ANDMC: An Algorithm for Author Name Disambiguation Based on Molecular Cross Clustering

Siyang Zhang[1,3(✉)], Xinhua E[2,3], Tao Huang[1], and Fan Yang[1,3]

[1] Beijing Advanced Innovation Center for Future Internet Technology, Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road, Haidian District, Beijing, People's Republic of China
{zhangyanrui,htao,yfan}@bupt.edu.cn
[2] Beijing University of Technology, No. 100, Pingleyuan, Chaoyang District, Beijing, People's Republic of China
517893410@qq.com
[3] Peng Cheng Laboratory, No.2, Xingke first street, Nanshan District, Shenzhen, Guangdong, People's Republic of China

Abstract. With the rapid development of information technology, the problem of name ambiguity has become one of the main problems in the fields of information retrieval, data mining and scientific measurement, which inevitably affects the accuracy of information calculations, reduces the credibility of the literature retrieval system, and affect the quality of information. To deal with this, name disambiguation technology has been proposed, which maps virtual relational networks to real social networks. However, most existing related work did not consider the problem of name coreference and the inability to correctly match due to the different writing formats between two same strings. This paper mainly proposes an algorithm for Author Name Disambiguation based on Molecular Cross Clustering (ANDMC) considering name coreference. Meanwhile, we explored the string matching algorithm called Improved Levenshtein Distance (ILD), which solves the problem of matching between two same strings with different writing format. The experimental results show that our algorithm outperforms the baseline method. (F1-score 9.48% 21.45% higher than SC and HAC).

Keywords: Name disambiguation · Coreference problem · String matching

1 Introduction

At present, there are several literature retrieval platforms in the world such as China National Knowledge Infrastructure (CNKI), DBLP, CiteSeer, PubMed,

This work is supported by National Natural Science Foundation of China (NSFC) (61702049).

G. Li et al. (Eds.): DASFAA 2019, LNCS 11448, pp. 173–185, 2019.
https://doi.org/10.1007/978-3-030-18590-9_12

etc. The content and quality of the digital library are seriously affected by the ambiguity of author's name, which is regarded as one of the most difficult issues facing digital library [1]. Therefore, how to reduce the impact due to the name ambiguity, and maximize the effectiveness of the digital library, has become a concern of researchers. The "Name Disambiguation" began to be raised and attracted the attention of a large number of experts and scholars.

Name Disambiguation, also known as Entity Resolution [2,3], Name Identification [4], which mainly solves the problem of name coreference and name ambiguity. The name coreference problem mainly appears in the English digital library. It is common that a single author has multiple names in digital library. For example, a possible form of author names A. Lim is Andrew Lim, Abel Lim, etc. The name ambiguity problem common that different authors may share identical names in the real world. For example, there are 57 papers authored by 2 different "Alok Gupta" in the DBLP database.

A lot of work has been studied for Name Disambiguation. For example, Shen, et al. [5] present a novel visual analytics system called NameClarifier to interactively disambiguate author names in publication. However, NameClarifier still heavily relies on human beings' subjective judgments. Kim, et al. [6] used Random Forest to derive the distance function and obtained a good accuracy rate, but the training set required a lot of manual labeling while the model have poor migration. Lin et al. [7] proposed an approach only use the coauthor and title attributes, but they did not consider the coreference problem. Xu et al. [8] considered that each kind of single feature has very strong fuzziness in the expression and used a similarity algorithm. However, many feature inability to correctly match due to the different writing formats between two same strings.

This paper mainly proposes an algorithm called Author Name Disambiguation based on Molecular Cross Clustering (ANDMC) considering name coreference. Meanwhile, we propose the string matching algorithm called Improved Levenshtein Distance (ILD), which solves the problem of matching between two same strings with different writing format. The experimental results show that our algorithm outperforms the baseline method. (F1 value 9.48% 21.45% higher than SC and HAC).

The structure of this paper is as follows: In Sect. 2, we introduce the related research work of name disambiguation. In Sect. 3, we introduce the core of this article including the similarity calculation method of the author name disambiguation and merging procedure. In Sect. 4, we describe our experiment and verify the proposed method. In Sect. 5, we summarize the method proposed in this paper. This part also addresses the shortcomings of the method and its ideas for future improvement.

