Relational Machine Learning Author Disambiguation

Ekaterina Bastrakova*, Rodney Ledesma*, Jose Millan*, Fabien Rico†‡, Djamel Zighed‡
*Data Mining and Knowledge Management - Université Lumière Lyon II, Lyon, France
†Université Claude Bernard Lyon I, Lyon, France
‡ERIC Lab - Université Lumière Lyon II, Lyon, France
{ekaterina.bastrakova, rodney.ledesma, jose.millan-perez}@univ-lyon2.fr, fabien.rico@univ-lyon1.fr, zighed@univ-lyon2.fr

***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 signatures are from the same author 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 different instances of an author's name present in academic publications 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 any new author.**

I. Introduction

Author disambiguation is an open issue in the world of academic digital libraries. This is a difficult task due to different problems, including, among others, homonyms (authors with the same first name and last name), incomplete information (e.g. only the initials), misspelling or change of the author's name. This issue has become relevant in the past years due to an increasing quantity of 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 signatures present in the information about scientific papers retrieved from structured data bases such as Web of Science or Springer, that can be adapted to the source data schema presented in this work. 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 II we briefly review related solutions for author disambiguation. In Section III 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 IV we describe the implementation of the workflow. In Section V we present the obtained results, and finally in Section VI we discuss the conclusions and the future work.

II. Related work

The author disambiguation challenge has 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 method [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 similarity coefficient [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.

III. 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

ISBN 978-952-68397-8-3 (PDF)

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 II and using 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.

A. 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 Fig. 1, we define the minimal information required for our workflow to work. The main entities of the source data is described below.

- The main entity of our source data are the articles. This contains the basic information of a publication: title, journal where it was published, DOI and the publication year.
- More information about the article is found in the subject and keywords entities. In the case of the subject, it refers to the area of knowledge the article belongs to (mathematics, biology, social sciences, etc).
- The references entity stores the information about the articles that are referenced by this specific article.
- The signature entity contains the information of every 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 article.

It is important to have in mind that this source data structure is used for the workflow presented in this article. Therefore an ETL process has to be performed with the information of the articles coming from structured databases of scientific publications, in order to load them into our data source structure and use the proposed method.

B. Complementing features

Exploding the source data, we calculated 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 added the *focus name*, a phonetic representation based on the last name of the author; additionally we calculated the possible ethnicities the author may belong to based on the author's names; and finally we calculated a LDA-based topic (generated by Latent Dirichlet Allocation [11]) for every article we have. This features are described in more detail below.

1) Focus Name: A focus name of a signature is defined as the simplified version of the last name of the author. For example, the authors *C. Smith*, *A. Smit* and *G. Smoot* all share the same focus name: *SMT*.

The focus name is calculated by using a phonetic algorithm - Metaphone. This algorithm 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 [12].

