Author Disambiguation

Ekaterina Bastrakova, Rodney Ledesma, Jose Millan

kat.bastrakova@gmail.com — reln13st@gmail.com — jampmil@gmail.com

Abstract

Author disambiguation is an open issue in the world of academic digital libraries. As many problems arise when trying to identify if two different authors are the same and then group them, this issue has become more relevant inside the scientific community. This paper illustrates a workflow that aims to solve this issue. By using the best of a relational database engine and data mining techniques implemented in R, we have implemented a workflow that correctly disambiguates authors present in papers retrieved from the Internet. To evaluate the results we perform a two-step-validation process inside the workflow, validating if two articles were written by the same author, and, if so, validating the authors grouped together as an unique disambiguated author. With the validations performed, the workflow implemented allows the process of identifying and disambiguating a any new author.

1 Introduction

Author disambiguation is an open issue in the world of academic digital libraries. This is a difficult task due to different problems such as authors with the same first name and last name, incomplete information (e.g. only the initials), misspelling, change of the author's name, among others. This issue has become relevant in the past years due an increasing quantity of the researchers, both in academics and industry, searching what has been published and by whom. Besides, research evaluation and ranking depend on the accuracy of this information, as well for the improvement of query systems.

There are several approaches to try to solve this problem. Among them, author grouping methods are emerging. These methods use similarity measures to find close articles and estimate real authors [1] [2]. Aligned with this method, we try to analyze all the available features, creating models to compare signatures and then cluster the real authors.

The workflow we propose takes advantage of the best of two worlds: a relational database engine and different data mining techniques, which successfully disambiguates authors present in papers retrieved from the Internet. Additionally, in order to evaluate the results, we perform a two-step-validation process inside the workflow, validating if two articles were written by the same author, and, if so, validating the authors grouped together as an unique disambiguated author.

The paper is organized as follows: in Section 2 we briefly review related solutions for author disambiguation. In Section 3 we describe the workflow for this solution, detailing the source data structure and the features calculated from it, along with the methods used to identify equal authors and group them. Following this, in Section 4 we describe the implementation of the workflow. In Section 5 we present the obtained results, and finally in Section 6 we discuss the conclusions and the future work.

2 Related Work

The author disambiguation challenge created a broad number of works and methods. The 2012 survey paper [3] proposed a taxonomy with three main categories: manual disambiguation methods [4] (which includes the initiative of creating the unique author IDs [5]), author assignment and author grouping techniques. Massive disambiguation tasks would require a lot of human resources, therefore it has to be done automatically using one of the last two methods. Author assignment techniques are trying to classify the article to a list of predefined labels (authors) using a supervised machine learning [6] or model-based clustering techniques [7].

Larger corpus of related work, including our paper, refers to the second category: author grouping. This method uses similarity measures to find close papers and estimate real authors. Type and availability of citation data in the initial dataset (author, title, publication venue, keywords etc.) highly affect the type of similarity measures used, hence the variety of methods. Measures include Jaccard index [1], Levenshtein and Euclidean distance, cosine similarity on a TF-IDF representation of the features and their various combinations; other papers propose to use custom distance function that is learned on labeled data [2].

Several works are based on graph-based similarity functions - like coauthorship graph [8], exploiting the idea that researchers of the same field tend to work together as well as that the researcher cannot be a coauthor of himself; or citation graph [9], based on the fact that authors tend to cite themselves. These approaches use direct link or "shortest path" metric .

Recently it has been successfully proposed to use ethnicities estimated from the surname of the author as a group of features [10]. We push this approach further, combining them into single feature: Ethnicity distance.

3 Proposed Workflow

For the solution we propose, it is important to have in mind the concept of *signature*. A signature of an author is basic information of that author present in a single article. It is usually composed by the name of the author, the position in the article and the institution that author belongs to. Our goal in the current work is to identify which signatures present in different articles belong to the same author.