2 Related Work

The problem of name ambiguity often appears in the literature retrieval platforms, digital library and other similar systems, which has become one of the main problems in the fields of information retrieval, data mining and scientific

measurement. [9] The "Name Disambiguation" which mainly solves the problem of name coreference and name ambiguity began to be raised.

The name coreference problem mainly appears in the English digital library. Newman et al. [10] proposed a heuristic method for complete matching the first letter of the last name and the first name, but some authors is the same as the spelling but different name such as "M. Li", "Min. Li" and "Ming. Li" are merged to reduce the accuracy.

The name ambiguity problem common that difference authors may share identical names in the real world. In general, existing methods for name disambiguation mainly fall into three categories: supervised based [11,12], semi-supervised based [13] and unsupervised based [14–17]. The supervised based method has a high accuracy rate, but the training of massive data requires a lot of manual labeling, which is time-consuming and labor-intensive. What is more, with the advancement of time, the data iteration is rapid. Therefore, the supervised based method has poor portability. Semi-supervised based method use user's feedbacks to get more useful information, but when the amount of data is very large, the user feedbacks information are very difficult to collect and also expend much manpower and material resources in the process of collecting [7]. The biggest advantage of the unsupervised based method is that it does not require a lot of training data and training time. On large-scale data, no method is more feasible and scalable than the unsupervised based method.

The factors that determine the performance of unsupervised based method, not only by the clustering algorithm but also by the calculation of similarity. On the problem of name ambiguity, both the selection of features and how to use these features to calculate similarity are as important as the choice of clustering algorithm. Shin et al. [18], Fan et al. [19] Kang et al. [20] selected coauthor relationships as features, but the author who has not coauthor cannot be distinguish. Lin et al. [7] proposed an approach only use the coauthor and title attributes, but they did not consider the coreference problem. Xu et al. [8] considered that each kind of single feature has very strong fuzziness in the expression and used a similarity algorithm. However, many feature inability to correctly match due to the different writing formats between two same strings.

Based on the previous research results, this paper further studies the Name Disambiguation. The main contributions of this paper can be summarized as follows:

1. Propose the string matching algorithm called Improved Levenshtein Distance (ILD), which solves the problem of matching between two same strings with different writing format. (F1-score 13.08% higher than LD).
2. Propose an algorithm called Author Name Disambiguation based on Molecular Cross Clustering (ANDMC) considering name coreference. (F1-score 21.45% higher than SC, F1-score 9.48% higher than HAC).

3 Proposed Approach

This paper proposes a molecular cross clustering method. The Fig. 1 shows the process of molecular cross clustering. We regard each paper as an atom. Firstly,

these papers are classified according to author's name, while keep the associated category records, and perform atom clustering [21] in the same category to form a molecular. Calculate the molecular similarity between molecular according to the associated category records differentiated by the standard segmentation feature values, and finally obtain the classification result. Each time extract the feature of the previous merge result, which could effectively increase the data amount of the corresponding feature and improve the accuracy of the merge.

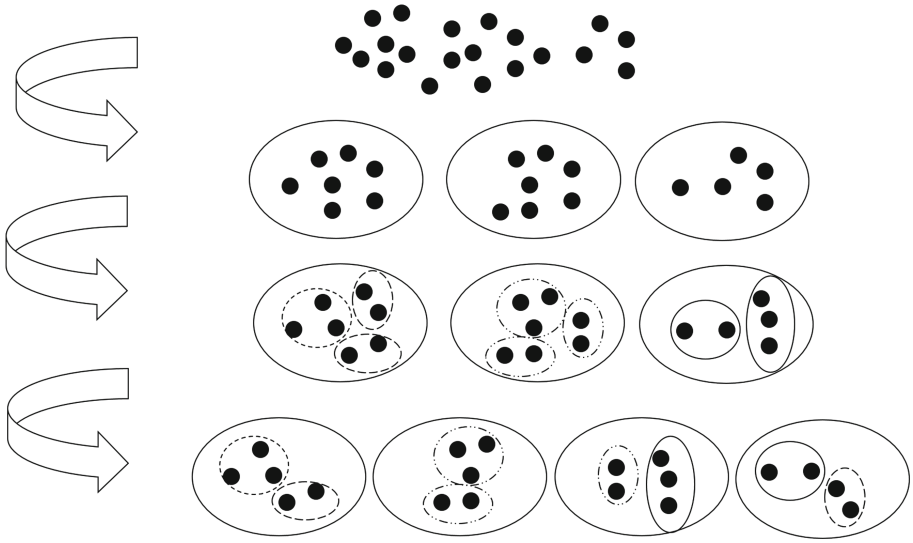

Fig. 1. The process of molecular cross clustering.

