Feature Extraction for the Author Name Disambiguation Problem in a Bibliographic Database

Jorge M. B. Silva
CRACS/INESC TEC
Rua do Campo Alegre 1021/1055
Porto, Portugal
jmbs@inesctec.pt

Fernando Silva
CRACS/INESC TEC & Universidade do Porto
Rua do Campo Alegre 1021/1055
Porto, Portugal
fds@dcc.fc.up.pt

ABSTRACT

Author name disambiguation in bibliographic databases has been, and still is, a challenging research task due to the high uncertainty there is when matching a publication author with a concrete researcher. Common approaches normally either resort to clustering to group author's publications, or use a binary classifier to decide whether a given publication is written by a specific author. Both approaches benefit from authors publishing similar works (e.g. subject areas and venues), from the previous publication history of an author (the higher, the better), and validated publication-author associations for model creation. However, whenever such an algorithm is confronted with different works from an author, or an author without publication history, often it makes wrong identifications.

In this paper, we describe a feature extraction method that aims to avoid the previous problems. Instead of generally characterizing an author, it selectively uses features that associate the author to a certain publication. We build a Random Forest model to assess the quality of our set of features. Its goal is to predict whether a given author is the true author of a certain publication. We use a bibliographic database named Authenticus with more than 250,000 validated author-publication associations to test model quality. Our model achieved a top result of 95.37% accuracy in predicting matches and 91.92% in a real test scenario. Furthermore, in the last case the model was able to correctly predict 61.86% of the cases where authors had no previous publication history.

CCS Concepts

•Computing methodologies → Supervised learning by classification; Feature selection;

SAC 2017,April 03-07, 2017, Marrakech, Morocco

http://dx.doi.org/10.1145/3019612.3019663

Keywords

Author Name Disambiguation, Author Identification, Supervised Learning, Feature Extraction, Random Forests

1. INTRODUCTION

Since 2010 there are more than 50 million articles published and each year more than 1 million new ones are added to this total. Furthermore, the number of published papers has increased at a 2.5% rate each year [12]. The exorbitant number of new publications, along with their metadata heterogeneity, utterly increased the difficulty of creating and maintaining reliable bibliographic databases [5]. The problem aggravates in the case of author identification. The same identity (or author) name can be stored in several formats and written in different forms. Moreover, distinct identities may have the same name. This problem is known as author name disambiguation and is the hardest challenge faced by current bibliographic databases such as DBLP, CiteSeer, MEDLINE, BDBComp [6, 4].

Author disambiguation in a bibliographic database is necessary primarily for two reasons. First, it allows users to search for publications of a particular author without having to manually filter incorrect results. Second, disambiguating author names improves the accuracy of bibliometric analysis [15]. This last point is particularly important for publicly funded research organizations where bibliometric analysis serves towards three principal objectives: production efficiency incentive, funding allocation, and research contribution demonstration. Due to the difficulty of the task, research organizations appoint experts to manually disambiguate author names in their publications [4]. Additionally, some relevant online databases, such as DBLP, still rely significantly on manual intervention.

Author name disambiguation approaches are categorized in two groups: author grouping and author assignment [5]. The first clusters publications according to the individual identity of authors. For example: partitioning the publications of 10 distinct authors named "J. Smith" from a set of 100 publications. The latter treats the problem as a binary classification problem which consists in deciding whether a publication is truly written by a specific author. In this paper we focus on the author assignment category. The common methodology in these algorithms consists in capturing the general characteristics of publications of an author a and use them to determine whether a publication p is similar

enough to be considered written by a. There are two approaches for this: use features of past publications to train a supervised model to identify publications of specific authors (in this case each label in the training set uniquely identifies an author) [9], or using a function that generalizes the features of previous publications of an author and compare them with the features of a new publication (in some cases the comparison function is also a supervised learning algorithm) [4]. These approaches have some drawbacks [5, 16], namely they assume that authors generally write similar papers (e.g. in terms of subject areas and publication venues), and also require a validated publication history of an author.