With this in mind, taking into consideration the related work presented in Section 2, and using the best of two worlds: a relational database engine and data mining techniques, we propose the following workflow for disambiguating authors. For this, we describe the source data needed for the task, along with the features we added to it for making the disambiguation process possible. Finally we present step by step the workflow and how it can successfully achieve this task.

3.1 Source Data Description

In order to disambiguate the signatures present in different articles we need to have a basic data structure that allows us to perform the disambiguation task. For this, as it can be seen in Figure 1, we define the minimal information required for our workflow to work.

The main entity of our source data are the articles (containing their basic information: tittle, journal where it was published, DOI and the publication year), the subject of the article, its keywords and the journals referenced. Together with the articles, we have the different signatures of each article. Each signature contains the information of the author (fist name, the initials of the authors names, the last name) along with the institution the author belonged at the moment of writing the signature.

3.2 Complementing Features

Exploding the source data, we calculate a set of features that complement the information about the articles and their signatures and allow us to determine if two signatures are from the same author or not. We add the focus name, a phonetic representation based on the last name of the author; additionally we calculate the possible ethnicities the author may belong to based on the author's names; and finally we calculate a LDA-based topic for every article we have. This features are described in more detail below.

3.2.1 Focus Name

A focus name of a signature is defined as the simplified version of the last name of the author, leaving any language complexity and possible spelling errors aside. For example, the authors *C. Smith*, *A. Smit* and *G. Smoot* all share the same focus name: *SMT*.

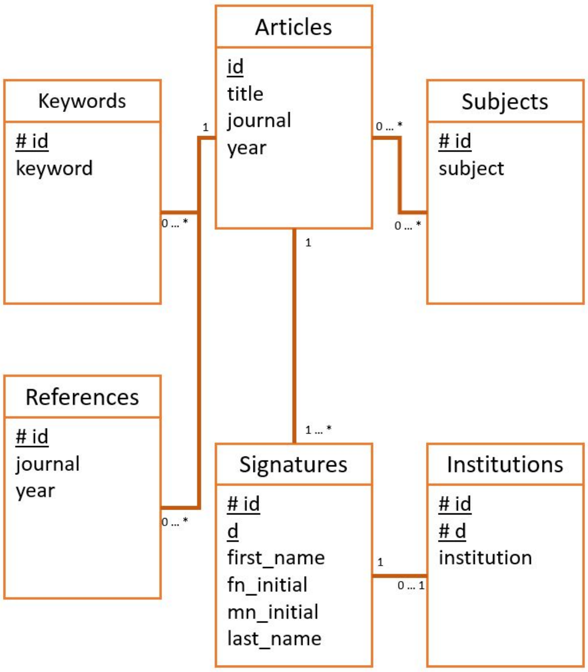

Figure 1: Source Data Structure

The focus name is calculated by using a phonetic algorithm: Metaphone. Metaphone returns a rough approximation of how a word sounds, which should be the same for words or names that sound similar or for words misspelled [11].

3.2.2 Ethnicity

Considering the source data available, an interesting feature can be calculated based solely on the last name of an author's signature: the possible Ethnicity of the author. For this, and based on different similar work performed [10] [12], we use the dataset of Frequently Occurring Surnames from the Census 2000 [13] as the base information to train a classifier that indicates the possible ethnicities to which an author may belong.

Among other information, this Census dataset provides us with the Surnames Occurring 100 or more times, along with the percentages to which those last names belong to. The possible ethnicities presented are:

- Percent of respondents with this surname that claimed their origin group to be Non-Hispanic White Only (pctwhite)
- Percent of Non-Hispanic Black Only (pctblack)
- Percent of Non-Hispanic Asian and Pacific Islander Only (pctapi)
- Percent of Non-Hispanic American Indian and Alaskan Native Only (pctaian)
- Percent of Non-Hispanic of Two or More Races (pct2prace)
- Percent of Hispanic Origin (pcthispanic)

