

Article

Not peer-reviewed version

Name Disambiguation Scheme Based on Heterogeneous Academic Sites

[Dojin Choi](#) , Junhyeok Jang , [Sangho Song](#) , [Hyeonbyeong Lee](#) , HyeonbyeongJongtae Lim , [Kyoungsoo Bok](#) , [Jaesoo Yoo](#) *

Posted Date: 7 November 2023

doi: 10.20944/preprints202311.0399.v1

Keywords: name disambiguation; author name disambiguation; deep-learning; multiclass classification; HAC



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Name Disambiguation Scheme Based on Heterogeneous Academic Sites

Dojin Choi ¹, Junhyeok Jang ², Sangho Song ², Hyeonbyeong Lee ², Jongtae Lim ², Kyoungsoo Bok ³ and Jaesoo Yoo ^{2*}

¹ Department of Computer Engineering, Changwon National University, 20, Changwondaehak-ro, Uichang-gu, Changwon-si, Gyeongsangnam-do, Korea; dojinchoi@changwon.ac.kr(D.C.);

² Department of Information and Communication Engineering, Chungbuk National University, Chung-dae-ro 1, Seowon-Gu, Cheongju 28644, Chungbuk, Korea; tkalfm8@naver.com(J.J.); ssh@cbnu.ac.kr(S.S.); lhb@cbnu.ac.kr(H.L.); jtlim@cbnu.ac.kr (J.L.); yjs@cbnu.ac.kr (J.Y.)

³ Department of Artificial Intelligence Convergence, Wonkwang University, Iksandae 460, Iksan 54538, Jeonbuk, Korea; ksbok@wku.ac.kr (K.B.)

* Correspondence: yjs@cbnu.ac.kr; Tel.: +82-43-261-3230

Abstract: Academic researchers publish their work in various formats, such as papers, patents, and research reports, on different academic sites. When searching for a particular researcher's work, it can be challenging to pinpoint the right individual, especially when there are multiple researchers with the same name. In order to handle this issue, we propose a name disambiguation scheme of researchers with the same name based on heterogeneous academic sites. The proposed scheme collects and integrates research results from these varied academic sites, focusing on attributes crucial for disambiguation. It then employs clustering techniques to identify individuals who share the same name. Additionally, the proposed rule-based algorithm name disambiguation method and the existing deep learning-based identification method. This approach allows for the selection of the most accurate disambiguation scheme, taking into account the metadata available in the academic sites, using a multiclass classification approach. To demonstrate the effectiveness of the proposed method, we conduct various performance evaluations, measuring accuracy, recall, and the F1-measure, highlighting the scheme's superior performance in the name disambiguation.

Keywords: name disambiguation; author name disambiguation; deep-learning; multiclass classification; HAC

1. Introduction

Generally, users enter specific keywords on academic search sites to search through scholarly databases. These sites provide scholarly data, such as articles and reports, that match these keywords as search results. These search results include articles that provide information about the authors and contents of the articles. Since academic search sites hold several research records, individuals with the same name, even in the same research field, are commonly encountered. That is, researchers researching in the same or different fields usually have identical names. Most academic search sites offer a feature to research within the author name search results. This feature is provided to refine and search again for the researcher or keyword the user actually wants to find. However, this feature poses a challenge as users are required to determine for themselves from the search results if the name belongs to a different researcher. Additionally, even if a specific academic search site distinguishes between individuals with the same name effectively, determining those based on results provided by different academic search sites is very challenging. This is inconvenient and is a basis for incorrect judgments made by individuals searching for academic information. Therefore, in order to utilize various academic search sites, a function that can identify individuals with the same name across different academic search sites is necessary [1,2].