In this paper we focus on the possibility of extracting features that are not general to all previous publications of an author a, but that relate him to a publication p. We aim at using these features in a supervised learning classifier to determine whether p is written by a. By not representing the general characteristics of a, we avoid the fallacy that authors always publish similar papers. Furthermore, we aim to produce a model where the training phase does not require training data from author a in order to correctly identify the authorship of his publications. As a result, we also avoid the other problem of requiring previous history. We use a public bibliographic database for Portuguese researchers named Authenticus [1] to conduct our work.

This paper is structured as follows: in Section 2 we review the state of art for the author name disambiguation problem. In Section 3 we describe the information available in the Authenticus database. In Section 4 we define our machine learning approach and detail the feature extraction process. In Section 5 we test feature relevance and model performance. Finally, in Section 6 we present the conclusions and discuss future work.

2. RELATED WORK

Author grouping and author assignment methods are the two main types of strategies for author name disambiguation [13]. The former methods cluster publications based on properties such as co-authors, institutions, and keywords. They rely on a pairwise similarity score between papers obtained by a function which can be pre-defined, or estimated using supervised learning. Some examples of pre-defined similarity functions are: rule-based scoring using bibliometric metadata related to authors and their publications [3], co-authorship network analysis [13], and similarity coefficients such a Jaccard Index and cosine, directly applied to some features [14]. Using supervised learning, Huang et al. [11] trained a SVM classifier to estimate the distance between two publications. Treeratpituk et al. [15] built a Random Forest classifier for the same purpose. Gurney et al. [8] constructed a logistic regression model to create a network of publication/author nodes where edges weights represent the probabilistic value of the two nodes being the same person. Then, they employ a community detection algorithm over the network to discriminate the pairings of nodes in relation to unique authorship.

Author assignment methods use former knowledge about the author to directly predict whether he published a certain paper. These methods can also use either supervised or unsupervised learning. The latter approaches cluster the characteristics of authors. To solve the problem, they initially assume that a publication p was written by a set of authors A. Then, the likelihood of each author having written p is estimated by recurring to publication's attributes in a hierarchical way. The crucial point in these methods is to define the hierarchical path. Han et al. [10] adopted the Expectation Maximization algorithm for the purpose, while in another example, Bhattacharya et al. [2] used a Naive Bayes model. Supervised learning techniques generalise characteristics of authors into a feature set, then a classifier is built to predict whether a publication fits the characteristics of an author. Han et al. [9] trained two classification algorithms (SVM and Naive Bayes) to identify authors of publications using as information: co-author names, work titles and publication venues. The labels in their training set uniquely identify publication's authors. D'Angelo et al. [4] statistically estimated groups of features that generalize the characteristics of previous publications of authors, and then define four filters that infer the author of a publication. Wand et al [16] applied the same feature extraction methodology with a Boosted-trees classification algorithm to predict authorship. In [7], Veloso et al. proposed SLAND an approach that uses a supervised rule-based associative classifier to infer the author of a publication. The association rules were demand-driven and the strategy also has a first phase which consists in using an unsupervised learning method to define the training data for the model construction.

In overall, the results reported in the cited works are great (on average, they report around 90% accuracy with the best ones achieving 99.6%). Nevertheless, in most cases the approaches are tested in specific datasets that do not reflect the conditions in a real-scenario application [8]. There are some critical difficulties that emerge in a real scenario. First, authors do not always write similar publications. This is a common difficulty for most of the cited works since they mostly use pairwise publication comparison [1], or generically represent author's characteristics to infer authorship. Second, most of the supervised learning approaches mentioned require training data from any author whose authorship is to be determined [5]. Since validated data from all the authors is hard to obtain, this becomes a problem in a real-scenario. Finally, there are typographical errors and missing information in metadata that may compromise the feature extraction process.

Our proposed method differs from those cited as it extracts features that estimate the likelihood of publication p being written by author a without focusing on all previous publications of a, but rather on specific ones that are similar to p. To the best of our knowledge, the closest work related to our feature extraction strategy is [7]. They adopt association rules in such a way that the features relate the author to the publication. Additionally, they also do not need author specific data to train the model. However our work is still different, since they use a demand-driven rule extraction to obtain features for authors and we define a set of generic features for any author instead.

