Author identification in bibliographic data using deep neural.pdf

by

TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 3, June 2021, pp. 911~919
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i3.18877

Author identification in bibliographic data using deep neural networks

Firdaus[1], Siti Nurmaini[2], Reza Firsandaya Malik[3], Annisa Darmawahyuni[4], Muhammad Naufal Rachmatullah[5], Andre Herviant Juliano[6], Tio Artha Nugraha[7], Varindo Ockta Keneddi Putra[8]
[1, 2, 4-8]Intelligent Systems Research Group, Universitas Sriwijaya, Palembang 30137, Indonesia
[3]Communication Networks and Information Security Research Lab, Universitas Sriwijaya, Palembang 30139, Indonesia

Article Info

Article history:

Received Jun 15, 2020
Revised Aug 23, 2020
Accepted Aug 31, 2020

Keywords:

Author name disambiguation
Bibliographic data
Deep neural networks
Homonym
Synonym

ABSTRACT

Author name disambiguation (AND) is a challenging task for scholars who mine bibliographic information for scientific knowledge. A constructive approach for resolving name ambiguity is to use computer algorithms to identify author names. Some algorithm-based disambiguation methods have been developed by computer and data scientists. Among them, supervised machine learning has been stated to produce decent to very accurate disambiguation results. This paper presents a combination of principal component analysis (PCA) as a feature reduction and deep neural networks (DNNs), as a supervised algorithm for classifying AND problems. The raw data is grouped into four classes, i.e., synonyms, homonyms, homonyms-synonyms, and non-homonyms-synonyms classification. We have taken into account several hyperparameters tuning, such as learning rate, batch size, number of the neuron and hidden units, and analyzed their impact on the accuracy of results. To the best of our knowledge, there are no previous studies with such a scheme. The proposed DNNs are validated with other ML techniques such as Naïve Bayes, random forest (RF), and support vector machine (SVM) to produce a good classifier. By exploring the result in all data, our proposed DNNs classifier has an outperformed other ML technique, with accuracy, precision, recall, and F1-score, which is 99.98%, 97.98%, 97.86%, and 99.99%, respectively. In the future, this approach can be easily extended to any dataset and any bibliographic records provider.

Corresponding Author:

Siti Nurmaini
Intelligent Systems Research Group
Universitas Sriwijaya
Palembang 30137, Indonesia
Email: siti_nurmaini@unsri.c.id

1. INTRODUCTION

Scholarly digital libraries provide services allowing the discovery of millions of bibliographic citation records, facilitating literature research. These repositories contain author and co-authors name, work and publication venue, and titles of particular publications [1]. A digital library also offers useful research and data functionality to help funding institutions grant individuals [2]. However, digital libraries are not free of errors, such as disparate citation formats, scanning, and data conversion, ambiguous author names, and abbreviations of publication venues and titles [2]. Among the errors, the main attention is directed to ambiguous author names, due to the difficulties inherent in the publications of the research community. It is challenging to

recognize a publication's data owned by an individual, a fundamental issue since personal names are not adequately distinct. A large number of researchers are currently active in various disciplines [3].

Author name disambiguation (AND) is a crucial task in digital libraries because it can affect the accuracy and quality of digital libraries [4]. Typically, AND issues may take place in two different forms; synonym and homonym. The same author may appear under distinct names in the synonym issue, as they publish in various publications with varying presentations [5]. On the other hand, different authors may have shared or similar names referred to as homonym [6]. The synonym and homonym problems are the major challenges of recognizing the authorship of publications [1, 7]. These may be created by various issues such as misspellings, name changes due to marriage, religious or gender conversions, or abbreviations.