Distinguishing and identifying individuals with the same name play a significant role in enhancing search accuracy. When users search the name of a researcher, the search results provide

all the research outputs of every researcher with the same name. By using the filtering feature provided by academic search sites and entering additional information about the desired content, users can increase the accuracy of the search results. Studies on name disambiguation have been conducted using schemes that use the metadata of academic search sites to discern authors with identical names [1–16]. Various studies have been conducted to address name disambiguation on academic sites. In [5], a scheme was proposed to discern individuals with the same name using the metadata of academic search sites and determining name matches based on the similarity of attributes between two different papers. Furthermore, studies have been conducted to establish rules for calculating the similarity of attributes between two different papers and conduct cluster analysis based on the calculated similarity [6,7]. A scheme that uses the metadata of a paper as a feature of deep neural networks has been proposed to discern individuals with the same name [10–15]. A study in which individuals with the same name could be discerned by modeling a graph based on the attributes of papers and author information and by using a graph auto encoder was also conducted [17]. Recently, various studies have been using graph neural networks and a graph embedding to perform learning based on graph modeling of papers and author information to discern individuals with the same name [11–15]. However, these existing schemes only use structured datasets. In actual academic search sites, the available metadata vary across sites; thus, research that considers this should be conducted. For instance, in academic search sites where specific metadata do not exist, weight learning for such metadata cannot be performed, necessitating research to address this gap. In addition, since users may be seeking research materials published on different sites, searching and collecting data from all these sites are essential. Lastly, a method that analyzes name disambiguation based on the information collected from different academic search sites is needed. Information collected from two or more sites may contain overlapping papers and those that exist on only one site. Name disambiguation is imperative in such an environment.

In this paper, we propose a name disambiguation system that enables name disambiguation analysis across different academic search sites by collecting papers from currently active academic search services. The proposed scheme conducts rule-based name disambiguation analysis that can operate dynamically based on metadata from different academic search sites. Additionally, a multi-classifier that can be used in conjunction with existing graph neural network-based name disambiguation schemes is also proposed. The proposed multiclass classification provides a feature to flexibly perform name disambiguation based on the input metadata. The excellence and validity of the proposed method are demonstrated through various performance evaluations.

This paper is organized as follows: The term "individuals with the same name" is defined, and the characteristics and problems of existing name disambiguation schemes are described in Section 2. The proposed name disambiguation method is detailed in Section 3. The superiority of the proposed method is demonstrated through a comparative analysis with existing schemes in Section 4, and the study is concluded, along with future research directions, in Section 5.

2. Related Work

2.1. Name Disambiguation

Rule-based name disambiguation schemes extract distinguishable attributes of authors from papers and create rules using these attributes. Each rule incorporates a weight, and these weights are applied in cluster analysis.

In [7], a rule-based name disambiguation scheme was proposed. The collected data were preprocessed based on established rules. The preprocessed data were then subjected to name disambiguation using one of the two schemes: the rule-based name disambiguation method or a classifier-based method. After collecting documents from the database, attributes, such as surname, first name, co-authors, affiliation, research field, and keywords, were extracted for the name disambiguation scheme. The surname extracted during the preprocessing stage was used as is, while the initial of the first name was included in a data block, along with the attributes. Rule-based similarity was calculated on a block-by-block basis, and hierarchical agglomerative clustering(HAC)

was performed based on the calculated similarity. The “similarity estimated by classifiers” method performed clustering based on similarity scores generated by classifiers and extracted stems from paper titles, abstracts, and keywords. For the extracted stems, similarity was calculated using the term frequency-inverse document frequency (TF-IDF) and latent semantic analysis models. The classifiers were trained using the information from data blocks. HAC was performed based on the similarity scores generated by the classifier. Name disambiguation was performed based on the results of the HAC.

Deep-learning-based name disambiguation schemes generate weights using document attributes, convert these into vector values, and then proceed with deep learning. The values obtained through learning are converted into inter-document distance values, after which cluster analysis is conducted to disambiguate names. The weights for deep learning represent document attributes in the form of graph data, extracting distinguishable attributes among multiple documents to create adjacency and feature matrices. This information is applied to a graph convolutional neural network (GCN) [18] to learn feature vectors. Name disambiguation is then performed based on the learned feature vectors using HAC.

