 was subsequently chosen based only on the best combination of the other thresholds, which substantially reduces the search space. We believe this to be an adequate procedure for finding the thresholds because the last step of this disambiguation approach (which is based on β_4) has only a minor influence on the final result. For the approach proposed by Caron and van Eck (2014) we initially had to define the block size classes that divide the blocks into several classes with regard to the internal number of author mentions. Similar to Caron and van Eck (2014), we defined six block size classes. Our specification of the classes aims at reducing the variance of optimal thresholds within a class and is based on a manual inspection of the distribution of optimal thresholds across block sizes. Then the best possible threshold for each class (maximizing $F1_{pair}$ and $F1_{best}$) is chosen.

For the approach of Backes (2018a), we had to modify the approach slightly to define a feasible procedure for fitting the parameter λ, which determines the quality limit l for a given block. Instead of linking all pairs of clusters whose similarity exceeds a given l in each iteration, we iteratively merged only those pairs of clusters whose similarity equals the maximum similarity of all current pairs of clusters (the clusters are recomputed after each merger). These similarities were taken as estimates for the quality limit that would yield the clustering of the corresponding merger step. This modification may produce results that are different to the original approach, because the order in which the author mentions are merged may change and the similarities between clusters depend on the previous mergers. However, we assume that these changes produce only minor differences that do not influence any general conclusions on the approach. Our implementation merges the most similar clusters in each iteration; that is, the most reliable mergers are applied iteratively until the quality limit is reached. Correspondingly, the original approach follows the idea that all cluster similarities exceeding a certain quality limit indicate reliable links between the corresponding clusters.

4. METHOD

We collected metadata for a subset of author mentions from the WoS for our analyses. To provide a gold standard that represents sets of author mentions corresponding to real-world authors, we only took author mentions with a ResearcherID linked to their publications in the WoS into account. More specifically, all person records that are marked as authors and that have a ResearcherID linked to at least one paper published in 2015 or later have been considered. It is very likely that this procedure excludes all author mentions with ResearcherIDs referring to nonauthor entities (e.g., organizations) and takes into account only such ResearcherIDs that have been maintained recently.

For an increasing number of author mentions, it can be expected that the quality of their disambiguation decreases (see also section 5). Our results would thus not be transferable to application scenarios with a larger number of author mentions than in our data set. At the same time, the limitation on a subset of author mentions from the WoS seems appropriate, because the same data is used for all approaches. This allows comparing the approaches under controlled conditions. Furthermore, our analyses allow an assessment of the relationship between the complexity of the disambiguation task (in terms of name block size) and the quality of the results produced (see section 5). This gives an idea of how well the approaches perform for an increasing amount of data. As including more author mentions in our data would drastically increase the computational costs, we refrained from including more author mentions than those annotated with a ResearcherID.

4.1. Blocking

Blocking author mentions based on authors' names is usually the first step in the disambiguation process. While different strategies have been proposed for this blocking step, they all aim at narrowing down the search space for the subsequent disambiguation task in a reliable and efficient way. For this purpose, a canonical representation of the author name is specified and all author mentions with identical name representation are assigned to the same block.

As this procedure only considers author names and is based on exact matches, it requires less computational resources compared to the subsequent steps of the disambiguation process. These subsequent steps can be applied then to smaller sets of author mentions. Because the computational complexity of the disambiguation approaches considered in our study is super-linear in the number of author mentions, the overall complexity can be reduced by splitting up the disambiguation in smaller tasks. A smaller number of author mentions also reduces the risk of making false links between author mentions, which improves the quality of the disambiguation results.

While reducing the block sizes, the blocking strategy at the same time needs to be reliable in the sense that for an author, a canonical name representation is very likely to include all of her or his author mentions. To achieve both goals, an adequate level of specificity of the canonical name representation used for blocking the author mentions is necessary. Using a general name representation (e.g., the first initial of the first names and the full surname) results in relatively large blocks. The number of splitting errors is rather small in these blocks, but the computational complexity of the subsequent steps in the disambiguation process is rather high. In contrast, using a specific name representation (e.g., all initials of the first names and the full surname) results in smaller blocks. Although the number of splitting errors in these blocks increases due to synonyms, the computational complexity of the subsequent steps is reduced in the disambiguation process. Empirical analyses assessing the errors introduced by different blocking schemes can be found in Backes (2018b). These analyses show that a general name representation based on the first initial of the first names and the full surname produces good results, especially with regard to recall. They also show that using all initials of the first names and the full surname produces good results in terms of F1 (see section 4.2). These results qualify the blocking scheme based on all initials of the first names and the full surname as a simple disambiguation approach without any subsequent steps. However, compared to using only the first initial and the surname, blocking the author mentions based on all initials of the first names and the full surname introduces additional splitting errors. These splitting errors introduced by the blocking step are of particular importance for subsequent steps, because they cannot be corrected later in the disambiguation process.

For the blocking step in our analyses, we used the first initial of the first names and the full surname as the canonical name representation. One reason for this choice is that this name representation has been used by many other studies related to author name disambiguation (Milojević, 2013). A second reason is that this is a very general blocking scheme, which reduces the risk of making splitting errors in the blocking step (Backes, 2018b). For a practical application with a large amount of data, this might not be feasible, because the general blocking scheme produces large blocks (Backes, 2018b). However, for our purpose of evaluating different approaches building upon the blocked author mentions, using a general blocking scheme allows us to focus on these subsequent steps. Due to the high recall, the upper bound for the disambiguation quality that can be achieved by the approaches is not reduced considerably by the blocking step, and the final result is more dependent on the subsequent steps rather than the blocking step. The small risk of making splitting errors due to this blocking scheme is also visible in our results (see Table 3).