In recent years, several studies with various approaches have been conducted to solve AND challenges [6, 8, 9]. Shin *et al.* [6] propose a conventional method using graph framework for author disambiguation, which resolved by graph processes including vertex (or node) splitting and merging based on co-authorship. Yet, it is still inadequate in that minor conditions such as permanent changes to names or affiliations such as 'author profile changes' cannot be adequately addressed. Lin *et al.* [8] implement hierarchical agglomerative clustering for handling the AND issue with two attributes, i.e., the co-authors and title attributes. The co-author's name in the record are grouped into clusters, and a concept of ranking confidence to measure the confidence of different similarity measurements is created. Hussain and Asghar [9] use a graph structural clustering algorithm disassociating authors using a group detection algorithm and graph operations. Unfortunately, it cannot detect highly ambiguous author names in cases where one researcher has multiple research interests. Some limitations that can be explored in the future include self-citations, hidden concepts and email addresses of authors. On the other hand, Ferreira and Gonçalves [1] classify the publication authorship approach into two types, i.e. author grouping and author assignment. The author grouping approach clusters the authors based on the similarity of the publication data attribute [10, 11], while the author assignment approach directly assigns a publication to the author by building a model that represents the author [12, 13].

This paper highlights the author's assignment type to recognize publication authorship. In the type, there are two approaches to learning; classification and clustering. The advantage of the classification method is its efficacy when faced with many citation examples for each author. In contrast, the clustering method needs privileged information about the appropriate number of authors or the number of author classes and may take some time to determine their parameters [1]. Some researchers used the author assignment approach with classification [12, 14, 15]. Still, the results are not satisfying in F1-score and accuracy [12, 14]. The use of the artificial neural networks approach is already explored to recognize the authorship of the publication. However, the performance gets poor recall with a good result on accuracy [15]. To enhance the performance of conventional neural network algorithms, a DNNs with multiple layers is proposed in this research. DNNs have a strong ability to feature learning in many tasks and solve the publication authorship problem [4]. DNNs can build a general model that could disambiguate author name on a step-by-step basis when new publication records are integrated into the dataset. This paper also explores four combinations of types of publications data (multiclass classification) problems in the classification task, i.e., synonyms, homonyms, homonyms-synonyms, and non-homonyms-synonyms classification, which are the main problems of author identification [16]. For comparisons, Naïve Bayes, random forest, SVM are used for benchmarking the result of classifier performance.

2. RESEARCH METHOD

In this paper, the method for author assignment method is through assigning a reference to a specific author by building a model that represents the author using the classification technique [1]. A publication dataset will have four different cases; homonym, synonym, synonym-homonym, and non-synonym-homonym. A homonym is the cases when different persons share the same name, and synonym is the cases when the name of a particular author is given in several different ways [1]. The synonym-homonym case means the sample data has both a synonym and homonym case. Otherwise, the non-synonym-homonym case must not have a synonym or homonym case.

This paper proposed the author identification processing that consists of four stages; (i) data preparation, (ii) feature extraction, (iii) classification, and (iv) performance evaluation (see in Figure 1). The digital bibliographic & library project (DBLP) labeled dataset is implemented in this study. In the feature extraction, the new features are extracted from each attribute in a dataset. While in the classification, the process and learn those features to represent the specific authors. The comparison of two classifiers, SVM and DNN, has been explored in this study. In the last, the classifiers will be evaluated with five performance metrics (i.e., accuracy, sensitivity, specificity, precision, and F1-score) to validate the proposed model.

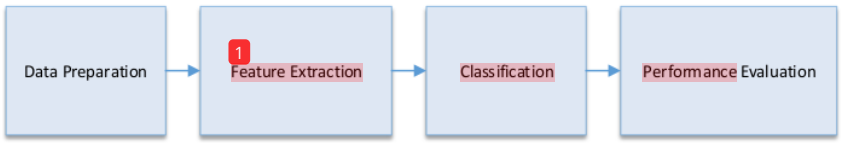

Figure 1. Author identification processing stage

2.1. Data preparation

In this paper, we implement the author name disambiguation labeled data generated by Dr. Giles research lab at the Pennsylvania State University [17, 18], and cleaned by Kim [19]. The cleaning process resulted in 5018 name instances with 480 distinct authors (author labels) and 456 distinct presented names where each author has 1 to 480 references. The dataset comprises the author's presented name, author label, authors name, venue, and title. The label of four AND problems are unavailable in the dataset. Therefore, we need categorizing these four cases with (1-4), for homonym, synonym, homonym-synonym, and non-homonym-synonym, respectively;

$$homonym = X \rightarrow Y\ 1 \mapsto m,\ m \geq 2 \quad (1)$$

$$synonym = Y \rightarrow X\ 1 \mapsto n,\ n \geq 2 \quad (2)$$

Where X is presented name, Y is author, m is the number of Y, and n is the number of X. For the homonym case, one X has the number of Y more than or equal to 2, whereas, for the synonym case, one Y has the number of X more than or equal to 2.

