Advances in Intelligent Systems Research, volume 172
Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)

Author Matching Using String Similarities and Deep Neural Networks

Zaqqi YAMANI[1], Siti NURMAINI[2], FIRDAUS[3], M Naufal R[4], Winda Kurnia SARI[5]

[1,2,3,4,5]*Faculty of Computer Science, Universitas Sriwijaya, Indonesia*

Abstract
Author Name Disambiguation (AND) is one of the problems in the process of classification of publication data, where the introduction of the characteristics of each author is difficult to recognize because of the frequent changes in patterns in writing author names. In the world of publications, author classification is needed to classify authors with types and fields of science based on published papers, therefore in this paper the author will classify with the aim of providing definite values whether the author is with the paper title a and author a with the paper title b is the same person or not. The method used by the author is one branch of Deep Learning, namely the Deep Neural Network DNN). DNN is used because it has high data processing acceleration capabilities with the use of GPU technology. With this method, AND data classified by the author produces a high level of accuracy with a value of 98%.

Keywords: *Author Name Disambiguation (AND), author matching, Deep Neural Network*

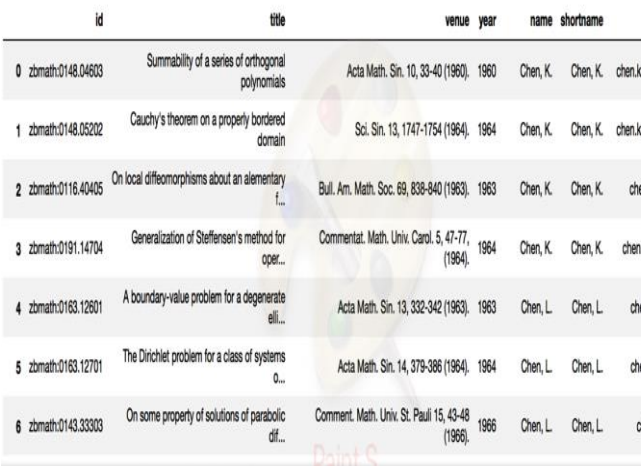
Introduction

One problem in the world of publication research is the introduction of the characteristics of the author (author) in writing names or often known as Author Name Disambiguation (AND). Often the occurrence of differences in name writing results in the perception that the same person is often considered a different person. And in the past few years, the number of publication data uploaded and disseminated online is increasing, making the problem of disambiguate in the name of the author getting bigger. In particular, the name ambiguity arises when the collection of citation notes containing the author's name (author names) is ambiguous and may appear in two different forms, in the first form the same author's name can appear under a different name called synonym, and in the second form Different author names may have similar names called homonyms [1].
In this paper, the author tries to identify the AND dataset. Where in the dataset consists of a collection of publication titles and names of authors who have been notated with a unique author identifier (id author)[2]. before finding a method for classifying AND data, the author first examines several methods that have been tried by several previous authors. The classifications of several methods that have been done before include "supervised AND techniques", "unsupervised AND techniques", "semi-supervised AND techniques", "Graph based AND techniques", "Heuristic based AND techniques" by producing several different analysis results -different [3].
After reviewing the techniques, the writer tries to classify AND data by using one method from Machine Learning, namely Deep Learning or more specifically by using Deep Neural Network (DNN)[4]. Recent researches, DNN able to show a strong ability to do many tasks, especially the classification process[1].With this method, the author tries to decipher the data through the scad-zbMATH dataset [2], to produce a high level of accuracy and precision of data from the dataset. Before classifying with DNN method, the dataset is preprocessed by using stop words removal method, combination between lines, calculation of distance from each string, and identification of id author based on the results of the combination.
After the combination is carried out, the process of calculating distance between strings is carried out and the data comparison process between the author id is made to 0 (author conditions are the same) and 1 (author conditions are not the same). Then, using the DNN method on the python programming language, the results obtained at the level of accuracy are 98% and the data precision is 99%.