Table 3. Overall results for all approaches

Approach	P_{pair}	R_{pair}	$F1_{pair}$	P_{best}	R_{best}	$F1_{best}$
Baseline (first initial)	0.095	0.998	0.173	0.322	0.999	0.487
Baseline (all initials)	0.210	0.854	0.338	0.603	0.905	0.724
Cota et al. (2007)	0.111	0.857	0.196	0.442	0.912	0.595
Schulz et al. (2014)	0.453	0.456	0.455	0.799	0.749	0.773
Caron and van Eck (2014)	0.831	0.785	0.808	0.916	0.884	0.900
Backes (2018a)	0.674	0.620	0.646	0.761	0.698	0.728

In our analyses, we only considered name blocks comprising at least five real-world authors. This selection allowed us to focus on rather difficult cases where the author mentions in a block actually have to be disambiguated across several authors. All in all, this data collection procedure results in 1,057,978 author mentions distributed over 2,484 name blocks and 29,244 distinct ResearcherIDs. The largest name block ("y. wang") comprises 7,296 author mentions.

4.2. Evaluation Metrics

The evaluation of author name disambiguation approaches is generally based on assessing their ability to discriminate between author mentions of different real-world authors (precision) and their ability to merge author mentions of the same real-world author (recall). Even though these concepts are widely accepted and referenced, various specific evaluation metrics have been used in the past. In the following, we focus on two types of evaluation metrics. First, we calculate pairwise precision (P_{pair}), pairwise recall (R_{pair}), and pairwise F1 ($F1_{pair}$) for each approach. These metrics have been used in many studies (e.g., Backes, 2018a; Caron & van Eck, 2014; Levin et al., 2012). Whereas pairwise precision measures how many links between author mentions in detected clusters are correct, pairwise recall measures how many links between author mentions of real-world authors are correctly detected. Pairwise F1 is the harmonic mean of these two metrics. Eqs. (5)–(7) provide a formal definition of these evaluation metrics, using the following notation:

- $|pairs_{author}|$ denotes the number of all pairs of author mentions where both author mentions refer to the same author;
- $|pairs_{cluster}|$ denotes the number of pairs of author mentions where both author mentions are assigned to the same cluster by the disambiguation algorithm; and
- $|pairs_{author} \cap pairs_{cluster}|$ denotes the number of author mentions where both author mentions refer to the same author and are assigned to the same cluster.

$$P_{pair} = \frac{|pairs_{author} \cap pairs_{cluster}|}{|pairs_{cluster}|} \tag{5}$$

$$R_{pair} = \frac{|pairs_{author} \cap pairs_{cluster}|}{|pairs_{author}|} \tag{6}$$

$$F1_{pair} = \frac{2P_{pair}R_{pair}}{P_{pair} + R_{pair}} \tag{7}$$

An important property of pairwise evaluation metrics is that they consider the disambiguation quality among all links between author mentions. For example, consider two clusters A and B for which the precision should be determined. Cluster A has 10 author mentions referring to one author and five author mentions to a second author. Cluster B has 10 author mentions referring to one author and five author mentions referring to different authors. These two clusters get different scores for the pairwise precision (for cluster A, $P_{pair} = \frac{55}{105} \approx 0.524$, while for cluster B, $P_{pair} = \frac{45}{105} \approx 0.429$). However, if we assign each cluster to one author, the two clusters are equally adequate: Ten author mentions are correct and five are incorrect in each case. To assess how the disambiguation approaches perform with regard to this task (and the corresponding task to find all author mentions for each author), we calculated metrics to measure how reliably a cluster can be attributed to exactly one author (best precision P_{best}) and how well an author can be attributed to exactly one cluster (best recall R_{best}). Eqs. (8)–(10) provide a formal definition of these evaluation metrics, using the following notation:

- $|author\ mentions_{best\ author}|$ is calculated as follows: for each cluster c, the maximum number of author mentions that refer to the same author $n_{c,\max\ author}$ is determined; $|author\ mentions_{best\ author}|$ is the sum of $n_{c,\max\ author}$ over all clusters.
- $|author\ mentions_{best\ cluster}|$ is calculated as follows: for each author a, the maximum number $n_{a,\max\ cluser}$ of author mentions that are assigned to the same cluster is determined; $|author\ mentions_{best\ cluster}|$ is the sum of $n_{a,\max\ cluser}$ over all authors.
- $|author\ mentions|$ denotes the number of all author mentions.

$$P_{best} = \frac{|author\ mentions_{best\ author}|}{|author\ mentions|} \tag{8}$$

$$R_{best} = \frac{|author\ mentions_{best\ cluster}|}{|author\ mentions|} \tag{9}$$

$$F1_{best} = \frac{2P_{best}R_{best}}{P_{best} + R_{best}} \tag{10}$$

An approach for evaluating the quality of author name disambiguation that is very similar to P_{best}, R_{best}, and $F1_{best}$ has been proposed by Li, Lai, et al. (2014). In this approach, splitting and lumping errors are calculated, which correspond to the notions recall and precision, respectively. However, the calculation of lumping errors does not necessarily take into account all clusters, but for each author the cluster with most of her or his author mentions. In contrast, P_{best} considers all clusters. Therefore, P_{best} is better suited to assess how reliable it is to take each cluster as one author given the disambiguated data (see also Torvik & Smalheiser, 2009 for a discussion of different perspectives for evaluating author name disambiguation). Furthermore, P_{best}, R_{best}, and $F1_{best}$ are better comparable with the pairwise evaluation metrics, because both types of metrics follow the precision-recall-F1 terminology and have the same scale. Another type of evaluation metrics that are very similar to P_{best}, R_{best}, and $F1_{best}$ are the closest cluster precision, closest cluster recall, and closest cluster F1 (Menestrina, Whang, & Garcia-Molina, 2010). These metrics are based on the Jaccard similarities between clusters and authors[2]. The closest cluster precision is the average maximum Jaccard similarity over all clusters. By using the maximum Jaccard similarities for each cluster,