Using these last names and the different percentages for the ethnicities, and based on the ethnicity based work in [12], we generate the bi-grams [14] derived from each individual last name and complement this information adding a phonetic representation of the last name using the Soundex algorithm [15], with which we create a Support Vector Machine model that predicts if a given last name belongs to an specific ethnicity or not. These implies that to our set of Complementing Features we add 6 new Ethnicity features, each related to one specific ethnicity present in the Census data, as demonstrated in Table 1.

	white	black	api	aian	2prace	hispanic
Name1	1	0	1	0	0	0
Name2	0	1	0	0	1	0

Table 1: Complementing ethnicity features

3.2.3 LDA Topic

Considering that it is very likely that an author writes articles in the same field, we exploit this to determine if two signatures are the same. For this we calculate the feature ”LDA Topic”. This a topic generated by Latent Dirichlet Allocation [16]. This contributes to the disambiguation process, as the LDA Topic is a more general topic than the given subject from the source data, and can detect implicit connections between the articles.

For this, and considering the source data, we calculate the topics of the articles using their titles and keywords as a text corpus, and separate them in 8 different groups, according the different areas of knowledge [17], and the different experiments we performed. According to this, in Table 2 the four most frequent topics can be appreciated.

Topic 1	Topic 2	Topic 3	Topic 4
”epilepsi”	”network”	”leukemia”	”cell”
”surgeri”	”magnet”	”chronic”	”receptor”
”radio”	”system”	”myeloid”	”apoptosi”
”cortic”	”mobil”	”acut”	”protein”
”dysplasia”	”wireless”	”respons”	”activ”

Table 2: Frequent terms in most frequent LDA Topics

3.3 Similarity Features

Having the complete set of features (from the source data and the complementing features described in Section 3.2), we need now to calculate the Similarity Features. These features, as their name indicates, are calculated by comparing the information of two different signatures (with their corresponding article information). A summary of these features can be appreciated in Table 3 below.

For the case of the First Name Initial, the Second Name Initial and the LDA Topic, we indicate if for the pair of signatures being compared, their values are equal or not. Additionally for the Publication Year of the signature's articles we calculate the absolute difference. Finally for the Keywords, the Journals References, the Subjects, the Tittle, the Coauthors of the article and the Ethnicities of the signature's last name, we calculate the Jaccard distance. For the last one, a more in depth explanation can be found in Section 3.3.1 below.

Feature Name	Calculation Description
First Name Initial	Equality
Second Name Initial	Equality
LDA Topic	Equality
Publication Year	Absolute difference
Keywords	Jaccard distance
References	Jaccard distance
Subject	Jaccard distance
Title	Jaccard distance
Coauthors	Jaccard distance
Ethnicity	Jaccard distance

Table 3: Features calculated for every pair of authors

3.3.1 Distance-based Features - Jaccard Distance

As a measure to indicate how close are two articles related according to the information we have from the source data, we use the Jaccard index. The Jaccard distance computes the similarities of asymmetric information on binary attributes. This is calculated as $d_{ij} = \frac{q+r}{p+q+r}$ where d_ij is the Jaccard index, p is the number of variables that are positive for both objects, q number of variables that are positive for the ith objects and negative for the jth object, r number of variables that are negative for the ith objects and positive for the jth object, and s number of variables that are negative for both objects [18].

For our specific Similarity Features, we calculate them as described below:

- For the Keywords, Subject and Title Distances, we set the variables for the Jaccard distance as each word of the correspondent source data.
- For the References Distance we use the journals referenced by each article and set the names as the variables for the Jaccard distance.
- For the Coauthors Distance we use the focus name of each coauthor of the current signature's article as the variables for the Jaccard distance.
- For the Ethnicity Distance we take the values of each ethnicity for the specific signature and use them as the variables for the Jaccard distance.

3.4 Process Flow

Having the source data, as described in Section 3.1, the first step of the workflow is to calculate the Complementing Features for Focus Name, LDA Topic and Ethnicities of the author's last name, as described in Section 3.2.

After this, we group the signatures (together with their corresponding article information) by focus name, and process each focus name group at the time. Within each focus name, we generate the cross product for every signature of that specific focus name, and for each pair of signatures the Similarity Features are generated, as described in Section 3.3.