$$homonymsynonym = homonym \cap synonym \quad (3)$$

$$nonhomonymsynonym = (homonym \cup syonoiym)^C \quad (4)$$

2.2. Feature extraction

The feature extraction for author identification can be presented in Figure 2. Figure 2 shows the preprocessing phase for author identification, data normalization, feature extraction, features concatenation, and features reduction. The features that become the classifier input are extracted from dataset attributes. The dataset attributes consist of two types of attributes; categorical (presented name, author name, venue, and title) and numerical (year). Categorical attributes are processed into a one-hot numeric array, while numerical attributes are left as is. The first feature group is extracted from the presented name attribute. These features are extracted by creating a one-hot numeric array of labels encoding distinct presented names. The encoding label is the conversion of categorical data into numerical. This process produces a feature in the form of a dense binary array with an array length equal to the number of distinct presented names. The second feature group is extracted from the authors name attribute. For the authors name attribute, a label is encoded for all the distinct authors names created in the same way as the presented name. Then, only year attributes are specifically normalized with a min-max scaling in (5).

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (5)$$

where X is the input, X_{sc} is the scaled input, X_{min} is the minimum value, and X_{max} is the maximum value.

The third group is the venue attribute, which has the same process used on the presented name. Unlike in other groups, there are two main preprocessing stages for title attribute in the text attributes such as text normalization and feature extraction. In addition, for text normalization, lemmatization and lancaster stemmer are used, while for feature extraction, term frequency-inverse document frequency (TF-IDF), $tfidf$, is used in (6).

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \quad (6)$$

where $tf(t, d)$is the term frequency, the number of time that term $t$occurs in document d, which document in corpus, D.

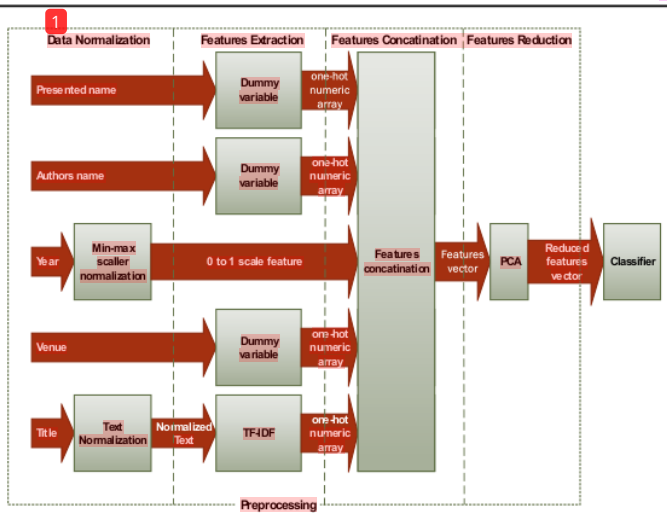

Figure 2. Author Identification preprocessing phase

In the feature concatenation, the extracted features are combined from the presented name, author label, authors name, year, venue, and title results in many features. Some author assignment studies have used the blocking method before conducting algorithmic disambiguation tasks for reducing computing time [20, 21]. Dissimilar to them, this paper reduces the dimensionality of features via principal component analysis (PCA). PCA uses an orthogonal transformation to change a set of observations of variables correlated to the value of variables that are not linearly correlated, called the principal component [22]. Han *et al.* propose PCA for AND problems because of its ability to remove the linear correlations and improve the generalization performance [23]. In this research, we fine-tune the number of features from 2 to 2500 features to find the best classifier performance.

2.3. Classification

The classifier gets input from extracted features in the previous process. The classifier learns the features of the training dataset to determine a reference to a specific author. In this paper, the proposed classifier was conducted using DNNs classifiers and Naïve Bayes, random forest, SVM as comparisons. The proposed classifier uses DNNs-based classifier. DNNs refer to neural networks with a large number of hidden layers. With deep architecture in Neural Networks, DNNs can represent higher complexity functions. This ability is possible by increasing the number of layers and neurons in the layer [24] (Figure 3).