[1] in the clustering case this is known as transitivity problem [16].

3. AUTHENTICUS DATABASE

The Authenticus project aims at facilitating and improving the results of bibliometric analysis performed by research organizations in Portugal. For the purpose they built a bibliographic database for Portuguese researchers which strives for both automation and accuracy. The database is periodically updated with new publications by querying sources such as: ISI Web-of-Knowledge, Scopus, Crossred and DBLP. It currently has 402,122 publications and 83,182 distinct authors (or researchers). We are using a subset of the database that includes 164,576 publications with at least one of its authors validated, and 10,428 researchers (authors) with at least one publication validated. This adds to a total of 298,800 author-publication associations validated. Lack of manual validated data is often a problem in an author name disambiguation task [13, 3] and it is the main reason why classification techniques are generally ruled out. In our case, since we use the Authenticus database we have a considerable amount of data to train a model with our features and to properly assess its quality.

The database contains multiple tables with information about publications and researchers. Some of the details stored in the database for each publication are: authors' names and emails, affiliations, publication venue, keywords, research areas and times cited. For each researcher profile, the database stores the information about name, emails, affiliations history and known publications. Through the connection between the researchers and publications, several tables are created with statistics for researchers. Some examples are: keywords used, times cited, published in which journals and conferences. We use most of this information to extract features for the classification model.

4. MACHINE LEARNING APPROACH

4.1 Problem Description

We formalize the problem of author disambiguation as the task of receiving a publication p with n authors $\{a_1, a_2, \ldots, a_n\}$, and matching them with n researchers $\{r_1, r_2, \ldots, r_n\}$ from the Authenticus database (D). Each author, a_i has a metadata record m_{a_i} consisting of a set of attributes retrieved from p's metadata. Some of its attributes such as journal, keywords and research areas are common to all authors, while information such as author name, email and affiliation are specific to each author. At the first stage we use a function Ω, such that $\Omega(m_{a_i}, D) = C_{a_i}$, where C_{a_i} is a list of researchers from D that are candidates for the matching with a_i and $\exists!\ c \in C_{a_i}$ such that c is the correct match for a_i. Moreover, each candidate $c \in C_{a_i}$ has a history record h_c, consisting of a set of attributes (e.g. keywords, scientific areas and co-authors) from the researcher's previous history of publications. Our goal is to use the available metadata to extract features that, when applied to a classification model, have the ability to identify the true author in a list of candidates for an author position. More concretely, we want to build a function Θ, such that $\Theta(C_{a_i}, m_{a_i}, C) = r$, where C is a list containing the lists of candidates for the other author positions and r is one of the true authors of the publication.

The Ω function in this context is known as blocking function and it is used to filter candidates for an author position. Therefore, drastically decreasing the number of computations required. In this paper we use email and name matching to select the candidates. The Θ function consists of two parts. A δ function that builds a feature set F_c relating the history of a researcher (h_c) not only with the metadata associated with the publication author (m_{a_i}) but also with other candidates (C), and a classification model M which given F_c, predicts the probability of c being the correct identification for author a_i.

Feature Group	# of Features
Name matching	2
Email matching	1
Co-authorship	5
Keywords	5
Research areas	2
Journal	1
Affiliations	4
History statistics	2

Table 1: Features grouping in our dataset.

4.2 Feature Extraction

In this section we describe our function δ which creates the feature set F for an instance of the author name disambiguation problem. We use a total of 22 features. Since some of the features are similar we categorized them in 8 groups. Table 1 names these groups and the number of features in each one.