Materials and Methods

Dataset

The data set used in this study is a dataset of the results of grouping publications, and the dataset is named SCAD-zbMATH[2]. The data used amounted to 575 lines of data with 7 data classifications. The classification consists of 'id', 'title', 'venue', 'year', 'name', 'short name' and 'id2'. Then ‘id2’ will be referred to as id author. The dataset is not included in the process of presetting. Can be seen in Fig. 1.



	id	title	venue	year	name	shortname	
0	zbmath:0148.04603	Summability of a series of orthogonal polynomials	Acta Math. Sin. 10, 33-40 (1960).	1960	Chen, K.	Chen, K.	chen.k
1	zbmath:0148.05202	Cauchy's theorem on a properly bordered domain	Sci. Sin. 13, 1747-1754 (1964).	1964	Chen, K.	Chen, K.	chen.k
2	zbmath:0116.40405	On local diffeomorphisms about an elementary f...	Bull. Am. Math. Soc. 69, 838-840 (1963).	1963	Chen, K.	Chen, K.	ch
3	zbmath:0191.14704	Generalization of Steffensen's method for oper...	Commentat. Math. Univ. Carol. 5, 47-77, (1964).	1964	Chen, K.	Chen, K.	chen
4	zbmath:0163.12601	A boundary-value problem for a degenerate elli...	Acta Math. Sin. 13, 332-342 (1963).	1963	Chen, L.	Chen, L.	ch
5	zbmath:0163.12701	The Dirichlet problem for a class of systems o...	Acta Math. Sin. 14, 379-386 (1964).	1964	Chen, L.	Chen, L.	ch
6	zbmath:0143.33303	On some property of solutions of parabolic dif...	Comment. Math. Univ. St. Pauli 15, 43-48 (1966).	1966	Chen, L.	Chen, L.	c

Figure 1 Dataset SCAD-zbMATH

PREPROCESSING

Stopword

Stop words are common words that usually appear in large numbers and are considered to have no meaning. Stop words are generally used in task information retrieval, including by Google. Examples of stop words for English include "of", "the".
This is done in the preprocessing process so that the string data does not become ambiguous, often in the title of the publication / title.

Pairwise Combinations

A combination is to combine several objects from a group regardless of sequence. In combination, the sequence is not considered[5]. The combination of data is also done in the preprocessing process, with the aim of juxtaposing or reconciling all rows of data and each column for the next process of comparison of each successful string combined. The combination of data sets totaling 575 rows and 7 columns will produce 165,025 rows and 14 columns, and next by converting all strings to numbers to find the similarity of each string with the process of calculating distances (sequence matcher). The process is carried out by comparing columns 1, 2, 3 and 4 and so on except the year column, carried out by reducing from year to year b and in absolute, then in the author id column is done by comparing the id, if the same is 0 and if the difference is worth 1 for further processing as binary.
To produce or justify the amount obtained from the combination results, the author has adjusted to the combination formula in mathematics as follows:

$$\frac{n!}{r!(n-r)!} = \binom{n}{r} \quad (1)$$

Feature Extraction (Similarity Distance)

Similarity is a process in the extraction feature by searching for similarity values or similarity of data. In the machine learning method, for the case of finding similarities from string data, the similarity technique becomes the most dominant technique used. This technique is able to build information through values or in other words translate a data string into data that has values [6][7].
In its application, this technique is carried out after the formation of paired data. In addition to author matching data, this technique is also used in the process of image-based emotion recognition. This technique is done by comparing two sets of data to see which data are the same and which data are different.
In many studies on AND, the similarity technique that is widely used is to use the coefficients of jaccard, jaro, euclidean and levenshtein. The most common jaccard coefficient is used to measure the similarity between two data sets. The results that appear are in the range of 0% to 100%. The higher the percentage, the more similar the population of the two data [8][4][9][10][11].
The jaccard coefficient is defined by a mathematical formula to produce measurements of the distance between strings.