Using these features, we predict if each pair of signatures are the same or not by using different data mining classification algorithms. In our work, four methods were implemented: Support Vector Machine, Logistic Regression, Gradient Boosting and Random Forest.

With the results from the previous step, the next goal is to build the clusters of disambiguated authors. For this, we use hierarchical clustering, that results in the corresponding clusters for every signature in the focus name. Every cluster that is calculated here represents a single disambiguated author.

The complete workflow of our solution can be appreciated in Figure 2.

4 Implementation

The source code of the implemented Workflow, along with the database schema and source data, are publicly available online in a GitHub repository [1]. As mentioned previously, the implementation of the presented work is made by using a combination of both a relational database, specifically PostgreSQL 9.5 [19], for storing and handling the data, and R [20] as the programming language for implementing the models and calculating the disambiguated clusters. This approach brings the best of two worlds, taking advantage of the relational database engine optimization (specially for multiple joins and cross-products), as well as the data mining libraries already implemented for R, optimizing the process using parallelization.

For calculating the Complementing and the Similarity Features, as for creating the different machine learning models, different libraries in both PostgreSQL and R have been used. Below we describe the details for each specific method.

- The Focus Name feature is calculated using the Metaphone algorithm, present in the *fuzzystrmatch* contrib library of PostgreSQL [21].
- For calculating the LDA Topic we make use of the *topicmodels* R package [22] with $k = 8$ as mentioned in Section 3.2.3.
- For calculating the Ethnicity Complementing Features, we generate bi-grams using the *ngram* R package [23]. Similarly to generate the Soundex phonetic representation of the last names we use the *phonetic* function present in the *stringdist* R package [24].