A GCN refers to a type of graph neural network that understands the relationships between vertices in a graph data structure, which predict associations or classify vertices. It integrates a convolution layer to more effectively enhance the learning of attribute vectors than conventional graph neural networks. The graph data, comprising vertices and edges, undergo computation using weight sharing, a process where the same filter is used to train all vertices. During the weight sharing process, redundant attributes operate with the same weights, which enhances the correlation between nodes that have edge relationships. By updating the information of all vertices in this manner, vector values can be determined for cluster analysis.

A name disambiguation scheme based on a GCN was proposed previously [12–14]. From all documents, those with name ambiguity were selected, and name-specific sets were formed to create candidate groups. During the global representation learning process, the attribute information (titles, keywords, co-authors, affiliations, and conferences) of all documents in the selected name-specific sets was extracted to form attribute data. The attribute information was segmented into individual words and then converted into vector values using Word2Vec. A feature matrix of the document, based on attributes, was created following the TF-IDF process using the transformed vector values. In the “three association graphs” process, edge creation conditions were set for name-specific sets of the candidate group. If the conditions exceeded a threshold, an edge relationship formed between two vertices, resulting in the creation of an adjacency matrix that integrated the edge information between vertices. The types of graphs produced included paper-to-paper graphs, co-author graphs, and paper-to-author graphs. The GCN was performed using the created adjacency and feature matrices. Ultimately, based on the learned feature vectors, HAC was carried out for name disambiguation.

2.2. Limitations of Previous Studies

Traditional name disambiguation schemes use pre-constructed structured datasets. Name disambiguation on actual research materials from heterogeneous academic search services in operation faces an issue: if metadata not considered in the name disambiguation dataset are absent, direct application of the name disambiguation scheme becomes unfeasible. Furthermore, as academic search services vary in the type of research materials they offer based on their purpose, different academic services must be searched to review works of a researcher published in various formats. Even if the research materials are of the same type, the absence of shared metadata between different academic search services can make name disambiguation exceedingly difficult without separate preprocessing. For instance, while some research materials may list affiliations as general as “Chungbuk National University,” others might provide detailed affiliations such as “Chungbuk National University, Information and Communication Engineering,” necessitating data preprocessing.

Additionally, as academic search services provide research materials specific to their purpose, finding all works of an author in one academic search site can be challenging if they have published different types of research materials. For example, if an author publishes a paper based on a particular research project and produces reports or patents as research outcomes, searching for these different types of research materials within a single academic search service becomes very difficult. Ultimately, users have to search for research materials on multiple academic search services. Hence, in this study, we collected research materials from various operational academic search services and performed name disambiguation analysis on the collected material. We also considered the metadata from various academic search services to apply a uniform preprocessing approach. This led to the advantage of identifying potential attributes to consider in the name disambiguation scheme.

3. Proposed Name Disambiguation Scheme

3.1. Overall structure

Traditional name disambiguation schemes, such as rule-based [7] and deep-learning-based schemes [13], use structured datasets. In this study, a method for name disambiguation by directly collecting data from multiple operational academic search sites was proposed. The proposed method standardizes affiliation information using a dedicated affiliation table when the affiliation is a university. Then, rules are defined using commonly used metadata to disambiguate names.

The metadata provided by existing academic search sites are diverse. Name disambiguation schemes do not consider the diversity of metadata provided by each academic search site. To consider this diversity of metadata, a method should selectively execute the name disambiguation scheme based on the input data and exhibit the best performance. Additionally, even if some metadata can be used in the proposed method, if the expected performance is inferior to that of existing schemes, applying the existing methods is more effective. We propose a multi-classification scheme that considers all these situations. When metadata are input, multi-classification is performed based on the rule-based scheme proposed in this paper, as well as the existing deep-learning-based name disambiguation method.