$$\boldsymbol{jaccard}\,(\boldsymbol{a},\boldsymbol{b}) = \frac{|\boldsymbol{a}\cap\boldsymbol{b}|}{|\boldsymbol{a}\cup\boldsymbol{b}|} = \frac{|\boldsymbol{a}\cap\boldsymbol{b}|}{|\boldsymbol{a}|+|\boldsymbol{b}|-|\boldsymbol{a}\cap\boldsymbol{b}|} \quad (2)$$

The jaccard coefficient has a weakness where this coefficient does not pay attention to the term frequency (how many times a term is contained in a document). It should be noted that the terms that rarely appear in a collection are very valuable in terms of information, but the Jaccard coefficient does not take this into account. So we need another way to normalize it. But for AND data processing, especially at author matching points, the jaccard coefficient calculation technique is considered better for use [8].

Deep Neural Network

Deep Neural Network is one branch of learning machines with a certain level of complexity, a neural network with more than two layers[12]. Deep neural networks use sophisticated mathematical modeling to process data in complex ways[13]. Standard neural networks consist of many processors called neurons, each producing a real value activation sequence. The input of neurons is activated through sensors that observe the environment, other neurons are activated through a weighted connection from previously active neurons [14].
The development of Deep Learning resulted in the ability to overcome MLP's shortcomings in dealing with a complex power structure, where the use of functions can make it easier for MLP to understand the process of changing inputs. The theorem of MLP is as follows:

$$y = \varphi(b\sum_{i=1}^{n} wixi + b) = \varphi(w^T x + b) \quad (3)$$

VALIDATION

Dataset

Confusion matrix is a method that is usually used to calculate accuracy in the concept of data mining or Decision Support Systems. In measuring performance using confusion matrix, there are 4 (four) terms as a representation of the results of the classification process [7]. The four terms are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). True Negative Value (TN) is the number of negative data detected correctly, while False Positive (FP) is negative data but detected as positive data. Meanwhile, True Positive (TP) is positive data that is detected correctly. False Negative (FN) is the opposite of True Positive, so the data is positive,but it is detected as negative data.
Precision is data taken based on information that is lacking [15]. In binary classification, precision can be made equal to positive predictive values. The following are precision rules.

Precision = (TP / (TP + FP)) * 100%

Recall is the deletion data that was successfully retrieved from data relevant to the query. In binary classification, recall is known as sensitivity. The emergence of relevant data taken is agreeing with the query can be seen by recall. The following is the role of recall.

Recall = (TP / (TP + FN)) * 100%

Accuracy is the percentage of the total data identified and assessed. Following are the rules for accuracy.
Accuracy = (TP + TN) / (TP + TN + FP + FN)) * 100%

Table 1. Confusion Matrix

		TRUE VALUES	
PREDICTION		TRUE	FALSE
	TRUE	TP Correction result	FP Unexpected result
	FALSE	FN Missing Result	TN Correct absence of result

RESEARCH METHODOLOGY

Data Preprocessing

The dataset used by the author is data with the number of 575 lines of string with 7 columns of data labels. Preparation begins with stop wording by using NLTK in python to remove words and punctuation in the title paper which will cause string ambiguity. Next, the upper case process is carried out on all strings so that all strings are small letters, the purpose is to facilitate recognition and search for similarities between strings.
After that, the next process is to combine the data with 2 parameters so that the number of rows of data generated becomes 165,025 lines with 14 column labels, for example there will be table id and id again and so on.
The next process is calculating the distance between strings with sequence matcher in python and generating calculation results in the form of numbers that show similarities between strings. Except for data year, distance calculation is not done, but by subtracting the year the result of the combination is then absolute, and data id2 (id author) is carried out by the comparative process and the value of 0 if equal and 1 if not equal the result of calculating the distance makes the data become 7 columns again. The preprocessing process produces author data equal to (0) a number of 6229 and author data is not the same (1) a number of 158796 and pre-prospering is depicted in fig. 2 while preprocessing results are described in Fig. 3.