The Table 1 lists five records containing the authors of paper, title of paper, and affiliation of paper. It is difficulty for us to make sure that the author "Andrew Lim" is the same person. According to our algorithm, firstly, we can divide this paper into two major categories, Andrew Lim {{1}, {2}, {4}, {5}} and A. Lim {3}. Secondly, it is difficult to directly judge whether Andrew Lim in 1 and 2 is the same person, but 1, 4 have the same collaborator Zhou Xu. After merge 1, 4 we can find that Hu Qin, who is the same collaborator with 2 that means it has a higher probability that 1, 2 are the same person. In the same way, we can easily get the set {1, 2, 4, 5}. At this time, calculate the similarity between the set {1, 2, 4, 5} and {3}, we can find that they have the same collaborator "Fan Wang", the same institution and the similar titles, etc.

The steps of algorithm for Author Name Disambiguation based on Molecular Cross Clustering as follow:

1. Data processing
2. Solve the problem of name ambiguity
   (a) Node relationship division
   (b) Affiliation string matching

Table 1. An example of name disambiguation.

1. Author: **Andrew Lim**, Fan Wang, Zhou Xu
Title: A Transportation Problem with Minimum Quantity Commitment
Affiliation: Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
2. Author: **Andrew Lim**, Zhenzhen Zhang, Hu Qin
Title: Pickup and Delivery Service with Manpower Planning in Hong Kong Public Hospitals
Affiliation: Department of Industrial and Systems Engineering, National University of Singapore, Singapore 117576;
3. Author: **A. Lim**, Fan Wang
Title: Multi-depot vehicle routing problem: a one-stage approach
Organization: Dept. of Ind. Eng. & Logistics Manage., Hong Kong Univ. of Sci. & Technol., China
4. Author: **Andrew Lim**, Hu Qin, Zhou Xu
Title: The freight allocation problem with lane cost balancing constraint
Organization: Department of Management Sciences, City University of Hong Kong, Tat Chee Ave, Kowloon Tong, Hong Kong, School of Management, Huazhong University of Science and Technology, No. 1037, Luoyu Road, Wuhan, China
5. Author: Lijun Wei, Zhenzhen Zhang, **Andrew Lim**
Title: An Adaptive Variable Neighborhood Search for a Heterogeneous Fleet Vehicle Routing Problem with Three-Dimensional Loading Constraints
Affiliation: School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, 330013 Jiangxi, China

3. Solve the problem of name coreference
   (a) Similar name cross match

3.1 Data Processing

Perform pre-processing operations such as integration, cleaning and de-duplication on the data to obtain initial data. Each piece of paper in the initial data as an atom.

Extract the following feature attributes for each paper P:

$$P = (A, \quad T, \quad I) \tag{1}$$

Where A represents the author of the paper, T represents the title of the paper, and I represents the affiliation of the paper.

We treat each paper as a node, let n be a name entity, denoted as n, and for the name n, its variant is denoted as Vn = V1, V2, ..., Vm, where the variant of n include the abbreviated forms, last name and first name rotated form, the change of connection symbol and combinations of them [22]. The set of papers corresponding to the name Vn is denoted by the set Pn = p1, p2, ..., pk, where pi = s1, s2, ..., sk represents a set of all papers containing the author names Vx. Ai = a1, A2, ..., ak represents the author set corresponding to the papers set pi. Ni = n1, n2, ..., nk represents the set of the same name authors corresponding to Ai.

3.2 Node Relation Division (NRD)

In the research of the name disambiguation, the relationship of cooperation between nodes has a strong influence on the correct division of nodes [20]. For two nodes with the same name attribute, if they all have a cooperative relationship with another node, the two nodes have greater similarity.