We now discuss what each feature (or group of features) represents and the process for its extraction. For the sake of understanding, we exemplify the feature extraction considering the case of a researcher (r) that is a candidate for author position a_i in publication p. Moreover, we have a list C which consists of all the candidates lists for all author positions in p and a metadata record m_{a_i} associated to a_i which consists in some author (name, email, affiliation) and publication (keywords, journal, citations, research areas) information.

$$\begin{aligned} total &\leftarrow \Pi_{total}\Gamma_{sum(attr)}\sigma_{researcher=r} \\ id, score &\leftarrow \Pi_{id,attr/total}\sigma_{researcher=r} \end{aligned} \quad (1)$$

Names matching: author's names are usually written with several initials and the last name in complete form. The two features in this group are binaries. The first is set to 1 in case that the initials in the author's name matches perfectly (in order and values) with the full name of researcher r. The second is 1 when the researcher's last name is fully written in the authors name.

Email matching: this feature represents the best score of the comparison between the author email in the publication and the emails stored for researcher r. For the purpose we use Jaro Winkler string comparison algorithm which returns a value between 0 and 1.

Co-authorship: features in this group estimate the likelihood of r in having co-authored with any of the candidates for the other author positions. We extract the features by assigning a probability score to each researcher that has already co-authored with r (we apply formula (1) to the

database to calculate this score). Then, we iterate through the candidates lists in C and for each one, we select the highest probability score among the candidates. Finally, we use the 5 top scores as features.

Keywords: features in this category estimate the probability of r using some of the keywords in p. Using formula (1) we extract the scores of the keywords already used by r. Then we match them with the ones in p and we return the top 5 scores as features.

Research areas: research areas help identifying publication's topics. Therefore we also estimate the likelihood of r in having published in the scientific areas of p. We query this information from the database (using formula (1)) and we get the top 2 scores as features.

Journal: in case the publication was published in a journal (i.e. conference papers not considered) we estimate the likelihood of r publishing in the same journal. More concretely, we count the number of times r published in the referred journal, and divide by his total number of publications.

Affiliations: this category consists in 4 binary features that relate r's affiliation history with the ones in the publication. The first feature is 1 in case the affiliation metadata in m_{a_i} is specifically related to a_i instead of a global affiliation for all the authors. For the remaining three features we use the affiliation history of r up until two years before p was published. The second feature is 1 in case r worked in that affiliation, the next is 1 in case r worked in the same city and the last one is 1 if he worked in the same country.

History statistics: these two features are based on general statistics of r's publications history. The first is the module of the difference between the average number of co-authors in r's publications and the number of authors in p. The second one is the difference since the year of r's last publication and the year p was published.

Often, it is not possible to extract all the mentioned features. For example, for publications with 3 keywords we cannot extract 5 keywords features. In these cases, we extract 3 features and we set as "-1" the remaining two. The "-1" value represents lack of information about some features and therefore we do not attempt to correct them with traditional missing values approaches during the training phase of the classification algorithm. Additionally, we scaled the features (exception to history statistics group) between 0 and 1, to define a maximum score for the feature which is known as an important step for classification models.

4.3 Classification Algorithm

Our surveyed related work revealed that the best results for author name disambiguation using classification algorithms were obtained using a Random Forest classifier and thus we also adopt such classifier. This algorithm is an ensemble learning method consisting of N different Decision Trees, where N is user defined. Each Decision Tree adjusts the features into a tree topology where the leafs are the classification decisions (known as labels) and nodes represent tests for a certain feature value. Classification starts at the tree's root, then a walk-through the nodes occurs depending on the value of the feature at each node's condition. When a leaf is reached, we have a prediction for the label. In a Random Forest every Decision Tree classifies a case, and each prediction is a vote. In the end, there is a count and the label with most votes wins. It is possible to obtain the classification confidence by dividing the number of votes on the winning label by the total number of trees.