The multi-classifier uses limited metadata from the actual data required by each method to select the most suitable scheme for name disambiguation. The multi-classifier is designed in an expandable manner to also consider new name disambiguation schemes that may emerge in the future.

Figure 1 presents the overall system architecture of the proposed scheme. The collector gathers research papers, project data, and affiliation information of research outcomes from academic search services. The preprocessor creates a set of name-ambiguity candidates with identical names, considered as subjects in this study, from all the documents collected by the collector. The collected affiliation information is transformed into a standardized form using an affiliation table. Attributes to be used for name disambiguation are then extracted. In the analyzer, the preprocessed attribute data are used to analyze the similarity between all documents of the name-ambiguity candidate group using both the rule-based and deep-learning schemes. Finally, in the discriminator, the analyzed document similarity data are represented as a distance matrix for clustering execution, and HAC is performed to divide the clusters by unique author documents and disambiguate names.

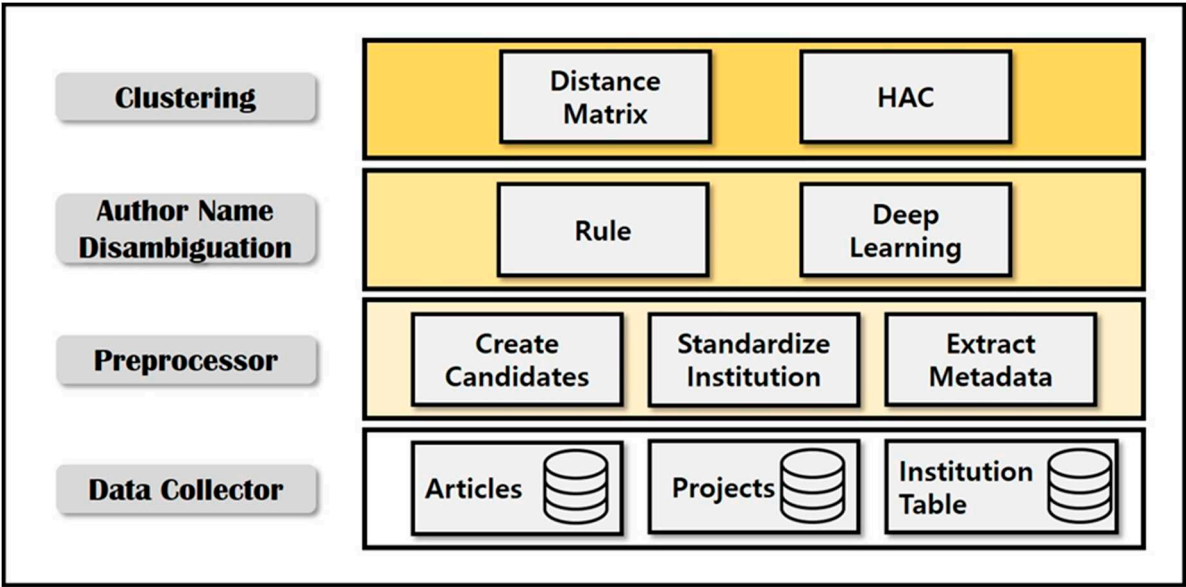


Figure 1. System Architecture of the Proposed Scheme.

3.2. Data Collector

The documents targeted for collection in this study were offered by heterogeneous academic search services, including research papers, as well as national R&D, patent, and research reports. Since heterogeneous forms of documents were collected from heterogeneous sites, understanding the metadata necessary for name disambiguation was crucial.