	0	1	2	3	4	5	6
0	0.823529	0.336842	0.625000	4	1.000000	1.000000	0
1	0.764706	0.152381	0.657534	3	1.000000	1.000000	1
2	0.705882	0.336000	0.550000	4	1.000000	1.000000	1
3	0.705882	0.259259	0.882353	3	0.875000	0.875000	1
4	0.647059	0.285714	0.823529	4	0.875000	0.875000	1
5	0.764706	0.369231	0.592593	6	0.875000	0.875000	1
6	0.705882	0.327869	0.636364	7	0.875000	0.875000	1
7	0.705882	0.179775	0.666667	10	0.875000	0.875000	1

Figure 2 Result of PreProcessing and Similarity

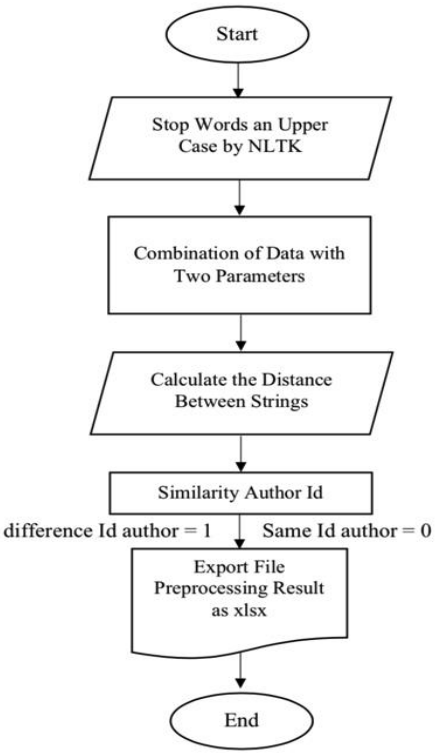


Figure 3 Preprocessing Flowchart

DNN Architecture

DNN consists of several layers where first there are input layers, hidden layers and output layers. In its implementation, the hidden layer used should be above 2 layers to produce good accuracy data. In this research, in the classification process of Author Name Disambiguation (AND) data, the author uses the number of hidden layers as in Fig. 4.

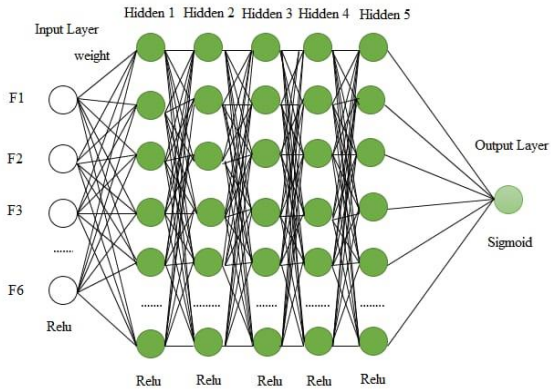


Figure 4. DNN Architecture

From Fig. 4, it can be understood that in the DNN architecture there are 6 features with 5 hidden layers using input layer 100 in the first hidden layer and 200 in the next hidden layer. In this architecture, the author uses the Relu Activation Function with the sigmoid Output Layer. Activation function is used to speed up the convergence process and Relu has a role to change the value to be non-linear and has a value of 0 and 1. The sigmoid function is used because the label used as the target in this study is binary, so sigmoid is chosen to be the Output Layer. In this study, datasets are split to be used in the training and testing process with a scale of 80: 20 with epoch 30 so as to produce an accuracy rate of 98%, 99% precision, 98% sensitivity, and 98% specificity with a 99% f-score. system architecture that is run in this study can be seen in the Fig 5.