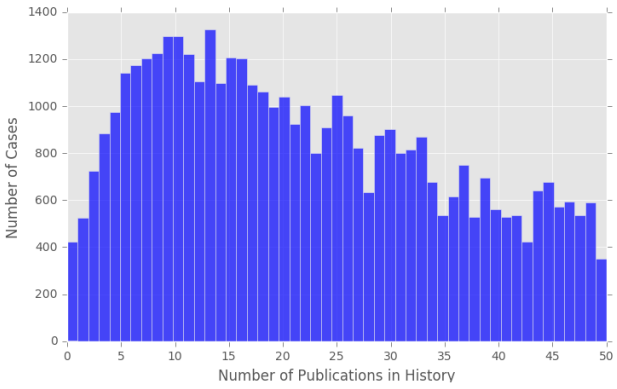

Figure 1: Frequency of positive case previous history

5. EXPERIMENTS

To evaluate the performance of the Random Forest classifier with our set of features, we first divided the Authenticus database into two groups: researchers' history and testing data. The first one serves as a knowledge base of previous publications from the researchers, while the other one represents publications we use to assess the model quality. To balance the number of publications the researchers have in their history versus the number of publications they have in the testing data, we divided the groups by publications dated before 2012 (inclusive) and after 2012. From the testing data, in order to decrease the computational time of our tests we randomly selected 40,000 publications. Then, we used our feature extraction function to create a dataset with the known validated authors labelled as a "match" (positive case). Furthermore, for each validated author, we picked a maximum of 4 incorrect candidates[2] and labelled them "no match" (negative case). To select 4 candidates from the whole list, we calculated the correlation between the features of each candidate and the ones from the true author using Pearson Correlation Coefficient and we chose the top results. Thus, these are potentially the worst cases for the model evaluation. The dataset has a total of 73,591 positive cases and 229,738 negative ones. Moreover, there are 8641 distinct researchers. Each positive case has an average of 65 publications as knowledge base (previous history). However, the dispersion of the values is extremely high (78 standard deviation) and we still have enough cases with low amounts of previous history. Figure 1 illustrates the distribution of cases depending on the number of publication in history. To ease visualisation we separated cases where the value is higher than 50 (30,255 cases). The dataset has 21,170 cases where the amount of history is relatively low (less than 20 publications). Additionally, we have 421 cases where the

[2]a maximum of 4 false authors per case were chosen to ease results interpretation and class balancing.

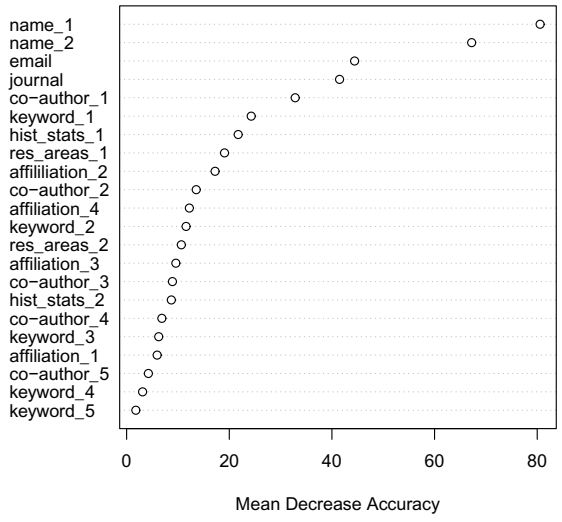

Figure 2: Feature rlevance in Random Forest Classifier (MDA)

true researcher has no history at all which is the worst case scenario for any author name disambiguation algorithm.

5.1 Feature Relevance

To estimate individual feature importance we trained a random forest classifier using the whole dataset and calculated the Mean Decrease Accuracy (MDA) of each feature. This test randomizes the order of values of a feature f and then determines model accuracy (i.e. associates the value of f in case i to a random case j). Thus estimating the relevance of f in the dataset. The higher a feature's MDA is the more important it is. Figure 2 illustrates the individual MDA scores.

The name matching features MDA (80.5% and 67.23%) is much higher than the other features. Such difference is explained because they are binary values, therefore there is not a smooth transition between values. This does not mean that they are not important, for example, affiliations group also consists of binary features but since they do not impact accuracy as much (are not so relevant) they have much lower scores. Email is the third most important feature (44,4%) which is expected since when this information is available it works almost as a unique identifier of the author. In fourth place the journal feature (41.5%) was a surprising result especially since its score difference when compared to similar features (co-author, research areas and keywords) is high. The remaining groups (exception to the last 3 features) have a relatively good importance with the values ranging from 5% to 32%. We highlight the tendency of groups where features are sorted by score (co-authorship, keywords, research areas). Within these groups, feature importance follows the same numerical order of the group. This is not only justified by the fact that the top feature is more important, but also because often we are not able to get all the features for a group, therefore the lowest rank features in a group are "-1" in several cases. This is particularly noticeable in the 3 worst features (4.2%, 3.1% and 1.7%) because most publication do not have at least 5 authors nor 4 or more keywords.