The collector, upon user keyword input, collects the attribute information of documents appearing in the keyword search results and stores it in the internal database. The collected data are used as attributes for analysis in the name disambiguation analyzer after preprocessing in the next preprocessing step. When collecting the material, understanding the metadata used by the heterogeneous academic search services providing research outcomes is essential. For instance, academic search services, primarily offering papers, contain metadata distinguishing between academic journals and conferences. However, sites that primarily offer project information do not have such distinguishing metadata. Among them, whether they support metadata to identify international journal listings also varies. Moreover, domestic academic search sites that use journal information as metadata express it in various forms, such as journals, academic journals, and proceedings, which should be considered during data collection.

Next, understanding the attributes of various types of documents is essential. In academic search services, a significant proportion of authors publishing papers are affiliated with universities. However, for sites providing R&D information, authors publishing research outputs have various affiliations, including research institutes, national departments, companies, and universities. Furthermore, the authors of papers can be categorized into the main, co-, and corresponding authors. However, for R&D, the authors consist of participating researchers and research leaders. Thus, depending on the type of research output, the nature and form of attributes differ. Understanding the meaning of similar attributes and collecting them accordingly are crucial steps.

The collector gathers all data usable for name disambiguation schemes. The research outputs of an author stored in the database after collecting the necessary values from academic search sites are listed in Table 1. All metadata that can be collected from academic search services are gathered. Among the collected metadata, the commonly used attributes and attributes for which advantageous weights can be given for name disambiguation are identified. Commonly usable metadata include the author name and affiliation, co-authors, title of the document, document keywords, and publication year. The attributes that can be given favorable weights for name disambiguation include the research field, email, academic journals, and academic conferences.

The attributes of research outputs used in the proposed scheme are listed in Table 1. Since the proposed method considered various academic search services, it used commonly existing metadata, such as the author name and affiliation, co-authors, publication year, academic journals, and academic conferences, as attribute values for rule-based name disambiguation schemes.

Table 1. Example of Collected Attributes.

Feature	Paper 1	Paper 2
title	An Author Name Disambiguation Method Considering Metadata Features	Development of Fuel Cell System Considering Weight
abstract	A same-name identification scheme that considers metadata to identify people with the same name on heterogeneous sites ...	Energy commercialization considering weight using an ultra-light tube-type fuel cell system ..
Keywords	Name disambiguation, Metadata	Fuel cell, Tube type
Year	2018	2020
Affiliation	Chungbuk National University	Pohang University of Science and Technology
First Author	Junhyeok Jang	Junhyeok Jang
Co-author	Sanghyeok Kim, Yuna Kim, Dojin Choi, Jaesoo Yoo	Taehyeong Kim, Jinyong Lee, Sunkyu Han, Minkyoo Lim
Journal	Big data technology journal	Resource technology journal
Publisher	Big Data Society	Society for New and Renewable Energy
e-mail(s)	dataman@kakao.com	azeez448@nate.com
Research area	bigdata	Energy, resource tech
Research Period	2018~2020	2018~2022

3.3. Preprocessing

Data collected directly from the collector cannot be used as they are; therefore, preprocessing is required. In the preprocessor, all documents containing the same author name are gathered as potential name disambiguation candidates. Attributes needed for name disambiguation analysis are then extracted from the documents. At this point, affiliation information is normalized using the affiliation table.

From the collected documents, candidate groups for name disambiguation need to be generated to narrow down the set of documents that can be considered as potential name matches. Name disambiguation is performed using document similarity within the created candidate sets. In this study, two or more documents with the same author name are considered as candidate groups.

Figure 2 shows an example of a name disambiguation candidate group. Collected data containing two or more documents with the author name "Jang Jun-hyuk" were generated as name disambiguation candidates. Within the name disambiguation candidate group, attributes, such as the title of the paper, affiliation, publication year, co-authors, journals, and academic conferences, were collected. To help explain the name disambiguation scheme proposed in this paper, documents "Jang Jun-hyuk_0," "Jang Jun-hyuk_1," and "Jang Jun-hyuk_2" represented unique research outputs of a single Jang Jun-hyuk author, while document "Jang Jun-hyuk_3" represented a research output of a different Jang Jun-hyuk with the same name. The proposed scheme constitutes the name disambiguation document candidate group from all research outputs with the same author name, regardless of the type of authorship (main author, co-author, or corresponding author). Each document in the name disambiguation candidate group was labeled with the author name, and numbers were appended after the name to distinguish documents.