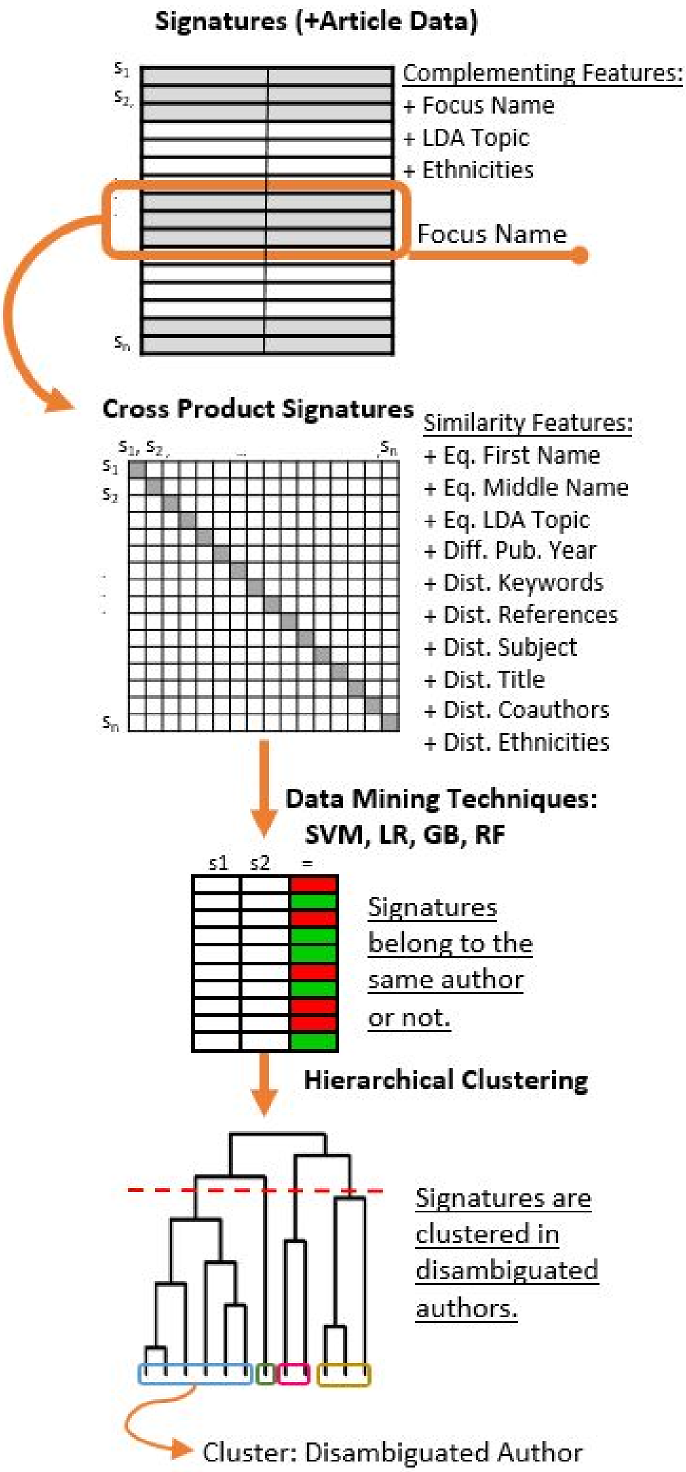

Figure 2: Author Disambiguation Process Workflow

- The Jaccart index is calculated using the *vegdist* function present in the *vegan* R package [25].
- The Random Forest model is created using the *randomForest* R Package [26] with the default parameters.
- The Gradient Boosting model is created using the *xgboost* R Package [27], Extreme Gradient Boosting, which is an efficient implementation of gradient boosting framework.

[1] https://github.com/DMKM1517/author_disambiguation

- The Support Vector Machine model is created using the *kernlab* R Package [28], which performs a kernel-based machine learning SVM implementation for classification. In the different test we performed, we found the *besseldot* kernel, with cost of constraints violation $C = 100$ as the best configuration for this classification problem.
- For the Logistic Regression model we make use of the *glmnet* R Package [29], that contains an efficient procedure for creating logistic regression models, using the function *glmnet* with the parameter "*binomial*".
- Finally, for the Hierarchical Clustering performed with the results of the different models, we make use of the *hclust* function present in the *stats* R core package [30], using the *complete* method.

Along with these libraries, we made use of publicly available data, coming from the Web of Science website [31], to create our source data. In order to train and test the models, we made use of a manually disambiguated set of 1330 unique signatures with their corresponding article information, that correspond to 236 real authors.

5 Results

In order to validate the results of the author disambiguation workflow we implemented a two-step-validation process. The first step is to check if a pair of signatures are the same or not, and then, after the authors have been clustered, the second step is to verify if the calculated cluster of an disambiguated author corresponds to the real cluster.

For the first step, and with the calculated values of the features, we built four models to classify if a pair of signatures are the same or not, using the following algorithms: Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR) and Support Vector Machines (SVM), as described before.

For training the models, we create the training and testing set with different focus names groups in order to eliminate bias within the sets (we need to have in mind that the information within a focus name group is highly related due to the the cross product of all the signatures). We divide the set of focus names so 70% of them are assigned for training and 30% for testing, which makes sure that our testing set contains no information that was already trained beforehand. The results for the first step validation can be appreciated in Table 4 below.

Model	Accuracy	Precision	Recall	F1
RF	0.9671	0.9850	0.9772	0.9811
GB	0.9751	0.9938	0.9777	0.9857
LR	0.9756	0.9877	0.9843	0.9860
SVM	0.9447	0.9594	0.9782	0.9687

Table 4: First Validation

We can see that all the models have an accuracy around 97% and an F1-measure around 98%. But, the best model in this step is Logistic Regression (LR), with 98.84% of F1-score in the testing set.