[2] The Jaccard similarity $J(a, c)$ between author a and cluster c is defined as $\frac{number\ of\ author\ mentions\ in\ c\ and\ a}{number\ of\ author\ mentions\ in\ c\ or\ a}$.

this approach is very similar to the idea that P_{best} is based on: For each cluster, only the author with the most author mentions in this cluster is taken into account[3]. However, in contrast to P_{best}, a closest cluster precision < 1 is possible if each cluster only contains author mentions of one author. When considering such a cluster as the oeuvre of one author, the precision should be 1 though: All author mentions in this cluster are correct (all author mentions refer to the same author, that is, the cluster is perfectly precise). Therefore, we decided to use P_{best}, R_{best}, and $F1_{best}$ as defined in Eqs. (8)–(10) for evaluating the disambiguation approaches in this study.

Each of Eqs. (5)–(10) can be applied either to the complete data set or to a subset of author mentions. For example, the results of one name block can be evaluated by only considering author mentions within this block when computing the evaluation metrics. All metrics can take values between 0 and 1, with higher values indicating a better disambiguation result.

5. RESULTS

5.1. Overall Results

The results for the approaches described in section 3 are summarized in Table 3. The table shows the evaluation metrics described in the previous section for each approach. All the approaches produced better results than the naïve baseline disambiguation based on first initial and surname; only three of the approaches produced better results than the baseline disambiguation based on all initials and surname. The approach proposed by Caron and van Eck (2014) performs best among the examined approaches with regard to both $F1_{pair}$ and $F1_{best}$. If one compares the approaches of Schulz et al. (2014) and Backes (2018a), the two evaluation metrics yield different rankings. Whereas the latter approach performs better with regard to $F1_{pair}$, the former performs better with regard to $F1_{best}$. Both of these approaches perform only slightly better than the baseline based on all initials. This might suggest that a simple approach based only on author names performs nearly as well as these approaches. However, the precision of the all-initials baseline is very small compared to the approaches of Schulz et al. (2014) and Backes (2018a). The all-initials baseline and the two approaches also differ in the variance of the disambiguation quality across block sizes (see Figure 1). This means that the approaches perform better or worse depending on the given data and the preferences regarding the trade-off between precision and recall. The approach of Cota et al. (2007) performs worse than the all-initials baseline, and only slightly better than the first-initial baseline. The precision in particular is very small for the approach of Cota et al. (2007), mainly due to a high number of false links between author mentions in the first step (merging author mentions with shared coauthors).

Figure 1 shows the distribution of the disambiguation quality over block sizes, using thresholds as described in section 3.2. The lines represent nonparametric regression estimates (calculated using the `loess()` function in the base package of R), with evaluation metrics as dependent variable and block size as independent variable. In addition to these regression estimates, the results for single blocks are plotted for large block sizes. As there are too many small blocks to adequately recognize the relationship between block length and evaluation metrics, results at the block level are only displayed for large blocks.

The results reveal that the disambiguation quality in terms of the F1 metrics varies strongly across name blocks. In particular, the F1 values decrease for large blocks. Therefore, the disambiguation process may produce biases with regard to the frequency of the corresponding name representation. One reason for the dependence of the disambiguation quality on the size of the name block is the larger search space to find clusters of author mentions. The larger search space increases the

[3] The closest cluster recall is calculated accordingly by changing the perspective from clusters to authors.