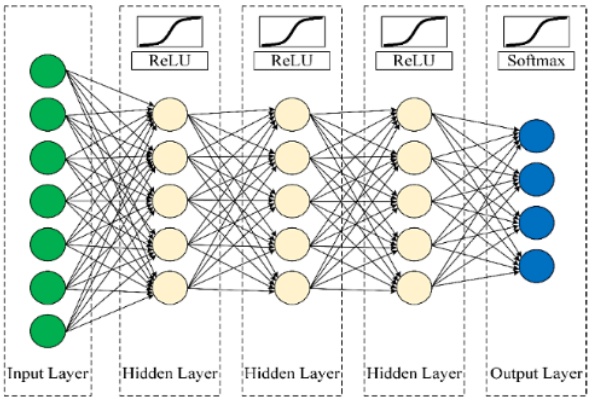

Figure 3. Proposed DNNs architecture

In this research, DNNs train with categorical cross-entropy loss function as in (7) and rectified linear unit (ReLU) as in (8) as activation functions. Three hidden layers with 50, 100, 150, and 200 neurons, without dropout, and 0.1 to 0.5 dropout values architecture are used in this experiment to get optimum classification performance (Table 1). The parameters that produce the best classification performance are selected as the neural network builder parameterser, which x is the input vector, y is the desired output, and $\hat{y}$ is the predicted output. We try 100 epochs, 0.01 learning rate and 64 batch size for all scenarios.

$$L(y,\hat{y})(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij}\cdot\log(\hat{y}_{ij}))\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij}\cdot\log(\hat{y}_{ij})) \quad (7)$$

$$f(x)(x)=\begin{cases}0, & x\le 0\\ x, & x>0\end{cases}\begin{cases}0, & x\le 0\\ x, & x>0\end{cases} \quad (8)$$

Table 1. DNNs architecture and tuning parameters

Layer	Number of Neurons	Activation Function
Input	PCA generated	-
Hidden layer 1	50, 100, 150, 200	ReLU
Hidden layer 2	50, 100, 150, 200	ReLU
..	…	…
Hidden layer 8	50, 100, 150, 200	ReLU
Output layer	266	Softmax

2.4. Evaluation

The dataset is divided by 80% of training data and remaining for the testing data. Before splitting, by considering the number of author references and the distribution of training and testing data, we removed authors who have less than five references from the dataset, so the number of data decreases from 5018 to 4419 name instances. The comparison of the raw dataset and the prepared dataset is presented in Table 2.

The number of rest author determines the number of classes used in the classification. Therefore, the number of classes set as 266 classes of authors. Table 3 presents the record number and portion of each four AND problem affected by data cleaning and splitting. Data cleaning gives a fair effect on the portion of four AND problems, while data splitting does not affect.

Table 2. Composition of raw dataset compared to the prepared dataset

	Raw dataset	Prepared dataset
Name instances	5018	4419
Distinct authors	480	266
Distinct presented names	456	303
Distinct venues	1004	923
Distinct co-author names	4653	3733
Year range	1959-2010	1959-2010
Synonym authors/row affected	46/1069	46/1120
Homonym presented names/row affected	62/787	15/328
Non-synonym-homonym row affected	2988	2861
Synonym-homonym row affected	174	110

Table 3. Composition of training and testing dataset

AND Problem	Raw dataset		Prepared dataset		Training dataset		Testing dataset	
	Record number	(%)	Record number	(%)	Record number	(%)	Record number	(%)
Synonym	1069	21.30%	1120	25.34%	900	25.46%	221	25.00%
Homonym	787	15.70%	328	7.42%	262	7.41%	66	7.46%
Synonym-Homonym	174	3.46%	110	2.50%	85	2.40%	25	2.83%
Non-Synonym-Homonym	2988	59.54%	2861	64.74%	2288	64.72%	572	64.70%
Total	5018	100.00%	4419	100.00%	3535	100.00%	884	100.00%

We applied few statistic methods to evaluate the performance of proposed methods, such as average accuracy as in (9), precision as in (10), recall as in (11), and F1-score as in (12) to evaluate our method performance [25].

$$Average\ Accuracy = \frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+tp_i+fp_i+tn_i}}{l} \quad (9)$$

$$Precision = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \quad (10)$$

$$Recall = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \quad (11)$$

$$F_1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (12)$$

Where tp is true positive, tn is true negative, fp is false positive, fn is false negative, and l is the number of classes. The performance evaluation method is carried out on four author name disambiguation issues; homonym, synonym, non-synonym-homonym, and synonym-homonym.