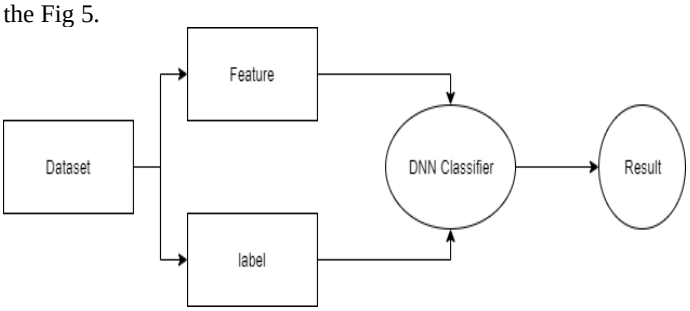


Figure 5. System Architecture

RESULTS AND DISCUSSION

Results

The classification process of Author Name Ambiguities (AND) using Deep Neural Network (DNN) gives high results on the level of accuracy, precision, specificity and sensitivity of the training and testing process where the values are as follows:

Table 2. Result of Training and Testing

	TRAINING	TESTING
- **ACCURACY**	98%	98%
- **PRESITION**	99%	99%
- **SENSITIVITY**	98%	98%
- **SPECIVICITY**	98%	99%

And the results of training and testing from processing these datasets can be seen in the graph in Fig. 4.

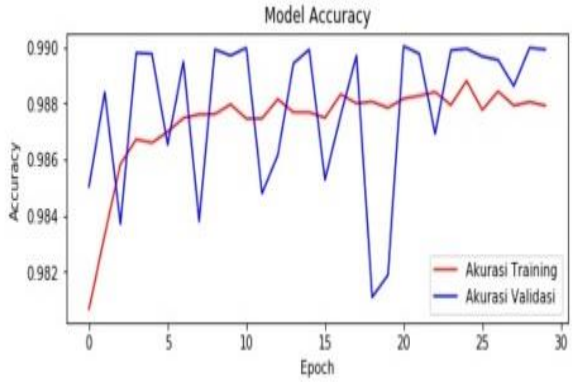


Figure 6 Training and Testing Accuration Curve

And the results of the loss curve with the results of 0.01 are described in the following Fig. 5.

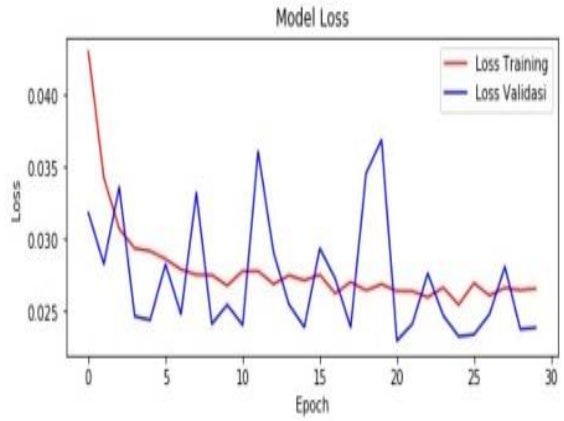


Figure 7 Training and Testing Loss Curve

Next is the value on the ROC curve showing a value of 1.0 and can be seen in Fig. 6.

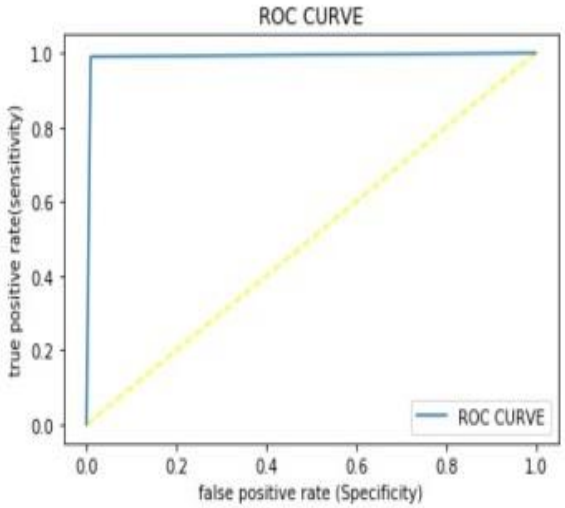


Figure 8 ROC Curve

Then the results of the confusion matrix from calculations carried out on a scale of 80: 20 from the amount of data with epoch 30 and in the python programming language with Jupiter notebook media can be seen in table 3.