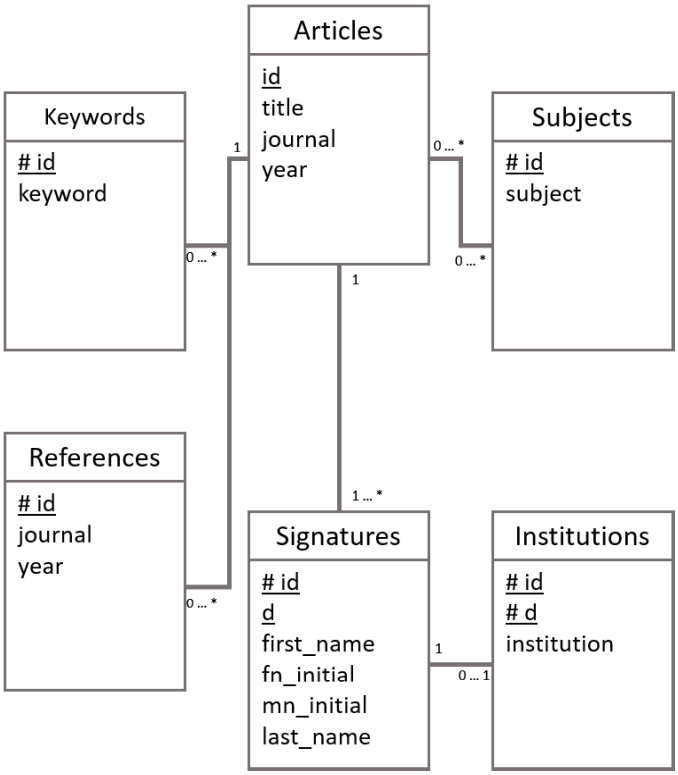

Fig. 1. Source Data Structure

The use of a focus name for the last name of the author leaves any language complexity and possible spelling errors aside, and at the same time, it helps to reduce different problems with the transliteration of their names. With this, authors that might have signatures with different transcriptions of names –where some letters can be written differently– can be grouped together and therefore, assigned to the same author.

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 similar works performed [10] [13], we use the dataset of Frequently Occurring Surnames from the Census 2000 [14] 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 of the different ethnicities that each last name can belong to. The possible ethnicities presented are:

- Percent of respondents 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, the ethnicity percentages and considering the related work of [13], we generated the bi-grams [15] derived from each individual last name, and complemented this information with a phonetic representation of the last name using the Soundex algorithm [16]. We used Soundex for this feature, instead of Metaphone for the Focus Name, as it returns a numerical representation of the last name (together with a letter) which is incorporated to the model and helps to predict the Ethnicity features. Then we created several Support Vector Machine models that predicted if a given last name belongs to a specific ethnicity or not. This implies that to our set of Complementing Features, we added 6 new Ethnicity features, each one related to a specific ethnicity present in the Census data, as demonstrated in Table I.

TABLE I. COMPLEMENTING ETHNICITY FEATURES

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

3) LDA Topic: Considering that it is very likely that an author writes articles in the same field, we exploited this to determine if two signatures are the same. For this, we calculated the feature "LDA Topic". This is a topic generated by Latent Dirichlet Allocation [11].

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

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.

TABLE II. FREQUENT TERMS IN MOST FREQUENT LDA TOPICS

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"

C. Similarity features

Having the complete set of features (from the source data and the complementing features described in Section III-B), then we calculated 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 III.

For the case of the First Name Initial, the Second Name Initial and the LDA Topic, we indicated 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 calculated the absolute difference. Finally for the Keywords, the Journals References, the Subjects, the Title, the Coauthors of the article and the Ethnicities of the signature's last name, we calculated the Jaccard similarity coefficient. For the last one, a more in depth explanation can be found in Section III-C1 below.

TABLE III. FEATURES CALCULATED FOR EVERY PAIR OF AUTHORS

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

1) Distance-based Features - Jaccard Similarity: As a measure to indicate how close are two articles related according to the information we have from the source data, we used the Jaccard similarity coefficient, which computes the similarities of asymmetric information on binary attributes. Equation 1 shows how the Jaccard Similarity Coefficient is calculated.

$$J_{ij} = \frac{p}{p+q+r} \tag{1}$$

where J_{ij} is the Jaccard similarity coefficient, 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 calculated them as described below:

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

D. Process flow

The process flow of our solution can be appreciated in Fig. 2. Having the source data, as described in Section III-A, 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 III-B.

After this, we group the signatures (together with their corresponding article information) by focus name, and process each of this groups independently (in parallel). 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 III-C.

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 applied: 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 used hierarchical clustering, whose results are the corresponding clusters for every signature in the focus name. Every cluster that is calculated here represents a single disambiguated author.

IV. Implementation

The source code of the implemented workflow, along with the database schema and source data, are publicly available online in our GitHub repository (https://github.com/DMKM1517/author_disambiguation).

We used publicly available data to test this implementation. The dataset was retrieved from the Web of Science website [19], and then loaded into our source data schema (described in III-A), using a small ETL process to transform the source structure to ours. The articles used for this was a collection of scientific papers coming from a wide range of different subjects (from computer science, mathematics and physics to music, history and biology).