3. RESULTS AND DISCUSSION

The feature extraction process generates a number of features for each attribute. 7793 features were generated with details presented in Table 4. The number can increase the computational cost. Thus, it reduced by applying PCA become 1382 features. To obtain an optimum classification performance, various DNNs structures in the learning process were examined. The 224 structures were determined and validated before the selection of the best model. All classifiers were arranged in two processes i.e., training and testing. All processing time result of 224 DNNs structures is presented in Table 5.

Table 4. Number of features.

Attribute	Feature Extraction Technique	Number of features
Presented name	Dummy variable	303
List of author's names	Dummy variable	3733
Year	None	1
Venue	Dummy variable	923
Title	TF-IDF	2833

Table 5. Model accuracy (%) for 244 DNNs structures

Neuron	Learning Rate	Hidden Layer 1	2	3	4	5	6	7	8
50	1E-01	99,5535	99,3068	99,2855	99,2660	99,2694	99,2660	99,2694	99,2651
50	1E-02	99,9804	99,9770	99,9481	99,9107	99,7891	99,7227	99,6538	99,5109
50	1E-03	99,9813	99,9796	99,9762	99,9702	99,9592	99,9294	99,9098	99,8622
50	1E-04	99,7967	99,8537	99,9192	99,9337	99,8852	99,8529	99,8316	99,7270
50	1E-05	99,3927	99,3102	99,2694	99,2838	99,2626	99,2847	99,2643	99,2728
50	1E-06	99,2524	99,2609	99,2524	99,2600	99,2583	99,2685	99,2668	99,2677
50	1E-07	99,2541	99,2498	99,2532	99,2507	99,2498	99,2515	99,2481	99,2498
100	1E-01	99,7355	99,3043	99,2694	99,2626	99,2541	99,2660	99,2694	99,2694
100	1E-02	99,9813	99,9779	99,9541	99,8461	99,7117	99,5926	99,4191	99,3808
100	1E-03	99,9830	99,9821	99,9779	99,9779	99,9660	99,9634	99,9541	99,9566
100	1E-04	99,9515	99,9762	99,9753	99,9736	99,9575	99,9524	99,9226	99,8707
100	1E-05	99,4718	99,2872	99,2889	99,2813	99,3238	99,2804	99,3485	99,2906
100	1E-06	99,2541	99,2498	99,2532	99,2566	99,2558	99,2634	99,2626	99,2634
100	1E-07	99,2498	99,2515	99,2498	99,2490	99,2498	99,2507	99,2515	99,2524
150	1E-01	99,7891	99,2958	99,2728	99,2694	99,2677	99,2694	99,2566	99,2558
150	1E-02	99,9770	99,9779	99,9439	99,7457	99,6079	99,4718	99,4335	99,3349
150	1E-03	99,9838	99,9821	99,9804	99,9762	99,9762	99,9660	99,9600	99,9396
150	1E-04	99,9753	99,9830	99,9804	99,9787	99,9770	99,9583	99,9405	99,9149
150	1E-05	99,5041	99,3230	99,3408	99,3646	99,3749	99,3689	99,3910	99,3570
150	1E-06	99,2600	99,2617	99,2566	99,2575	99,2575	99,2626	99,2634	99,2796
150	1E-07	99,2515	99,2507	99,2541	99,2566	99,2490	99,2507	99,2524	99,2498
200	1E-01	99,8180	99,2728	99,2694	99,2660	99,2660	99,2694	99,2694	99,2609
200	1E-02	99,9753	99,9779	99,9209	99,6036	99,5866	99,3706	99,3400	99,2762
200	1E-03	**99,9838**	99,9830	99,9804	99,9753	99,9745	99,9711	99,9643	99,9651
200	1E-04	99,9813	99,9813	99,9804	99,9813	99,9787	99,9719	99,9515	99,9132
200	1E-05	99,5620	99,3629	99,3689	99,3961	99,4148	99,4318	99,4888	99,4540
200	1E-06	99,2583	99,2626	99,2660	99,2753	99,2677	99,2719	99,2694	99,2677
200	1E-07	99,2524	99,2532	99,2583	99,2490	99,2498	99,2524	99,2507	99,2515