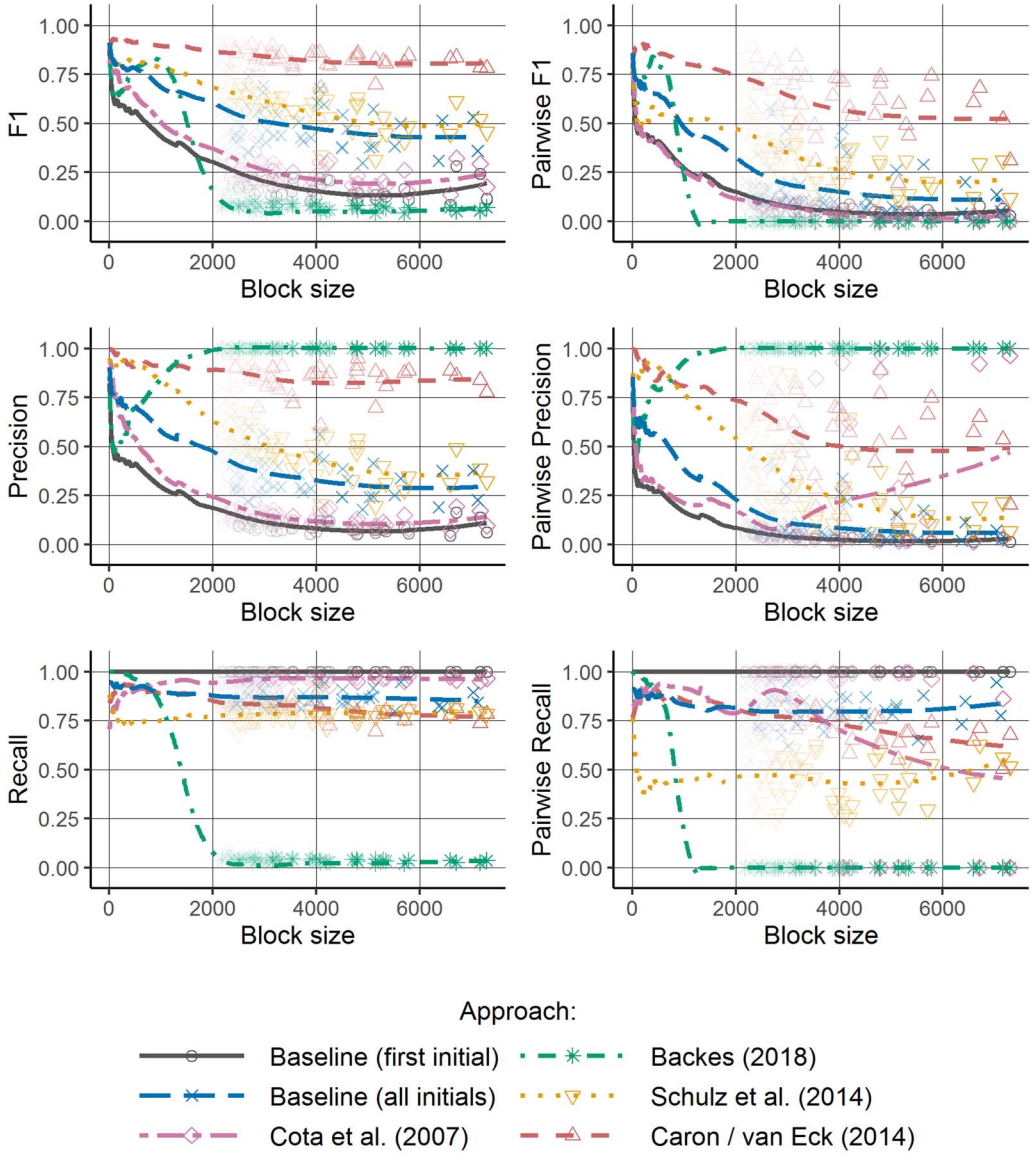

Figure 1. F1, precision, and recall values for all approaches across block sizes using thresholds as originally proposed by the authors. The lines show nonparametric regression estimates based on all blocks and the points show results for single blocks (only results for large blocks are displayed this way).

search complexity in general, implying a greater potential for false links between author mentions. Some approaches try to reduce this problem by allowing block size-dependent thresholds (see the next section). Even though the negative relationship between block size and disambiguation quality can be observed for all approaches, the decline in quality is not equal. Especially for the approach of Caron and van Eck (2014), the influence of the block size is relatively small.

Besides the scores for the F1 metrics, Figure 1 also shows the distribution of (pairwise) precision and recall values across block sizes. According to these results, the approach of Caron and van Eck (2014) favors precision over recall, even for large blocks. The approach of Backes (2018a) scores very high on the precision metrics, but very low on the recall metrics for large blocks. This suggests that the specification of thresholds only works for small blocks in this

case (see the next section). The other approaches produce results with rather small precision for large blocks, while their recall values are relatively high.

5.2. The Influence of Parametrization on the Disambiguation Quality

Among the approaches included in our comparison, Caron and van Eck (2014) and Backes (2018a) used block-size-dependent thresholds. As described above, the first approach is based on defining one threshold for each of six block size classes, whereas the threshold is linearly dependent on the block size in the second approach. Table 4 shows the block size classes and corresponding thresholds used by our implementation for the approach of Caron and van Eck (2014). In contrast, the approaches of both Cota et al. (2007) and Schulz et al. (2014) use global thresholds for all block sizes.

To assess how much the results could be improved by allowing different thresholds for the blocks, we determined the thresholds producing the best result for each block. Figure 2 shows the evaluation results obtained by using these optimal thresholds for each single name block—instead of using the same thresholds for (a) all blocks, (b) a group of blocks, or (c) determining the thresholds based on a global rule as described in section 3.2. These results represent an upper bound for the quality over all possible thresholds if the thresholds are specified for each name block separately. The difference in the results between Figure 1 (using thresholds as originally proposed) and Figure 2 (using flexible thresholds) indicates the improvement potential for each approach by optimizing how the thresholds are specified. As the specification of flexible thresholds requires reliably disambiguated data beforehand, this strategy is not feasible in application scenarios. Flexible thresholds for each block would not greatly improve the quality of the approach proposed by Cota et al. (2007) because the results based on global thresholds are very close to the results based on completely flexible thresholds. The reason is that the quality is dominated by the first step of the approach, which does not employ any threshold at all. The second step, on the other hand, does not change the results significantly; the effect of the thresholds is rather small. In contrast, the approach of Schulz et al. (2014) benefits from using flexible thresholds, especially for large blocks.