Having this we manually disambiguated signatures from unique authors from the Web of Science dataset. The result of this process was a dataset of 1330 signatures, with their corresponding article information, that corresponded to 236 real authors.

As previously mentioned, the implementation of the presented work was made by using a combination of both a relational database, specifically PostgreSQL 9.5 [20], for storing and handling the data, and R 3.2.3 [21] as the programming language for implementing the models and calculating the disambiguated clusters. This approach takes 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 was calculated using the Metaphone algorithm, present in the *fuzzystrmatch* contrib library of PostgreSQL [22].
- For calculating the LDA Topic we used the *topicmodels* R package [23] with $k = 8$ as mentioned in Section III-B3.

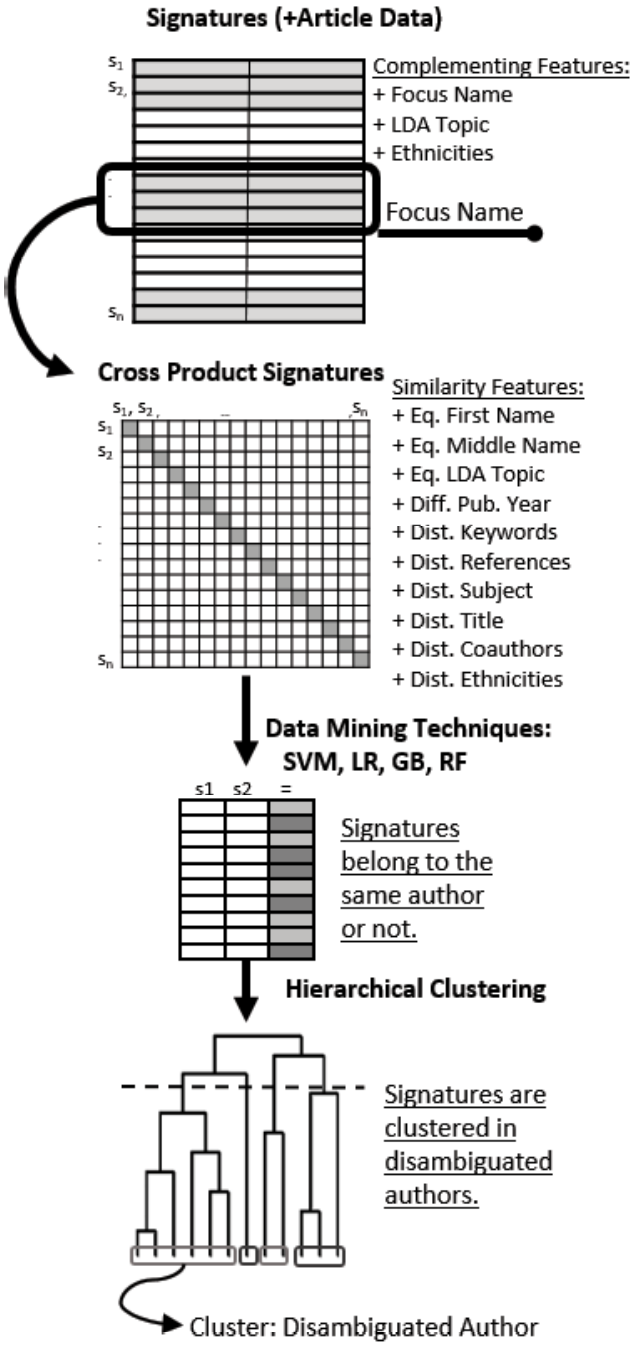

Fig. 2. Author Disambiguation Process Workflow

- For calculating the Ethnicity Complementing Features, we generated bi-grams using the *ngram* R package [24]. Similarly, in order to generate the Soundex phonetic representation of the last names we used the *phonetic* function present in the *stringdist* R package [25].
- The Jaccard coefficient was calculated using the *vegdist* function existing in the *vegan* R package [26].
- The Random Forest model was created using the *ran-*

domForest R Package [27] with the default parameters.