Table 3. Result of Training and Testing of Confusion Matrix

TRUE VALUES

PREDICTION		TRUE	FALSE
		Training	Training
	TRUE	125689	54
	FALSE	1323	4954

TRUE VALUES

		TRUE	FALSE
		Testing	Testing
	TRUE	31462	11
	FALSE	322	1210

From the results of this research, it can be seen that DNN is able to produce a high level of accuracy in the classification process, and is also able to produce high scores on precision, and recall. this proves that predictions with results that are close to accurate in the author matching process can be done using neural network classification.

Discussion

The results of the AND classification with this DNN model can produce the best accuracy. However, in the Author Names Disambiguation (AND) dataset, this requires several very detailed steps in the preparation. This can determine the performance of the DNN model that is built and can affect the results of the model whether it will show maximum results or not.

SUMMARY

The data classification process in the Author Name Disambiguation (AND) dataset requires several detailed steps in the preprocessing process as an attempt to form data similarities that have string values. With the correct preprocessing results, the DNN model will present the best results in the classification model. With the correct preprocessing, the results of the DNN model still give the best results even though the data used is imbalance, and there is no need to process the data balancing first. In the data I use, although after preprocessing the data remains imbalance, the results of the DNN model still present data with 98% accuracy, 99% precision, 98% sensitivity, and 98% specificity with a 99% f1-score.

REFERENCES

[1] 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.

[2] M. C. Müller, F. Reitz, and N. Roy, "Data sets for author name disambiguation: an empirical analysis and a new resource," *Scientometrics*, vol. 111, no. 3, pp. 1467–1500, 2017.

[3] I. Hussain and S. Asghar, "A survey of author name disambiguation techniques: 2010–2016," *Knowl. Eng. Rev.*, vol. 32, p. e22, 2017.

[4] H. N. Tran, T. Huynh, and T. Do, "Author name disambiguation by using deep neural network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8397 LNAI, no. PART 1, pp. 123–132, 2014.

[5] D. Heckerman and D. M. Chickering, "Learning Bayesian Networks : The Combination of Knowledge and Statistical Data," vol. 243, pp. 197–243, 1995.

[6] F. Mozafari and H. Tahayori, "Emotion Detection by Using Similarity Techniques," *2019 7th Iran. Jt. Congr. Fuzzy Intell. Syst. CFIS 2019*, pp. 1–5, 2019.

[7] J. Lu, C. Lin, J. Wang, and C. Li, "Tutorial Proposal : Synergy of Database Techniques and Machine Learning Models for String Similarity Search and Join."

[8] A. A. Ferreira, M. A. Gonçalves, and A. H. F. Laender, "A brief survey of automatic methods for author name disambiguation," *ACM SIGMOD Rec.*, vol. 41, no. 2, p. 15, 2012.

[9] M. Song, E. H. J. Kim, and H. J. Kim, "Exploring author name disambiguation on PubMed-scale," *J. Informetr.*, vol. 9, no. 4, pp. 924–941, 2015.

[10] Q. Shen, T. Wu, H. Yang, Y. Wu, H. Qu, and W. Cui, "NameClarifier: A Visual Analytics System for Author Name Disambiguation," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 141–150, 2017.

[11] Y. Zhu and Q. Li, "Enhancing object distinction utilizing probabilistic topic model," *Proc. - 2013 Int. Conf. Cloud Comput. Big Data, CLOUDCOM-ASIA 2013*, pp. 177–182, 2013.

[12] S. Nurmaini, R. U. P, M. N. R, and A. Gani, "Cardiac Arrhythmias Classification Using Deep Neural Networks and Principle Component Analysis Algorithm," vol. 10, no. 2, 2018.

[13] J. Zhao, P. Wang, and K. Huang, "A semi-supervised approach for author disambiguation in KDD CUP 2013," in *Proceedings of the 2013 KDD Cup 2013 Workshop on - KDD Cup '13*, 2013.

[14] J. Schmidhuber, "Deep learning in neural networks : An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[15] Y. Cho and L. K. Saul, "Kernel Methods for Deep Learning," pp. 1–9.