Similar to the approach of Cota et al. (2007), the difference between the original implementation and the one with flexible thresholds is rather small for the approach of Caron and van Eck (2014). However, the original implementation already uses different thresholds based on the block size classes. As the comparison with an implementation based on a constant threshold for all block sizes shows, this improves the results. Table 5 shows the evaluation results for the approach of Caron and van Eck (2014) with three different types of thresholds: a constant threshold for all blocks ("Constant"), the thresholds of the block size classes shown in Table 4

Table 4. Block size classes and thresholds for Caron and van Eck (2014)

Block size	Threshold ($F1_{pair}$)	Threshold ($F1_{best}$)
1–500	21	19
501–1,000	22	21
1,001–2,000	25	23
2,001–3,000	27	25
3,001–4,500	29	25
>4,500	29	27

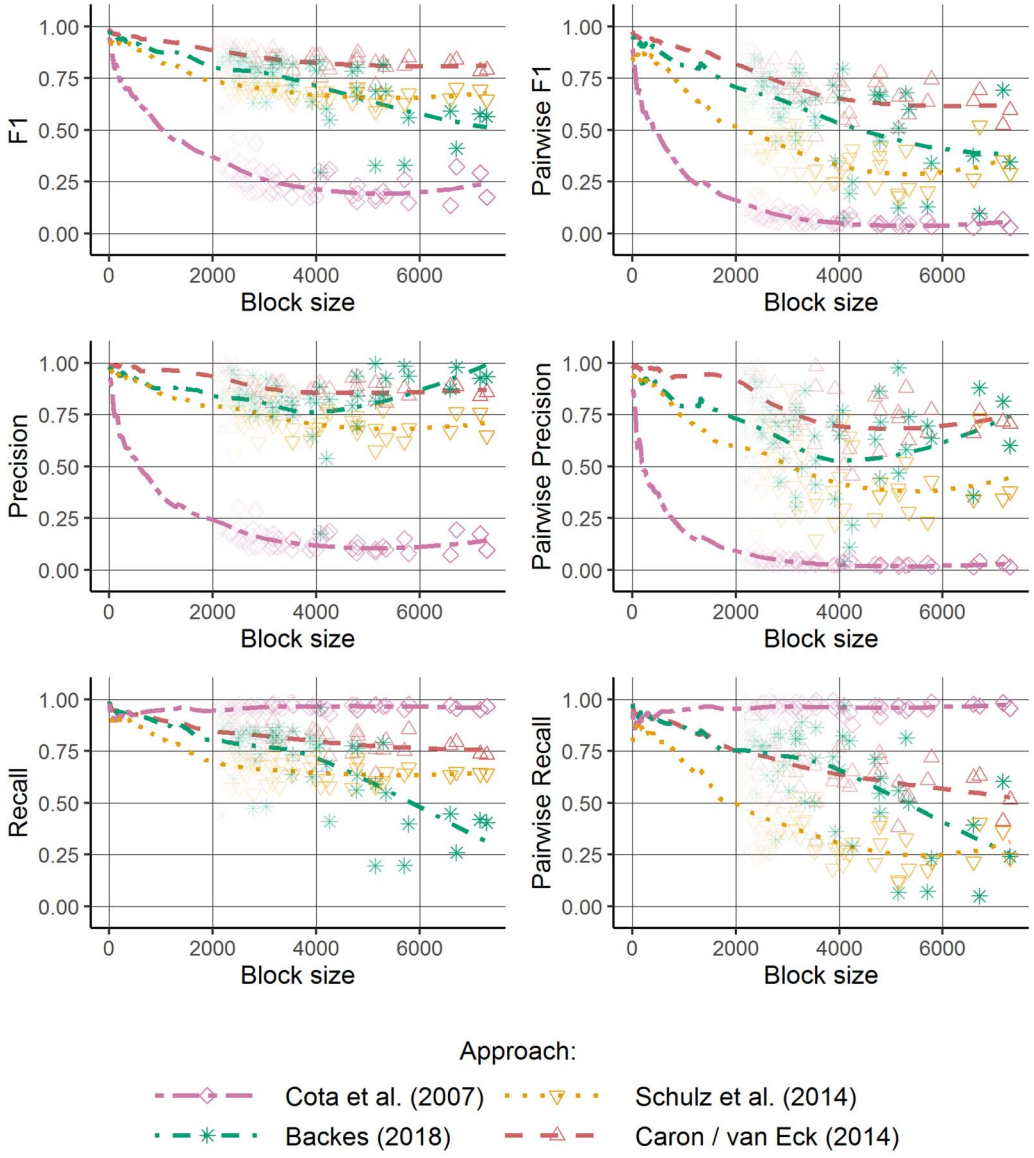

Figure 2. F1, precision, and recall values for all approaches across block sizes using flexible thresholds (the best possible threshold[s] is [are] used for each block). The lines show nonparametric regression estimates based on all blocks; the points show results for single blocks (only results for large blocks are displayed this way).

Table 5. Results for different types of thresholds for Caron and van Eck (2014)

Type of threshold	P_{pair}	R_{pair}	$F1_{pair}$	P_{best}	R_{best}	$F1_{best}$
Constant	0.690	0.741	0.714	0.879	0.880	0.880
Block size classes	0.831	0.787	0.808	0.916	0.885	0.900
Flexible	0.907	0.850	0.878	0.954	0.897	0.924

("Block size classes"), and the optimal threshold for each single block ("Flexible"). These results show that the original implementation produces better results than those obtained using a constant threshold. This means that the somewhat rough partitioning between six block size classes allows for adequate differentiation with regard to the threshold and this strategy improves the disambiguation result compared to a constant threshold over all block sizes. In contrast, the strategy of specifying a threshold which is linearly dependent on the block size, as employed by the approach of Backes (2018a), is unable to find good thresholds over the complete range of block sizes. This is due mainly to a drop in the recall (together with an increasing precision) for large blocks. The thresholds chosen by the algorithm are thus too high for large blocks. Hence, a linear relationship between block size and threshold does not appear to be an adequate strategy for large blocks. The fitted thresholds for the approach of Caron and van Eck (2014) also confirm that a nonlinear relationship between block size and threshold may be more suitable. When using flexible thresholds instead of specifying them based on a linear relationship with the block size, the results for the approach of Backes (2018a) are close (even though with more variation among large blocks) to the results for the approach of Caron and van Eck (2014). This suggests that the approach of Backes (2018a) has the potential for producing good results if adequate thresholds are specified.