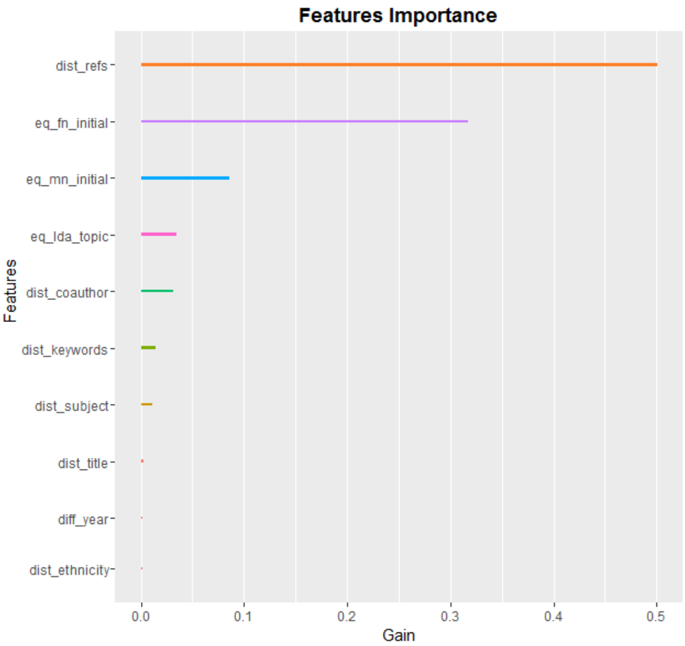

Figure 3: Features Importance for GB

Furthermore, the *xgboost* R package [27] provides a function to look at the gain of the features. The Figure 3 shows the features importance for the Gradient Boosting algorithm, where we can see that the best feature is the distance of the referenced journals, followed by the initials of first name of the author. In contrast, the distance of the titles and ethnicities and the difference of the publication year provide the least gain.

Continuing, the process is very similar for the second step. Using the results of the algorithms from the previous step, we build a distance matrix for every author inside a focus name and then run a hierarchical clustering process that gives us the corresponding cluster of each author that was processed.

As commonly performed in author disambiguation research, we evaluate the predicted clusters over testing data using both B3 and pairwise precision, recall and F-measure, defined as:

$$Precision_{Pairwise} = \frac{|p(R) \cap p(C)|}{|p(C))|} \tag{1}$$

$$Recall_{Pairwise} = \frac{|p(R) \cap p(C)|}{|p(R))|} \tag{2}$$

$$F1_{Pairwise} = \frac{2 * Precision_{Pairwise} * Recall_{Pairwise}}{Precision_{Pairwise} + Recall_{Pairwise}} \tag{3}$$

$$Precision_{B3} = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|R(S_i) \cap C(S_i)|}{|C(s_i)|} \tag{4}$$

$$Recall_{B3} = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|R(S_i) \cap C(S_i)|}{|R(s_i)|} \tag{5}$$

$$F1_{B3} = \frac{2 * Precision_{B3} * Recall_{B3}}{Precision_{B3} + Recall_{B3}} \tag{6}$$

where S is the set of signatures inside an focus name group, $R(S_i)$ is the cluster of the real author of the signature S_i, while $C(S_i)$ is the calculated cluster by the model for the signature S_i. Finally p(X) is all the possible pairs of signatures on the cluster X.

These measures are shown in the Table 5. In this case, Random Forest has the best precision with 92.8%, nevertheless its recall is low making the overall measure not the best. On the other hand, the Logistic Regression model performed better than the others, with an F1-measure of 84.8% . It is also important to notice that the Gradient Boosting model has high measures as well, having the best recall with 95.8%, but having a lower F1-measure than the Logistic Regression Model.

Model	Method	Precision	Recall	F1
RF	Pairwise	0.9289	0.4645	0.6193
	B3	0.9274	0.7652	0.8385
GB	Pairwise	0.7218	0.9587	0.8236
	B3	0.7523	0.9496	0.8395
LR	Pairwise	0.7850	0.9231	0.8485
	B3	0.7913	0.9074	0.8454
SVM	Pairwise	0.5957	0.9128	0.7210
	B3	0.6013	0.8795	0.7143

Table 5: Second Validation

6 Conclusions

In this work, we have presented and implemented a solution for author disambiguation, adding *complementing* and *similarity* features in order to improve the accuracy of the solution. With the proposed workflow for author disambiguation, using four different models, we successfully identified equal signatures and we were able to cluster them into the corresponding disambiguated authors.