The set of collaborators of the name Ni can be denoted as:

$$C_i = A_i - N_i = \{a1 - n1, a2, n2, \ldots ak - nk\} = \{c1, c2, \ldots, ck\} \tag{2}$$

Traversal the set N_i, each n_i as a node. Traversal the set C_i, the author in each set c_i generates a node which has a cooperative relationship with the node ni. We use the graph database to generates the author relationship network, and finds the number of connections of the author ni to nj denoted as Num(Lij), according to the Jaccard coefficient similarity function, the similarity between the node ni and the node nj is:

$$sim(n_i, n_j) = \frac{Num(L_{ij})}{|c_i \cup c_j|} \tag{3}$$

When the similarity is greater than the threshold value, ni and nj will be merged.

3.3 Affiliation String Matching (ASM)

The main difficulties in matching affiliation string for English databased is that affiliation write different formats. For example, there have four affiliations as follows: “IBM India Res. Lab, New Delhi”, “IBM India Research Laboratory”, “IBM India Research Lab, New Delhi, India” and “IBM India Research Lab, New Delhi, India 110 070”. It is clearly shown that the above four affiliations belong to the same affiliation, but the writing in different formats which lead the computer cannot match them together correctly. At present, there are many similarity algorithm for string matching, such as Jaccard algorithm, Euclidean Distance, Levenshtein Distance, etc. However, the calculation of the whole affiliation string is not satisfactory. For example, two affiliations as follows:

1. “School of Electrical Engineering & Automation, Henan, Polytechnic University, Jiaozuo, People’s Republic of China”

2. "Department of Electrical Engineering and Automation, Tianjin University, Tianjin, People's Republic of China"

If we directly calculate the similarity of the affiliation names, it is likely to judge them as the same affiliation, but they are not the same affiliation actually. There is also a problem with the calculation of Levenshtein Distance. For example, there are two strings include word "Research" and "Res", the Levenshtein distance of two words is 5, and the similarity is 40%. We find that, in reality, these two words actually belong to a same word. In order to solves the problem of matching between two same strings with different writing format while enhance the accurate of similarity calculate. In this paper, we cut each word in the affiliation. We optimize Levenshtein Distance algorithm as ILD (Improved Levenshtein Distance algorithm) to calculate the similarity of each word. For the affiliation X and the affiliation Y, cut through the separator to obtain the set X = x1, x2, xp and set Y = y1, y2, yq. Construct the relational matching matrix E with the number of rows p and the number of columns q:

$$E_{pq} = \{sim(i,j)\} \tag{4}$$

For each xis1, s2, ..., sm, yjs1, s2, ..., sn construct the relationship matching matrix LD between xi and yj whose row number is m+1 and column number is n+1. The first column of the matrix represents X, and the first row represents Y:

$$LD_{(m+1)\times(n+1)} = \{ld_{ij}\} \quad (0 \leqq i \leqq m, \quad 0 \leqq j \leqq n) \tag{5}$$

Fill the relationship matching matrix LD according to the following formula:

$$ld_{ij} = \begin{cases} i & j = 0 \\ j & i = 0 \\ min(ld_{i-1j-1}, ld_{i-1j}, ld_{ij-1}) + 1 & i,j > 0, x_i \neq x_j \\ ld_{i-1j-1} & i,j > 0, x_i = x_j \end{cases} \tag{6}$$

After fill the matrix LD, the element dmn is the edit distance between xi and yj, which is recorded as:

$$d(x_i, y_i) = \begin{cases} d_{min(m,n)min(m,n)} & x_i \in y_j \ or \ y_j \in x_i \\ d_{mn} & else \end{cases} \tag{7}$$

The similarity sim(xi, yj) is calculated as:

$$sim(x_i, y_j) = 1 - \frac{d(x_i, y_i)}{max(len(x_i), len(y_j))} \tag{8}$$

Where len(xi) and len(yj) are the lengths of the string xi and the string yj, respectively. When sim(xi, yj) = 1, the string xi and yj exactly match. For the matrix Epq, if exist at least one sim(xi, yj) = 1 on the p-row or q-column, we think that the affiliation X and the affiliation Y have one word exactly matched which is recorded as:

$$CM(k) = \begin{cases} 1 \ existsim(x,y) = 1 \ in \ link \ k \\ 0 \quad else \end{cases} \tag{9}$$