The results in Figure 2 and Table 5 demonstrate that the disambiguation quality can be improved if flexible thresholds dependent on the block size are specified. However, the specification of adequate thresholds is generally a nontrivial task, as it depends on the data at hand. Likewise, the thresholds proposed previously for the approaches examined in this paper do not correspond to the thresholds fitted with regard to our data set.

5.3. The Influence of Attributes Considered for Assessing Similarities

Another important feature of disambiguation approaches is the set of the author mentions' attributes they consider for assessing the similarity between author mentions. The different quality of the disambiguated data may result from considering different sets of attributes. For example, while Caron and van Eck (2014) included the attributes listed in Table 2, Schulz et al. (2014) only considered shared coauthors, shared cited references, shared citing papers, and self-citations. As less information was considered in the latter approach, this may be a reason why Caron and van Eck (2014) is better able to detect correct links between author mentions.

To get an idea of how important the set of attributes considered by the approaches is, we compared modified versions of the three approaches producing the best results in their original versions. Using a subset of the originally proposed attributes for an approach is generally possible, simply by including these attributes as before and omitting the other attributes. However, it is not always similarly easy to include new attributes. The approach of Backes (2018a) is very flexible in this regard, because attributes (e.g., journal or subject) are weighted equally, and features (e.g., *Nature* or *Science* for the attribute "journal") are weighted automatically. Both types of weights could be easily applied to new attributes. In contrast, Schulz et al. (2014) and Caron and van Eck (2014) provide specific weights for each attribute. For these two approaches, it is not specified how new attributes can be weighted for calculating the similarity between author mentions, making them less flexible for the consideration of new attributes in the disambiguation process.

For our comparison, we disambiguated the data with the approach proposed by Caron and van Eck (2014) once more, but this time based on a reduced set of attributes, such that it corresponds to the attributes considered in the approach of Schulz et al. (2014). Furthermore, we disambiguated the data another two times with the approach proposed by Backes (2018a): in one case based on attributes similar to those considered by Schulz et al.

(2014), in the other case based on attributes similar to those considered by Caron and van Eck (2014). In these two cases, the sets of attributes are not exactly the same, because self-citations cannot be included in the approach of Backes (2018a) in the same way as in the other two approaches. In the approach of Backes (2018a), similarities are calculated based on the features that two author mentions have in common for the same attributes.

For example, if the author names for cited references of two author mentions are represented by $R_1 = \{r_{11}, r_{12}, r_{13}, r_{14}\}$ and $R_2 = \{r_{21}, r_{22}, r_{23}\}$, respectively, the approach could consider the names occurring in both R_1 and R_2 for determining the similarity of the two author mentions. However, self-citations can only be detected by comparing the author names of cited references of one author mention with the name of the author itself of the second author mention. Such a comparison between two different attributes (here: author name and author names of cited references) is not intended in the original approach. There are no shared self-citations and the specificity of self-citations cannot be captured with the framework introduced by Backes (2018a) for calculating similarities between clusters of author mentions (we refrained from modifying this framework, which may be a possibility to include self-citations).

To keep the attribute sets comparable and still include self-citations in the approaches of Schulz et al. (2014) and Caron and van Eck (2014), we used information as close as possible in the approach of Backes (2018a) by including referenced author names instead of self-citations. We consider this choice to be appropriate. In the case that two of an author's mentions have self-citations to a third author mention of the same author, these mentions would also occur as shared referenced authors. Vice versa, if two author mentions share referenced authors, it is likely that self-citations are among these, because self-citations are usually overrepresented among cited references. An alternative to this choice of attribute sets would be to exclude self-citations and author names of cited references. However, our analyses show that these two alternatives (with or without referenced authors and self-citations) produce similar results, and the conclusions are the same for both alternatives.

For each comparison and each approach, we separately specified the thresholds as described in section 3.2. The results of the outlined implementations are summarized in Table 6. The results show that differences between the approaches still exist. Characteristics of the approaches other than the set of attributes are therefore also relevant for the quality of an algorithm. In our analyses, the approach of Caron and van Eck (2014) produces the best results in any case, which indicates that the differentiation of block size classes for specifying thresholds and the weighting of attributes based on expert knowledge are appropriate concepts for disambiguating bibliometric data. Even though not as good as this approach, the approach of Backes (2018a) also produces

Table 6. Comparisons based on similar sets of attributes

Attribute set	Approach	$F1_{pair}$	$F1_{best}$
Schulz et al. (2014)	Schulz et al. (2014)	0.455	0.773
	Caron and van Eck (2014)	0.637	0.807
Schulz et al. (2014)	Schulz et al. (2014)	0.455	0.773
	Backes (2018a)	0.770	0.819
Caron and van Eck (2014)	Caron and van Eck (2014)	0.808	0.900
	Backes (2018a)	0.721	0.765

good results in the comparisons. Its strategy to consider the specificity of particular features for determining the similarity of author mentions seems to be a promising approach, even if uniform weights are applied on the attribute level.