The best results of DNNs are a model with one hidden layer DNNs structures and 200 neurons for each layer with 0.5 dropout value. From the classification process, the DNNs model structure is selected based on the highest accuracy. Both training dan testing processes, but it more important in the testing process. The highest average accuracy for all data, about 99.99 % in training and 99.98% in testing. The same results for all AND problems, the accuracy value about 99%. However, the recall value for homonym-synonym is under 70% (see Table 6 and Figure 4). The number of sample data with a homonym-synonym condition is less than other conditions around 110 data from a total of 4419 data or only about 2.5% of the total data. The imbalance data can decrease the ML performance.

Table 6. Proposed DNNs classification performances

AND Problem	Accuracy	Precision	Recall	F1 Score
All	99,9951	99,3289	99,3293	99,3123
	99,9838	97,9887	97,8641	99,9919
Homonym	99,2730	87,9282	88,9600	88,3163
	98,0303	79,8333	77,4524	98,9198
Synonym	100,0000	100,0000	100,0000	100,0000
	99,9380	93,3712	95,3463	99,9686
Homonym-Synonym	99,5848	84,1176	88,2353	85,6209
	98,1538	69,2308	69,2308	99,0769
Non-Homonym-Synonym	100,0000	100,0000	100,0000	100,0000
	100,0000	100,0000	100,0000	100,0000

Note : ■Training and ■Testing

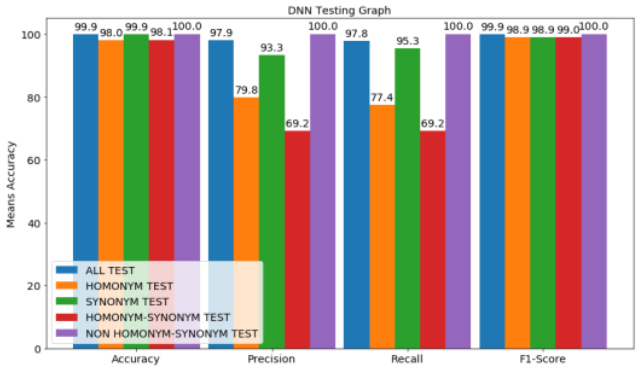

Figure 4. The DNNs testing

For validating the proposed DNNs approach, three techniques like Naïve Bayes, Random Forest, and support vector machine (SVM) are compared in terms of accuracy, precision, recall, and F1-Score. Table 7 shows the classifications accuracy performances of the DNNs classifier compared with other ML techniques, such as 99.8% for all types of data, 98.0% for homonym, 99.8% for synonym, 98.1% for homonym-synonym, and 100% for non-homonym-synonym. In all metrics of precision, recall, and F1-score, DNNs outperforms other ML technique. Actually, SVMs deliver a unique solution in the classification task, since the optimality problem is convex. This is an advantage compared to ANNs, which have multiple solutions associated with local minima. However, DNNs used a deep structure of hidden layers. It can overcome the drawback. Therefore, all performances are improved by about 1% over the SVM.

Our proposed method with DNNs classifier in author identification on bibliographic data containing homonym and synonym data produce a good performance. By exploring the result, our method with DNNs classifier has a better performance than other ML techniques. Comparing to the same research with the same dataset [19], our proposed DNNs method has a better result in a recall, i.e., 97.9% compared to Naïve Bayes, Random Forest classifiers, and support vector machine. As shown in Table 8, the non-synonym-homonym category works perfectly in all performance measurements, which has 100%. It is not surprising because of the category of non-synonym-homonym is not the main issue for author identification. We explained above, synonym and homonym are critical problems. The synonym-homonym category harder problem to solve. These issues must be explored per each category for the author identification by its characteristics.