	author	name	title	inst	year	co-au	joc
0	Jang Jun-hyeok	Jang Jun-hyeok_0	Author Name Disambiguation Tasks Considering Metadata features	Chungbuk National University	2018	[Jang Jun-hyeok, Kim Sang-hyeok, Kim Yoon-a, Choi De-jin, Yoo Jae-soo]	Korea Bigdata Society
1	Jang Jun-hyeok	Jang Jun-hyeok_1	Metadata Learning by Using Machine Learning	Chungbuk National University	2021	[Jang Jun-hyeok, Kim Yoon-a, Choi De-jin, Yoo Jae-soo]	Korea Bigdata Society
2	Jang Jun-hyeok	Jang Jun-hyeok_2	Pitcher's Contribution to ERA by the Batting Lineup	Sports Science Technology Company	2022	[Jang Jun-hyeok, Kim Yoon-a, Choi De-jin, Yoo Jae-soo]	Korea Bigdata Society
3	Jang Jun-hyeok	Jang Jun-hyeok_3	Fuel Cell System Considering Weight	Pohang University of Science and Technology	2020	[Jang Jun-hyeok, Kim Tae-hyeong, Lee Jin-youn, Han Sun-kyu, Lim Min-gyo]	The Korean Society for New and Renewable Energy

Figure 2. Example of a Name Disambiguation Candidates.

In this study, we normalized the affiliations listed in documents. The listing of affiliations (when the author affiliation is a university) can vary across academic search services. Furthermore, authors may have different styles of listing their affiliations. For example, an affiliation Pennsylvania State University, depending on different styles, can be written as “Pennsylvania State University, Pennsylvania State Univ., PSU, Penn. State Univ., or Penn. State College,” among other variations. Additionally, in some instances, such as “Information Sciences and Technology, Penn. State Univ.,” a specific department or detailed affiliation information is included. For such cases, a standardized affiliation form needs to be normalized. Web of Science, a globally renowned academic search service, provides affiliation metadata to alleviate confusion caused by various affiliation entries and to verify various forms of affiliation information. Leveraging this, the same affiliation information can be normalized. In our research, affiliations listed in various forms were normalized to a unified format. All institution names and their synonyms listed on academic search services were stored in a database. Based on the stored information, synonymous institution names were standardized into a representative institution name for affiliation notation. Affiliations with detailed information (e.g., departments) were converted into only the university name to efficiently process the affiliation information.

After creating the name disambiguation candidate group, attributes to be used in the name disambiguation algorithm were extracted. Notably, the variation in metadata across academic search services must be considered. Some academic search services provide information only about the paper, while others offer data like R&D research reports. The diverse metadata provided by academic search services must be preprocessed into a commonly usable format. The preprocessor extracted and used only the metadata to be input into the name disambiguation analyzer.

3.4. Author Name Disambiguation

The name disambiguation analyzer uses the name disambiguation candidate group created by the preprocessor. It calculates the similarity between documents within the candidate group to compare whether two documents were written by unique authors. The name disambiguation analyzer defines a method to calculate the similarity between attributes of two documents. It computes the sum of similarities between attributes to determine the final similarity score. To discern significant attributes, weights are assigned to attributes based on both the rule-based and deep-learning schemes.

3.4.1. Rule-based Scheme

The attributes to be used in the name disambiguation analyzer were extracted by the preprocessor. The name disambiguation analyzer defined rules for disambiguating names and assigned weights based on the importance of each rule. The similarity calculation rules and weights of the proposed rule-based scheme are listed in Table 2. The proposed rule-based approach first performed name disambiguation based on exception cases. An exception case refers to a specific rule that determines whether two documents are written by the same author. If a document does not fall under an exception case, the similarity is calculated according to the proposed four rules. The similarity values for each rule are summed up, and if the total similarity exceeds a certain threshold, the research is determined to have been authored by the same individual.