- The Gradient Boosting model was created using the *xgboost* R Package [28], Extreme Gradient Boosting, which is an efficient implementation of gradient boosting framework.
- The Support Vector Machine model was created using the *kernlab* R Package [29], which performs a kernel-based machine learning SVM implementation for classification. In the different tests we performed, we found the *besseldot* kernel, with the parameters of $C = 100$ and $\sigma = 1$, as the best configuration for this classification problem.
- For the Logistic Regression model we used the *glmnet* R Package [30], 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 used the *hclust* function existing in the *stats* R core package [31], using the *complete* method.

V. 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 a disambiguated author corresponds to the real cluster.

For the first step, along 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 Machine (SVM), as described before.

For training the models, we created the training and testing sets with different focus name groups in order to eliminate bias within the sets (the information within a focus name group is highly correlated, due to the cross product of all the signatures). We divided the set of focus names so 70% of them were assigned for training and 30% for testing, which assured that our testing set contained no information that was already trained beforehand. The results for the first step validation can be appreciated in Table IV.

TABLE IV. First validation

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

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 98.60% of F1-score in the testing set.

Furthermore, the *xgboost* R package [28] provides a function to look at the gain of the features. Fig. 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.

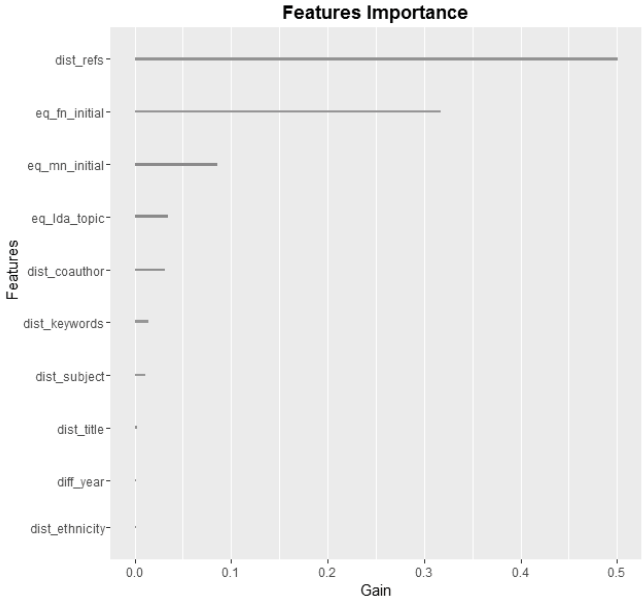

Fig. 3. Features Importance for GB

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 built a distance matrix for every author inside a focus name and then ran a hierarchical clustering process that gave us the corresponding cluster of each author that was processed.

As commonly performed in author disambiguation research, we evaluated the predicted clusters over testing data using both pairwise (Equations 2, 3 and 4) and B3 (Equations 5, 6 and 7) precision, recall and F-measure.

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

$$Recall_{Pairwise} = \frac{|p(R) \cap p(C)|}{|p(R))|} \tag{3}$$

$$F1_{Pairwise} = \frac{2 * Precision_{Pairwise} * Recall_{Pairwise}}{Precision_{Pairwise} + Recall_{Pairwise}} \tag{4}$$

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

$$Recall_{B3} = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|R(S_i) \cap C(S_i)|}{|R(s_i)|} \tag{6}$$

$$F1_{B3} = \frac{2 * Precision_{B3} * Recall_{B3}}{Precision_{B3} + Recall_{B3}} \tag{7}$$

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 V. 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.

TABLE V. SECOND VALIDATION

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

VI. 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 V, 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. On the other hand, the ethnicity features, along with the year and title of the article, do not provide relevant information for the presented disambiguation process. Further investigation is required in order to improve the contribution of the ethnicity features for the presented workflow.

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 poorest 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] 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.

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

[13] 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

[14] 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

[15] 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

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

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

[20] 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

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

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

[23] 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.

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

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

[26] 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

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

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

[29] 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/

[30] 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/

[31] 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