However, the results in Table 6 also reveal that the choice of attributes has a significant effect on the disambiguation quality. This can be concluded from the differences between the evaluation metrics for the approach of Caron and van Eck (2014) in its original implementation ($F1_{pair}$: 0.808, $F1_{best}$: 0.900), and its implementation used for the comparison with the approach of Schulz et al. (2014) ($F1_{pair}$: 0.637, $F1_{best}$: 0.807): The consideration of more attributes (the original implementation) produces better results. The importance of the choice of attributes also becomes obvious with regard to the results of the approach proposed by Backes (2018a). In this case, however, using more attributes does not necessarily produce better results: Using the same attributes as the approach of Schulz et al. (2014) produces better results than the original implementation (which is based on a larger set of attributes). The reason may be that some of the attributes considered in the original implementation have too much influence in the disambiguation procedure due to the uniform weights on the attribute level. Backes (2018a) also provides the possibility to apply different weights on the attribute level. This might be an alternative for improving the results when including the additional attributes. However, we did not consider this alternative, as the weights for the attributes are not specified automatically by the approach. They would have to be specified manually. Again, this suggests that not only the choice of attributes, but also their weights, play a key role for the quality of disambiguation algorithms.

6. DISCUSSION

In this study, we compared different author name disambiguation approaches based on a data set containing author identifiers in the form of ResearcherIDs. This allows a better comparison of different approaches than previous evaluations, because the comparisons in previous evaluations are generally based on different databases (which are scarcely comparable then). Our results show that all approaches included in the comparison perform better than a baseline that only uses a canonical name representation of the authors for disambiguation. The comparison in this study does not point to the recommendation of one approach for all situations that require a disambiguation of author names. It provides evidence of when which approach can produce good results—especially with regard to the size of corresponding name block sizes. Our analyses show that the parametrization of the approaches can have a significant effect on the results. This effect depends largely on the data at hand. Therefore, a proper implementation of an algorithm always has to take into account the characteristics of the data that has to be disambiguated. In the context of this study (based on its data set), the approach proposed by Caron and van Eck (2014) produced the best results.

Beyond the comparison of the original versions of the approaches, we also examined the role that the set of attributes used by the different approaches has on the results. As the approaches vary in the attributes they used for assessing the similarities between author mentions, differences in the results may rely on the choice of attributes. Our analyses show indeed that this choice has an effect on the results. Differences between the approaches, however, still remain when controlling for the set of attributes included. This means that other features of the approaches (e.g., how similarities are computed, or how similar author mentions are combined to clusters) also have an effect on the disambiguation quality. Based on these findings, we recommend that future research further examines the importance of single attributes and how they should ideally be weighted. The effect of the clustering strategy on the results might be also a topic for future research.

Regarding the evaluation of disambiguation approaches, we tested the results against author profiles from ResearcherID. As these profiles are curated by researchers themselves, the approaches are tested against human-based compilations of publications (i.e., compilations of those humans who are in the best position to reliably assign the publications to their personal sets). It would be interesting to compare the disambiguation approaches with other human-based compilations (e.g., ORCID) to see whether our results are still valid. We do not expect that the results will change significantly; we assume, however, that all human-based compilations are concerned with more or less erroneous records.

Understanding how author name disambiguation approaches behave is important to improve the applied algorithms and to assess the effect they have on analyses that are based on the disambiguated data. A good understanding of this behavior is the basis for reliable bibliometric analyses at the individual level. It is clear that the same is true for any other unit (e.g., institutions or research groups) that is addressed in research evaluation studies.

ACKNOWLEDGMENTS

The bibliometric data used in this paper are from an in-house database developed and maintained in cooperation with the Max Planck Digital Library (MPDL, Munich) and derived from the Science Citation Index Expanded (SCI-E), Social Sciences Citation Index (SSCI), Arts and Humanities Citation Index (AHCI) prepared by Clarivate Analytics, formerly the IP & Science business of Thomson Reuters (Philadelphia, Pennsylvania, USA). We would like to thank Robin Haunschild and Thomas Scheidsteger from the Central Information Service for the institutes of the Chemical Physical Technical (CPT) Section of the Max Planck Society (IVS-CPT) for providing the computational infrastructure for conducting our analyses. Furthermore, we thank Nees Jan van Eck and Tobias Backes for useful discussions about a previous version of this paper.

AUTHOR CONTRIBUTIONS

Alexander Tekles: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing—original draft. Lutz Bornmann: Conceptualization, Supervision, Writing—review & editing.

COMPETING INTERESTS

The authors have no competing interests.

FUNDING

We did not receive specific funding for the research published in this paper.

DATA AVAILABILITY

Access to the Web of Science bibliographic data requires a license from Clarivate Analytics; therefore we cannot make the data publicly available.

REFERENCES

Backes, T. (2018a). Effective unsupervised author disambiguation with relative frequencies. In J. Chen, M. A. Gonçalves, & J. M. Allen (Eds.), *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries* (pp. 203–212). Fort Worth, TX: ACM. **DOI:** https://doi.org/10.1145/3197026.3197036

Backes, T. (2018b). The impact of name-matching and blocking on author disambiguation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 803–812). Torino, Italy: ACM. **DOI:** https://doi.org/10.1145/3269206.3271699

Caron, E., & van Eck, N. J. (2014). Large scale author name disambiguation using rule-based scoring and clustering. In E. Noyons (Ed.), *Proceedings of the Science and Technology Indicators Conference 2014 Leiden* (pp. 79–86). Leiden: Universiteit Leiden—CWTS.