Table 7. Comparison of DNNs performances (%) for all data with other ML techniques

Classifier	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	99,8716	67,9285	73,1047	69,2674
Random Forest	99,8044	56,8699	58,2380	55,2470
SVM	99,9753	94,5677	95,2300	94,5997
DNN	99,9838	97,9887	97,8641	99,9919

Table 8. Comparison of DNNs performances (%) for each AND problems with other ML techniques

AND Problem	Classifier	Accuracy	Precision	Recall	F1 Score
Homonym	Naïve Bayes	97,2028	37,9487	43,4615	39,6410
	Random Forest	97,7553	42,4691	48,7302	44,0116
	SVM	98,3333	81,8333	80,8333	81,2778
	DNN	98,0303	79,8333	77,4524	98,9198
Synonym	Naïve Bayes	99,5059	70,0362	79,3536	73,3338
	Random Forest	99,4357	47,8859	55,9347	50,3384
	SVM	99,8918	94,2460	95,3953	94,1228
	DNN	99,9380	93,3712	95,3463	99,9686
Homonym-Synonym	Naïve Bayes	96,5714	55,3571	57,1429	56,1224
	Random Forest	94,9091	23,6364	31,8182	25,7576
	SVM	98,0000	75,0000	72,9167	73,8095
	DNN	98,1538	69,2308	69,2308	99,0769
Non-Homonym-Synonym	Naïve Bayes	99,8357	65,4443	70,3323	66,9863
	Random Forest	99,7518	48,6593	51,2205	48,3987
	SVM	99,9827	94,0594	95,0495	94,3894
	DNN	100,0000	100,0000	100,0000	100,0000

4. CONCLUSION

From the experiment results, it can be concluded that the method produces good results for all problems with an average accuracy of 99.98%. The method solves the synonym problem better than homonym; besides, the performance regarding the combined synonym-homonym problem is still less than satisfactory. The complexity of recognizing and assigning publications to the respective authors is not a simple task. Some techniques have been proposed for solving author name disambiguation, specifically in synonym and homonym problems. Four machine learning algorithms have been compared to obtain precise performance. The results revealed that NNs with one layer significantly outperformed other machine learning techniques with an average accuracy of 99.98%. Setting up a NNs algorithm is much more tedious than using an off-the-shelf classifier like SVM. For large-data analytical methods associated with machine learning algorithms, deeper NNs using DNNs are promising algorithms in various fields of application, including author name disambiguation. DNNs employ various deep learning algorithms based on network structure, activation function, and model parameters, with their output depending on the data representation format. From the experimental results of this research, however, both DNNs and SVM obtain higher performance in synonym problems than in homonym problems. With the proposed DNNs, the performances in synonyms result in values for accuracy, precision, recall, and F1-score of 99.94%, 93.37%, 95.35%, and 99.97%, respectively. In future work, in the big data era for the modern digital library, DNNs, the proposed method is typically very helpful for working with larger datasets. Besides, for homonym and homonym-synonym, an appropriate method should be implemented in other datasets and increased performance. The use of feature engineering based on semantic approached for title attribute could improve the performance of all cases.

ACKNOWLEDGEMENTS

We thank the Ministry of Research, Technology, and Higher Education, Republic of Indonesia (Kemenristekdikti RI), for funding the research on "Penelitian Disertasi Doktor" Research Grant with contract number 211/SP2H/LT/DRPM/IV/2019.

REFERENCES

[1] A. A. Ferreira and M. A. Gonçalves, "A Brief Survey of Automatic Methods for Author Name Disambiguation," *ACM SIGMOD Record*, vol. 41, no. 2, pp. 15-26, 2012.

[2] I. Hussain and S. Asghar, "A survey of author name disambiguation techniques: 2010-2016," *Knowl. Eng. Rev The Knowledge Engineering Review,* vol. 00:0, pp. 1-24, 2017.