In the Section 5, we identified that the feature that provides the highest gain for the models is the distance of referenced journals. This can be explained as an author usually references journals of his/her research area, which at the end, they are rather constant. This could lead to a new study focusing preferably on the communities of the authors. Similarly, the initials of the author are key in the disambiguation process, as it can be supposed beforehand.

After these, the calculated LDA topic has a high relevance in the classification, much more than the given subject of the article from the source data. This implies that calculating the topic from the title and keywords helps the disambiguation process to a greater extend than the labeled subject of the paper.

With the two-step validation process we implemented, we determined that the best model for this problem is the Logistic Regression. With this model we achieved an F1-score of 98.60% in the first validation and 84.85% in the second one. It is also important to mention that even though it was not the best, the Gradient Boosting Model achieved similar results and should be taken into account when choosing a definitive model for this problem. On the other hand, the SVM model gave the most poor results with 96.87% in the first step of the validation and 78.10% in the second one.

Apart from this, we combined the use of a relational database management system and a statistical in-memory software, which both of them are well-accepted in the community and have plenty of extensions to work through. The benefits of this, besides the publicly available libraries, are that we do not exhaust the memory and we can work with large datasets, having the possibility to create a scalable workflow that can evolve into real applications.

Even though the results that we achieved are satisfactory, further work to improve them can be done. Some ideas for this include calculating the distances with other methods, for example using graphs for co-authorship or for community detection; experimenting with other algorithms and techniques, for instance deep learning; and also including or discovering new features, such as DOI (for those papers that have it) or other unique information. Additionally, we could integrate a user feedback and use reinforcement learning to improve the solution.

References

[1] A. Campar, B. Kolbay, H. Aguilera, I. Stankovic, K. Co, F. Rico, and D. A. Zighed, *Foundations of Intelligent Systems: 22nd International Symposium, ISMIS 2015, Lyon, France, October 21-23, 2015, Proceedings.* Cham: Springer International Publishing, 2015, ch. Author Disambiguation, pp. 458–464.

[2] V. I. Torvik, M. Weeber, D. R. Swanson, and N. R. Smalheiser, "A probabilistic similarity metric for medline records: A model for author name disambiguation." in *AMIA*. AMIA, 2003.

[3] A. A. Ferreira, M. A. Gonçalves, and A. H. F. Laender, "A brief survey of automatic methods for author name disambiguation." *SIGMOD Record*, vol. 41, no. 2, pp. 15–26, 2012.

[4] C. L. Scoville, E. D. Johnson, and A. L. McConnell, "When a. rose is not a. rose: the vagaries of author

searching," *Medical reference services quarterly*, vol. 22, no. 4, pp. 1–11, 2003.

[5] Open researcher and contributor id. [Online]. Available: http://orcid.org/

[6] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. F. Laender, "Effective self-training author name disambiguation in scholarly digital libraries." in *JCDL*, J. Hunter, C. Lagoze, C. L. Giles, and Y.-F. Li, Eds. ACM, 2010, pp. 39–48.

[7] H. Han, W. Xu, H. Zha, and C. L. Giles, "A hierarchical naive bayes mixture model for name disambiguation in author citations." in *SAC*, H. Haddad, L. M. Liebrock, A. Omicini, and R. L. Wainwright, Eds. ACM, 2005, pp. 1065–1069.

[8] F. H. Levin and C. A. Heuser, "Evaluating the use of social networks in author name disambiguation in digital libraries." *JIDM*, vol. 1, no. 2, pp. 183–198, 2010.

[9] D. M. McRae-Spencer and N. R. Shadbolt, "Also by the same author: Aktiveauthor, a citation graph approach to name disambiguation." in *JCDL*, G. Marchionini, M. L. Nelson, and C. C. Marshall, Eds. ACM, 2006, pp. 53–54.