#	TPR	TNR	F1-Score	Accuracy	AUC
3	80.014%	93.004%	36.081%	80.771%	81.446%
4	88.534%	**93.862%**	73.226%	89.358%	90.148%
5	93.896%	89.635%	84.699%	92.967%	94.624%
6	94.994%	91.143%	87.398%	94.129%	95.665%
7	95.419%	88.685%	87.068%	93.839%	95.483%
8	95.984%	90.130%	88.699%	94.604%	96.593%
9	96.344%	90.365%	89.416%	94.919%	97.112%
10	96.462%	90.765%	89.802%	95.105%	97.191%
11	96.534%	90.785%	89.932%	95.162%	97.271%
12	96.543%	90.959%	90.028%	95.212%	97.289%
13	96.579%	91.030%	90.122%	95.256%	97.315%
14	96.580%	90.979%	90.100%	95.244%	97.329%
15	96.585%	90.987%	90.112%	95.249%	97.331%
16	96.661%	91.213%	90.345%	95.361%	97.626%
17	**96.686%**	91.114%	90.341%	95.354%	97.619%
18	96.663%	91.129%	90.309%	95.341%	97.632%
19	**96.686%**	91.189%	**90.377%**	**95.373%**	97.629%
20	96.684%	91.196%	90.376%	**95.373%**	97.627%
21	96.686%	91.152%	90.359%	95.364%	97.623%
22	96.666%	91.204%	90.349%	95.362%	**97.645%**

Table 2: $M \times N$ cross-validation results changing the number of features. # - Number of features. TPR - True Positive Rate. TNR - True Negative Rate. AUC - Area Under Curve

5.2 Model Evaluation

To assess the quality of the random forest classifier we used a $M \times N$ cross validation strategy to decrease overfit in our results. In more detail, we divide the dataset in 10 folds (M folds), and then each one in 10 more parts (N folds). For every M fold we test each N fold by using $M \setminus N$ to train the model and N to evaluate it. We use stratified folds thus the number of positive and negative cases is the same in all folds. As a result, each test evaluates 735 positive cases and 2,973 false ones.

We tested model quality using different subsets of features. Following the MDA ranks we started with the whole group and after each test we removed the worst feature until there were only 3 features left. Table 2 demonstrates the results of our test.

The subsets of features with 9 or more features have similar scores in every assessed metric. The difference between the highest score and lowest among these group is 0.3% for true positive rate (TPR), 0.9% for true negative rate (TNR), 0.9% for F1-score, 0.4% for accuracy and 0.5% for area under the curve (AUC). The metrics decline for subsets with less than 9 features. Exception to the TNR score whose best score by a margin of more than 2.6% is in the subset with only four features (2 from name matching, email and journal). Nevertheless, the TPR drastically decreases for this subset (more than 8.0%). In overall the subset with 19 features (consists in removing the last two features of keywords and the last one from co-authorship groups) is the best to train a Random Forest classifier.

Our previous test demonstrated that the set of features extracted achieves a top accuracy of 95.373%. This score is related to the ability in predicting whether a researcher is

Fold	Cases	BF	MF	Correct	MA	GA
0	7396	216	416	6764	94.21%	91.45%
1	7263	181	413	6669	94.17%	91.82%
2	7526	198	416	6912	94.32%	91.84%
3	7402	180	425	6797	94.12%	91.83%
4	7337	194	365	6778	94.89%	92.38%
5	7373	208	424	6741	94.08%	91.43%
6	7331	185	389	6757	94.56%	92.17%
7	7353	218	369	6766	94.83%	92.02%
8	7312	163	418	6731	94.15%	92.05%
9	7298	173	393	6732	94.48%	92.24%
Total	**73591**	**1916**	**4028**	**67647**	**94.38%**	**91.92%**