[3] S. Milojević, "Accuracy of simple, initials-based methods for author name disambiguation," *J. Informetr.*, vol. 7, no. 4, pp. 767-773, 2013.
[4] H. N. Tran, T. Huynh, and T. Do, "Author name disambiguation by using deep neural network," in *Asian Conference on Intelligent Information and Database Systems*, 2014, pp. 123-132.
[5] F. Momeni and P. Mayr, "Using Co-authorship Networks for Author Name Disambiguation," *Proc. 16th ACM/IEEE-CS Jt. Conf. Digit. Libr. - JCDL '16*, 2016, pp. 261-262.
[6] D. Shin, T. Kim, J. Choi, and J. Kim, "Author name disambiguation using a graph model with node splitting and merging based on bibliographic information," *Scientometrics*, vol. 100, no. 1, pp. 15-50, 2014.
[7] N. R. Smalheiser and V. I. Torvik, "Author name disambiguation," *Annu. Rev. Inf. Sci. Technol.*, vol. 43, no. 1, pp. 1-43, 2009.
[8] X. Lin, J. Zhu, Y. Tang, F. Yang, B. Peng, and W. Li, "A novel approach for author name disambiguation using ranking confidence," *International Conference on Database Systems for Advanced Applications*, 2017, pp. 169-182.
[9] I. Hussain and S. Asghar, "LUCID: Author name disambiguation using graph Structural Clustering," in *2017 Intelligent Systems Conference, IntelliSys 2017*, 2018, vol. 2018-Janua, pp. 406-413.
[10] R. G. Cota, A. A. Ferreira, C. Nascimento, M. A. Gonçalves, and A. H. F. Laender, "An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations," *J. Am. Soc. Inf. Sci. Technol.*, vol. 61, no. 9, pp. 1853-1870, 2010.
[11] A. Culotta, P. Kanani, R. Hall, M. Wick, and A. McCallum, "Author disambiguation using error-driven machine learning with a ranking loss function," in *Sixth International Workshop on Information Integration on the Web (IIWeb-07), Vancouver, Canada*, 2007.
[12] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. F. Laender, "Effective self-training author name disambiguation in scholarly digital libraries," in *Proceedings of the 10th annual joint conference on Digital libraries*, 2010, pp. 39-48.
[13] J. Tang, A. C. M. Fong, B. Wang, and J. Zhang, "A unified probabilistic framework for name disambiguation in digital library," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 975-987, 2011.
[14] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, "Two supervised learning approaches for name disambiguation in author citations," in *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.*, 2004, pp. 296-305.
[15] S. F. Schifano, T. Sgarbanti, and L. Tomassetti, "Authorship recognition and disambiguation of scientific papers using a neural networks approach," in *Proceedings of Science*, 2018.
[16] S. S. Khan and M. G. Madden, "A survey of author name disambiguation techniques: 2010–2016," *The Knowledge Engineering Review,* vol. 32, 2017.
[17] H. Han, H. Zha, and C. L. Giles, "Name disambiguation spectral in author citations using a k-way clustering method," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, JCDL, Denver, CO, USA*, 2005, pp. 7-11.
[18] 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*, 2005, pp. 1065–1069.
[19] J. Kim and J. Kim, "The impact of imbalanced training data on machine learning for author name disambiguation," *Scientometrics*, vol. 117, no. 1, pp. 511-526, 2018.
[20] P. Mitra, J. Kang, D. Lee, and B. On, "Comparative study of name disambiguation problem using a scalable blocking-based framework," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, 2005, pp. 344-353.
[21] T. Backes, "The impact of name-matching and blocking on author disambiguation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 803-812.
[22] Q. Qin, J. Li, L. Zhang, Y. Yue, and C. Liu, "Combining Low-dimensional Wavelet Features and Support Vector Machine for Arrhythmia Beat Classification," *Sci. Rep.*, vol. 7, no. 1, pp. 1-12, 2017.
[23] D. Han, S. Liu, Y. Hu, B. Wang, and Y. Sun, "ELM-based name disambiguation in bibliography," *World Wide Web*, vol. 18, no. 2, pp. 253-263, 2015.
[24] W. Liu *et al.*, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11-26, 2017.
[25] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427-437, 2009.

Author identification in bibliographic data using deep neural.pdf

ORIGINALITY REPORT

85%	85%	8%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	journal.uad.ac.id Internet Source	84%
2	Submitted to Vietnam Maritime University Student Paper	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		