Cota, R. G., Gonçalves, M. A., & Laender, A. H. F. (2007). A heuristic-based hierarchical clustering method for author name disambiguation in digital libraries. Paper presented at the *XXII Simpósio Brasileiro de Banco de Dados*, João Pessoa.

Ferreira, A. A., Gonçalves, M. A., & Laender, A. H. F. (2012). A brief survey of automatic methods for author name disambiguation. *ACM SIGMOD Record*, *41*(2), 15–26. **DOI:** https://doi.org/10.1145/2350036.2350040

Ferreira, A. A., Veloso, A., Gonçalves, M. A., & Laender, A. H. F. (2010). Effective self-training author name disambiguation in scholarly digital libraries. Paper presented at the *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, Gold Coast, Queensland, Australia. **DOI:** https://doi.org/10.1145/1816123.1816130

Ferreira, A. A., Veloso, A., Gonçalves, M. A., & Laender, A. H. F. (2014). Self-training author name disambiguation for information scarce scenarios. *Journal of the Association for Information Science and Technology*, *65*(6), 1257–1278. **DOI:** https://doi.org/10.1002/asi.22992

Hussain, I., & Asghar, S. (2017). A survey of author name disambiguation techniques: 2010–2016. *The Knowledge Engineering Review*, *32*. **DOI:** https://doi.org/10.1017/s0269888917000182

Hussain, I., & Asghar, S. (2018). DISC: Disambiguating homonyms using graph structural clustering. *Journal of Information Science*, *44*(6), 830–847. **DOI:** https://doi.org/10.1177/0165551518761011

Kim, J. (2018). Evaluating author name disambiguation for digital libraries: A case of DBLP. *Scientometrics*, *116*(3), 1867–1886. **DOI:** https://doi.org/10.1007/s11192-018-2824-5

Kim, J. (2019). Scale-free collaboration networks: An author name disambiguation perspective. *Journal of the Association for Information Science and Technology*, *70*(7), 685–700. **DOI:** https://doi.org/10.1002/asi.24158

Kim, J., & Diesner, J. (2016). Distortive effects of initial-based name disambiguation on measurements of large-scale coauthorship networks. *Journal of the Association for Information Science and Technology*, *67*(6), 1446–1461. **DOI:** https://doi.org/10.1002/asi.23489

Levin, M., Krawczyk, S., Bethard, S., & Jurafsky, D. (2012). Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology*, *63*(5), 1030–1047. **DOI:** https://doi.org/10.1002/asi.22621

Li, G.-C., Lai, R., D'Amour, A., Doolin, D. M., Sun, Y., Torvik, V. I., … Fleming, L. (2014). Disambiguation and co-authorship networks of the U.S. patent inventor database (1975–2010). *Research Policy*, *43*(6), 941–955. **DOI:** https://doi.org/10.1016/j.respol.2014.01.012

Liu, Y., Li, W., Huang, Z., & Fang, Q. (2015). A fast method based on multiple clustering for name disambiguation in bibliographic citations. *Journal of the Association for Information Science and Technology*, *66*(3), 634–644. **DOI:** https://doi.org/10.1002/asi.23183

Menestrina, D., Whang, S. E., & Garcia-Molina, H. (2010). Evaluating entity resolution results. *Proceedings of the VLDB Endowment*, *3*(1), 208–219. **DOI:** https://doi.org/10.14778/1920841.1920871

Milojević, S. (2013). Accuracy of simple, initials-based methods for author name disambiguation. *Journal of Informetrics*, *7*(4), 767–773. **DOI:** https://doi.org/10.1016/j.joi.2013.06.006

Newcombe, H. B. (1967). Record linking: The design of efficient systems for linking records into individual and family histories. *American Journal of Human Genetics*, *19*(3), 335–359.

On, B.-W., Lee, D., Kang, J., & Mitra, P. (2005). Comparative study of name disambiguation problem using a scalable blocking-based framework. In M. Marlino (Ed.), *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 344–353). Denver, CO: ACM. **DOI:** https://doi.org/10.1145/1065385.1065463

Schulz, C., Mazloumian, A., Petersen, A. M., Penner, O., & Helbing, D. (2014). Exploiting citation networks for large-scale author name disambiguation. *EPJ Data Science*, *3*(11). **DOI:** https://doi.org/10.1140/epjds/s13688-014-0011-3

Smalheiser, N. R., & Torvik, V. I. (2009). Author name disambiguation. *Annual Review of Information Science and Technology*, *43*(1), 1–43. **DOI:** https://doi.org/10.1002/aris.2009.1440430113

Torvik, V. I., & Smalheiser, N. R. (2009). Author name disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data*, *3*(3), 1–29. **DOI:** https://doi.org/10.1145/1552303.1552304, **PMID:** 20072710

Wu, H., Li, B., Pei, Y., & He, J. (2014). Unsupervised author disambiguation using Dempster–Shafer theory. *Scientometrics*, *101*(3), 1955–1972. **DOI:** https://doi.org/10.1007/s11192-014-1283-x

Wu, J., & Ding, X.-H. (2013). Author name disambiguation in scientific collaboration and mobility cases. *Scientometrics*, *96*(3), 683–697. **DOI:** https://doi.org/10.1007/s11192-013-0978-8

Zhu, J., Wu, X., Lin, X., Huang, C., Fung, G. P. C., & Tang, Y. (2017). A novel multiple layers name disambiguation framework for digital libraries using dynamic clustering. *Scientometrics*, *114*(3), 781–794. **DOI:** https://doi.org/10.1007/s11192-017-2611-8