[10] G. Louppe, H. Al-Natsheh, M. Susik, and E. Maguire, "Ethnicity sensitive author disambiguation using semi-supervised learning." *CoRR*, vol. abs/1508.07744, 2015.

[11] L. Philips, "Hanging on the metaphone," *Computer Language*, vol. 7, no. 12 (December), p. 39, 1990.

[12] P. Treeratpituk and C. L. Giles, "Name-ethnicity classification and ethnicity-sensitive name matching." in *AAAI*, J. Hoffmann and B. Selman, Eds. AAAI Press, 2012. [Online]. Available: http://dblp.uni-trier.de/db/conf/aaai/aaai2012.html#TreeratpitukG12

[13] D. L. Word, C. D. Coleman, R. Nunziata, and R. Kominski, "Demographic Aspects of Surnames from Census 2000," Tech. Rep., 2000. [Online]. Available: http://www2.census.gov/topics/genealogy/2000surnames/surnames.pdf

[14] J. Fürnkranz, "A study using n-gram features for text categorization," Austrian Research Institute for Artificial Intelligence, Wien, Austria, Tech. Rep. OEFAI-TR-98-30, 1998. [Online]. Available: http://www.ofai.at/cgi-bin/tr-online?number+98-30

[15] J. Jacobs, "Finding words that sound alike. the soundex algorithm." pp. 473–474, 1982.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2012.

[17] M. Dunn, C. Vasquez Sandoval, I. Ibarra, and P. Saccomani. (2014) Theory of Knowledge - Areas of Knowledge. [Online]. Available: http://www.theoryofknowledge.net/areas-of-knowledge/

[18] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, ser. Pearson international Edition. Pearson Addison Wesley, 2006. [Online]. Available: https://books.google.fr/books?id=YHsWngEACAAJ

[19] The PostgreSQL Global Development Group. (2016) Postgresql 9.5.3 documentation. [Online]. Available: https://www.postgresql.org/docs/9.5/static/release-9-5.html

[20] The R Foundation. (2016) The r project for statistical computing. [Online]. Available: https://www.r-project.org/

[21] The PostgreSQL Global Development Group. (2016) Postgresql 9.5.3 documentation - fuzzystrmatch. [Online]. Available: https://www.postgresql.org/docs/9.5/static/fuzzystrmatch.html

[22] B. Grün and K. Hornik, "topicmodels: An R package for fitting topic models," *Journal of Statistical Software*, vol. 40, no. 13, pp. 1–30, 2011.

[23] D. Schmidt. (2016) ngram: Fast n-gram tokenization. R package version 3.0.0. [Online]. Available: https://cran.r-project.org/package=ngram

[24] M. van der Loo. (2014) The stringdist package for approximate string matching. [Online]. Available: http://CRAN.R-project.org/package=stringdist

[25] J. Oksanen, F. G. Blanchet, R. Kindt, P. Legendre, P. R. Minchin, R. B. O 'hara, G. L. Simpson, P. Solymos, M. Henry, H. Stevens, H. Wagner, and J. Oksanen. (2016) vegan: Community Ecology Package. [Online]. Available: https://github.com/vegandevs/vegan

[26] A. Liaw and M. Wiener. (2002) Classification and regression by randomforest. [Online]. Available: http://CRAN.R-project.org/doc/Rnews/

[27] T. Chen, T. He, and M. Benesty. (2016) xgboost: Extreme Gradient Boosting. [Online]. Available: https://cran.r-project.org/web/packages/xgboost/index.html

[28] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, "kernlab – an S4 package for kernel methods in R," *Journal of Statistical Software*, vol. 11, no. 9, pp. 1–20, 2004. [Online]. Available: http://www.jstatsoft.org/v11/i09/

[29] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010. [Online]. Available: http://www.jstatsoft.org/v33/i01/

[30] R Core Team and contributors worldwide. (2016) The r stats package. [Online]. Available: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/stats-package.html

[31] Web of Science Core Collection. [Online]. Available: http://apps.webofknowledge.com