The similarity of the word exactly match in the affiliation X and Y as follows:

$$sim(X,Y)_{cm} = \frac{average(\sum_p CM(p), \sum_q CM(q))}{average(p,q)} \tag{10}$$

The similarity of the word non-exactly match in the affiliation X and Y as follows:

$$\begin{aligned} sim(X,Y)_{other} = & \frac{average(\sum_p max(sim(X,Y)), \sum_p max(sim(X,Y)))}{average(p,q)} \\ & - \frac{average(\sum_p CM(p), \sum_q CM(q))}{average(p,q)} \end{aligned} \tag{11}$$

The similarity between the affiliation X and Y is:

$$sim(X,Y) = sim(x,y)_{cm} \times W_1 + sim(X,Y)_{other} \times W_2 \tag{12}$$

3.4 Similar Name Cross Match (SNCM)

For a name entity n, each variant in Vn = V1, V2, ... Vm has solved the problem of name ambiguity. This part mainly solves problem of name coreference. We need to calculate the similarity between each Ai in Vx, which denoted as "Vx.Ai" and each of Aj in Vy, which denoted as "Vy.Aj". We calculate the corresponding similarity Sx according to the features A, T, I, and set the weight W, respectively. The similarity between Vx.Ai and Vy.Aj are as follows:

$$S(V_x.A_i, V_y, A_j) = \begin{cases} S_A = S_{JACCARD}(V_x.A_i, V_y.A_j) \\ S_T = S_{LD}(V_x.T_i, V_y.T_j) \\ S_I = S_{ILD}(V_x.I_i, V_y.I_j)) \end{cases} \tag{13}$$

We chose to put similar name cross matching in the last step due to the current similarity calculation does not guarantee 100% accuracy for authors with the same name and a large number of duplicate names. Since each of our mergers is based on the previous step. As a result, we must ensure that the accuracy of the previous merge is as high as possible. If this step is advanced, it will greatly affect the accuracy of the subsequent steps.

4 Experiments

4.1 Data Sets

In our experiments, we perform evaluations on a dataset constructed by Tang et al. [21], which contains the citations collected from the DBLP Website. We downloaded this dataset from the Kaggle. However, the data set is only labelled within the same name range, and the name containing the abbreviation is less. Therefore, we add some real intellectual property disclosure data on the basis of this data set to verify our method. Select some authors as experimental samples.

When evaluating the classification results, we use the author whose name is prone to the same name as a sample. We use manual methods to create standard categories. The process is as follows: For each author name in Table 2, we retrieve all the papers published by the name in the database. Classify the authors of the same name by human annotated, as best as possible to accurately.

Table 2. Evaluation dataset.

Name	Number	Year
Alok Gupta	57	1996–2009
Ming Li	34	2003–2018
M. Li	15	1991–2014
Min Li	30	2001–2018
F. Wang	34	1998–2017
Fan Wang	55	1989–2016
A. Lim	7	1993–2005
Andrew Lim	8	2008–2014
X. Zhang	61	1984–2012
Xin Zhang	46	2002–2018

4.2 Evaluation Indicators

To evaluate and compare the performance of different methods on the Name Disambiguation tasks. In this paper, we use pairwise precision, pairwise recall and pairwise f1-measure to measure the results. We define the measures as follows:

$$PairwisePrecision = \frac{\#PairsCorrectlyPredictedToSameAuthor}{\#TotalPairsPredictedToSameAuthor} \tag{14}$$

$$PairwiseRecall = \frac{\#PairsCorrectlyPredictedToSameAuthor}{\#TotalPairsToSameAuthor} \tag{15}$$

$$PairwiseF - Measure = \frac{2 \times PairwiseRecall \times PairPrecision}{PairwiseRecall + PairPrecision} \tag{16}$$

In the above formula, #PairsCorrectlyPredictedToSameAuthor refers to the number of papers that with the same label predicted by an approach and have the same label in the human annotated data set. #TotalPairsPredictedToSameAuthor refers to the number of papers that with the same label predicted by an approach. #TotalPairsToSameAuthor refers to the number of papers that have the same label in the human annotated data set.

Table 3. Table captions should be placed above the tables.