Table 3: Real scenario test results. BF - Blocking method Fail, MF - Model Fail, MA - Model Accuracy, GA - Global Accuracy.

the true author of a publication. However, this information is not enough to solve the problem described in section 4.1. Consider that in a list of candidates for an author position, more than one researcher is labelled as a match[3], in these cases we require a tiebreaker to select only one. For the purpose we use the confidence of the Random Forest classifier. To assess the accuracy in selecting the correct researcher from a list of candidates we divided the 40,000 publications in our dataset (D) into 10 folds. We test each fold K by training the Random Forest with $D \setminus N$ and evaluating it using the publications p in K. For each p we create lists of candidates for each author position and apply the classifier in order to select the match with highest confidence. Then we evaluate the results for those cases where data is validated. Table 3 demonstrates test results.

Our dataset includes 73,591 author-publications confirmed pairs that we can verify the correctness of our model selection. In 1916 cases our blocking method failed to select the correct researcher as a candidate due to misspelling errors in the author's name. For the remaining 71,675 cases, our model assigned the highest confidence of matching to the correct researcher in the candidates list, 67,647 times (94.38% accuracy). Considering the errors of the blocking method, in general we predicted the correct researcher 91.92% of times.

On average the candidates list for the correct cases had 134 candidates (375 standard deviation (std)), while the wrong cases had 481 (756 std). The average confidence score of the correct cases was 0.92 (0.18 std) and 0.50 (0.36 std) for the other ones. Additionally, the average difference of scores between the true author and the selected candidate was 0.23 (0.25 std). Figure 3 illustrates the frequency of best scores estimated for the wrong cases and the difference of that score and the one from the correct candidate. In most of the wrong cases, the difference of scores leading to a wrong decision is smaller than 0.2. In fact in almost 1200 cases the difference was smaller than 0.05. Thus, indicating that when the model fails, the true author still had a close score to the candidate selected.

To estimate the ability of the model in predicting authors

[3]the same problem occurs in the case all candidates are labelled as no match.

Figure 3: Score analysis for the wrong model selection.

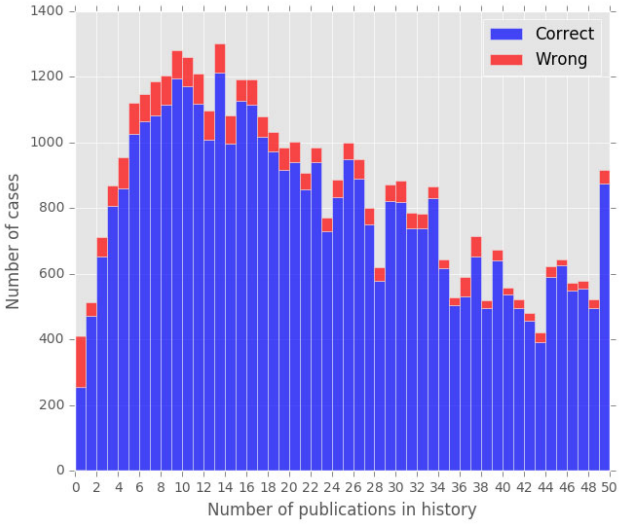

Figure 4: History analysis for correct and wrong cases.

with few past publications or even none, we calculated the number of correct and incorrect cases evaluated by the model considering the number of publications the true author had in his history. Figure 4 illustrates the results of this study. To ease visualisation we removed cases with more than 50 publications in history. For these cases, we had 28,033 correct predictions and 1190 wrong ones (4.1% error rate). On average the model incorrectly predicts 6.8% of the cases when the number of publications is fewer than 50. The model is much more likely to fail when the author has 0 publications. Still for this case the model correctly predicted 253 cases out of 409 (62%). For the remaining number of publications the model had very similar results, ranging for a minimum of 2.7% wrong cases to a maximum of 9.8%. Therefore, we can assume as long as the true researcher has at least one publication in his history, the performance of the model is very reliable. For 0 publications in history, the percentage of correct predictions is still reasonably good since this represents the worst case scenario.