Author	LD			ILD		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Alok Gupta	100.00	100.00	100.00	100.00	90.48	95.00
Ming Li	60.87	70.00	65.12	87.50	70.00	77.78
M. Li	72.73	80.00	76.19	100.00	100.00	100.00
Min Li	80.95	89.47	85.00	100.00	100.00	100.00
F. Wang	50.00	100.00	66.67	100.00	100.00	100.00
Fan Wang	100.00	100.00	100.00	100.00	100.00	100.00
A. Lim	57.14	66.67	61.54	100.00	100.00	100.00
Andrew Lim	100.00	100.00	100.00	100.00	50.00	66.67
X. Zhang	80.56	85.29	82.86	100.00	82.35	90.32
Xin Zhang	40.00	72.73	51.61	100.00	81.82	90.00
Average	74.22	86.42	78.90	**98.75**	**87.46**	**91.98**

4.3 Experimental Results

We considered the baseline methods on LD algorithm. In this step, we only evaluate based on the feature of affiliation, and do not evaluate the results based on other feature. Table 3 shows the results of some examples in our data sets.

Obviously, it can be seen from the experimental results that the ILD algorithm has a better improvement than the LD algorithm in each evaluation value (+17.76% over LD by average F1 score, +24.53% over LD by average Precision). On the other hand, our method has higher precision than baseline methods (+18.3% over SC, +8.51% over HAC by the average Precision value).

According to the name similarity matching, the number of names existing in each name set as follows (Table 4):

Table 4. Evaluation dataset.

Name	Num. authors	Num. records
Alok Gupta	2	57
Ming Li, M. Li, Min Li	44	79
F. Wang, Fan Wang	28	89
A. Lim, Andrew Lim	3	15
X. Zhang, Xin Zhang	72	107

In this paper, we considered several baseline methods based on Hierarchical Agglomerative Clustering (HAC) [24], [23] and single-clustering (SC) [20]. SC only uses the feature of collaborator for disambiguation. HAC uses Jaccard

Similarity and ILD Similarity algorithms with the feature of author's name, affiliation, and collaborator. For a fair comparison, we use the same threshold for the same attribute feature. For each feature, we compare and select the thresholds to ensure that the highest recall rate based on the precision as high as possible. Table 5 gives the threshold values of features.

Table 5. Threshold values of features.

Feature	A	I	T
Thresholds	0.6	0.7	0.5

Table 6 gives the results of some examples in the data set. Obviously, our method outperforms the baseline method in name disambiguation (+21.45% over SC, +9.48% over HAC by average F1 score). On the other hand, our method has higher precision than baseline methods (+18.3% over SC, +8.51% over HAC by the average Precision value).

Table 6. Results of name disambiguation.

Author	SC			HAC			NAS		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Alok Gupta	36.54	33.33	34.86	80.77	73.68	77.06	80.77	73.68	77.06
A. Lim, Andrew Lim	50.00	33.33	40.00	61.54	53.33	57.14	100.00	93.33	96.55
X. Zhang, Xin Zhang	100.00	87.93	93.58	100.00	89.66	94.55	100.00	93.10	96.43
Ming Li, M. Li, Min Li	93.10	86.27	89.56	86.27	92.16	89.12	92.16	100.00	95.92
F. Wang, Fan Wang	100.00	78.57	88.00	100.00	78.57	88.00	98.21	78.57	87.30
Average	75.93	63.89	69.20	85.72	77.48	81.17	**94.23**	**87.74**	**90.65**

5 Conclusion and Discussion

Name Disambiguation in the digital library is an important task because different authors can share the same name, and an author can have many name variant. This paper mainly proposes an algorithm called Author Name Disambiguation based on Molecular Cross Clustering (ANDMC). We have also explored a string matching algorithm called Improved Levenshtein Distance (ILD). Experimental results indicate that the proposed method significantly outperforms the baseline methods. It's performance in the problem of name coreference is quite satisfying. Meanwhile, we solve the problem of matching between two same strings with different writing format. In the future, we will pay more attention to the speed of the algorithm and improve the efficiency of the algorithm.