6. CONCLUSIONS & FUTURE WORK

In this paper we addressed the author name disambiguation problem. We avoided the common problems of generalisation of these algorithms by defining a set of features that relates an author to a specific publication, instead of summarizing his global characteristics. To determine the quality of the features extracted we used them to build a Random Forest model that predicts whether a researcher in the Authenticus database matches the profile of an author in a publication. We started experimentation by using the Mean Decrease Accuracy to determine individual features rank. Then, using this rank we performed feature selection by testing several subsets of features. We concluded that we could remove 3 of our initial features and that the model obtained was able to achieve 95.37% accuracy in predicting a match. Furthermore, we tested the selected model in a real case scenario where the Random Forest model in 94.38% of the times was able to select the correct researcher from the candidates list (when the true author was in the candidates list, 91.92% for all cases). Therefore proving that the features extracted are able to create a classifier suited for the author name disambiguation problem. We demonstrated that when our model fails, the score assigned to the true researcher is close to the selected candidate in most cases. Furthermore, we confirm the ability of the algorithm in correctly predicting most cases independently of the number of publications in the history of the author. Additionally, we were able to find the correct candidate for 62% of the cases where authors had no history.

There are some steps we must complete before deploying this set of features as a solution for the real problem. First we have to test other classification algorithms such as Naive Bayes, SVM or ensemble methods. Second, we need to further improve the predicting ability for authors with no history, if necessary we may have to create another model just for these cases. Finally, we have to improve our blocking function to deal with the misspellings. This function represented a global error of 2.40%.

7. ACKNOWLEDGMENTS

This work is partially funded by the ERDF through the COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT as part of project UID/EEA/50014/2013. The authors are thankful to Sylwia Bugla and Fábio Domingues for their work on Authenticus data set.

8. REFERENCES

[1] P. Authenticus. Authenticus bibliographic website. https://www.authenticus.pt/en/home/view_article/1. Accessed: 2016-07-04.

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SDM*, volume 5. SIAM, 2006.

[3] E. Caron and N. J. van Eck. Large scale author name disambiguation using rule-based scoring and clustering. In *Context counts: Pathways to master big and little data. Proceedings of the STI conference*, pages 79–86, 2014.

[4] C. A. D'Angelo, C. Giuffrida, and G. Abramo. A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments. *Journal of the American Society for Information Science and Technology*, 62(2):257–269, 2011.

[5] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender. A brief survey of automatic methods for author name disambiguation. *Acm Sigmod Record*, 41(2):15–26, 2012.

[6] A. A. Ferreira, R. Silva, M. A. Gonçalves, A. Veloso, and A. H. Laender. Active associative sampling for author name disambiguation. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 175–184. ACM, 2012.

[7] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. Laender. Effective self-training author name disambiguation in scholarly digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 39–48. ACM, 2010.

[8] T. Gurney, E. Horlings, and P. Van Den Besselaar. Author disambiguation using multi-aspect similarity indicators. *Scientometrics*, 91(2):435–449, 2012.

[9] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Digital Libraries, 2004. Proceedings of the 2004 joint ACM/IEEE conference on*, pages 296–305. IEEE, 2004.

[10] H. Han, W. Xu, H. Zha, and C. L. Giles. A hierarchical naive bayes mixture model for name disambiguation in author citations. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1065–1069. ACM, 2005.

[11] J. Huang, S. Ertekin, and C. L. Giles. Efficient name disambiguation for large-scale databases. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 536–544. Springer, 2006.

[12] A. E. Jinha. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, 23(3):258–263, 2010.

[13] F. Momeni and P. Mayr. Using co-authorship networks for author name disambiguation. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, pages 261–262. ACM, 2016.

[14] L. Tang and J. P. Walsh. Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps. *Scientometrics*, 84(3):763–784, 2010.

[15] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 39–48. ACM, 2009.

[16] J. Wang, K. Berzins, D. Hicks, J. Melkers, F. Xiao, and D. Pinheiro. A boosted-trees method for name disambiguation. *Scientometrics*, 93(2):391–411, 2012.