References

1. Hussain, I., Asghar, S.: A survey of author name disambiguation techniques. Knowl. Eng. Rev. **32**, 1–24 (2018)
2. Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J.: Swoosh: a generic approach to entity resolution. The VLDB J. **18**, 255–276 (2008)
3. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Trans. Knowl. Discov. Data **1** (2007) Article no. 5
4. Li, X., Morie, P., Roth, D.: Identification and tracing of ambiguous names: discriminative and generative approaches. In: Proceedings of 19th National Conference on Artificial Intelligence (AAAI 2004), pp. 419–424 (2004)
5. Shen, Q., Wu, T., Yang, H., Wu, Y., Qu, H., Cui, W.: NameClarifier: a visual analytics system for author name disambiguation. IEEE Trans. Vis. Comput. Graph. **23**(1), 141–150 (2017)
6. Kim, K., Khabsa, M., Giles, C.L.: Random Forest DBSCAN for USPTO inventor name disambiguation, pp. 269–270 (2016)
7. Lin, X., Zhu, J., Tang, Y., Yang, F., Peng, B., Li, W.: A novel approach for author name disambiguation using ranking confidence. In: Bao, Z., Trajcevski, G., Chang, L., Hua, W. (eds.) DASFAA 2017. LNCS, vol. 10179, pp. 169–182. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55705-2_13
8. Xu, X., Li, Y., Liptrott, M., Bessis, N.: NDFMF: an author name disambiguation algorithm based on the fusion of multiple features. In: IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo 2018, pp. 187–190 (2018)
9. Ferreira, A., Goncalves, M.A., Laender, A.H.: A brief survey of automatic methods for author name disambiguation. ACM Sigmod Rec. **41**(2), 15–26 (2012)
10. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**, 026113 (2004)
11. Han, H., Giles, L., Zha, H., et al.: Two supervised learning approaches for name disambiguation in author citations. In: Proceedings of JCDL (2004)
12. Huang, J., Ertekin, S., Giles, C.L.: Efficient name disambiguation for large-scale databases. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 536–544. Springer, Heidelberg (2006). https://doi.org/10.1007/11871637_53
13. Quan, L., Bo, W., Yuan, D.U., Wang, X., Yuhua, L.I.: Disambiguating authors by pairwise classification. Tsinghua Sci. Technol. **15**(6), 668–677 (2010)
14. Malin, B.: Unsupervised name disambiguation via social network similarity. In: SIAM SDM Workshop on Link Analysis, Counterterrorism and Security (2005)
15. Pedersen, T., Purandare, A., Kulkarni, A.: Name discrimination by clustering similar contexts. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 226–237. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30586-6_24
16. Cen, L., Dragut, E.C., Si, L., Ouzzani, M.: Author disambiguation by hierarchical agglomerative clustering with adaptive stopping criterion. In: SIGIR 2013, 28 July–1 August 2013
17. Evans, M.D.: A new approach to journal and conference name disambiguation through k-means clustering of internet and document surrogates (2013)
18. Shin, D., Kim, T., Jung, H., et al.: Automatic method for author name disambiguation using social networks. In: IEEE International Conference on Advanced Information NETWORKING and Applications, Aina 2010, Perth, Australia, 20–13 April. DBLP, pp. 1263–1270 (2010)

19. Fan, X., Wang, J., Pu, X., et al.: On graph-based name disambiguation. J. Data Inf. Qual. **2**(2), 10 (2011)
20. Kang, I.-S., et al.: On co-authorship for author disambiguation. Inf. Process. Manag. **45**(1), 84–97 (2009)
21. Tang, J., Fong, A.C.M., Wang, B., Zhang, J.: A unified probabilistic framework for name disambiguation in digital library. IEEE Trans. Knowl. Data Eng. **24**(6), 975–987 (2012)
22. Tang, J., Lu, Q., Wang, T., Wang, J., Li, W.: A bipartite graph based social network splicing method for person name disambiguation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011). ACM, New York, pp. 1233–1234 (2011)
23. Tan, Y.F., Kan, M.Y., Lee, D.: Search engine driven author disambiguation. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, JCDL, Chapel Hill, NC, USA, 11–15 June, pp. 314–315 (2006)
24. Zepeda-Mendoza, M.L., Resendis-Antonio, O.: Hierarchical agglomerative clustering. Encycl. Syst. Biol. **43**(1), 886–887 (2013)