Table 2. Attribute and weight application rules.

Name	Rule	Weight
Exception Case	The titles are exactly the same	4
	The author affiliations are exactly the same	
	The co-authors are exactly the same	
	Jaro-Winkler Similarity [19]	
Affiliation	s_j	0 ~ 1
	$= \begin{cases} 0, & m = 0 \\ \frac{1}{3}(\frac{m}{ s_1 } + \frac{m}{ s_2 } + \frac{m-t}{m}), & otherwise \end{cases}$	
	$s_w = s_j + \omega p(1 - s_j)$	
Year	Difference of publication year	0 ~ 1
	$p = -\left(\frac{ y_{d1}-y_{d2} }{cy}\right) - 1$	
	(1)	
Co-author	Number of identical co-authors	0 ~ 0.5
	$c = \frac{1-e^{- x }}{2}$	
	(2)	
Co-author	Proportion of identical of co-authors	0 ~ 0.5
	$r = \frac{x/y}{2}$	
	(3)	
venue	exactly the same or not	0 or 1

First, if the two documents match the exception case attributes, they are awarded four points, and other attributes are not considered. The first exception case is when the titles of the two documents being compared are the same. The second exception case is when, after normalizing the affiliation of the document through the affiliation table in the preprocessor, the affiliations are found to be identical. In case of a discrepancy in the detailed affiliation, the Jaro–Winkler [19] similarity, a method that considers the number and position of common characters between two strings, was used to calculate the similarity of affiliations. If the Jaro–Winkler similarity score exceeded a predefined threshold, the documents were deemed to have the same affiliation.

The publication year attribute represents the difference in publication years between the two papers. Equation 1 represents the weight calculation based on the publication year (p). Here, y_{d1} and y_{d2} indicate the publication years of the two documents, and cy represents the publication year span set by the user. In this study, the volume of documents collected by the collector varied widely depending on the keywords entered when collecting documents. Moreover, when new subject keywords emerge, past data need not be collected. Therefore, setting the publication year when collecting documents was necessary.

For the co-author count attribute, the number of identical co-authors in the two papers was compared. Equation 2 displays the weight calculation based on the number of co-authors (c), where x is the number of identical co-authors between the two documents. The co-author ratio attribute represents the ratio of identical co-authors to the total number of co-authors in the two papers. Equation 3 represents the weight calculation based on the co-author ratio (r). In this context, x is the number of identical co-authors, similar to the co-author count formula, and y is the total number of co-authors in the document with more co-authors among the two being compared.

For the journal and conference attributes, the names of the journals and conferences where the two papers are published were compared. If the two documents were identical, a weight of one was

assigned; if they were not identical or did not include the journal and conference attributes, a weight of zero was assigned.

Figure 3 shows an example of weight determination based on the rules listed in Table 2. For the affiliation attribute, the Jaro–Winkler distance value was derived from the redundant words “Chungbuk National Univ.” and “Chungbuk National Univ. Bigdata Depart,” resulting in the s_j value. Using the s_j value, the Jaro–Winkler similarity value, represented as s_w , was calculated, and a score of 0.83 was obtained. The publication year attribute defines the data collection period as 5 years. By using the absolute value function to calculate the publication year difference between the two documents, a score of 0.4 was assigned. The co-author attribute determines the number of identical co-authors in the two papers, excluding the co-author with the same name, “Jang Jun-hyuk.” The co-author ratio divides the number of co-authors by the number of co-authors in the document with the larger number of co-authors. Here, the combined values of the co-author count and ratio were halved, and scores of 0.475 and 0.375, respectively, were obtained. Thus, the co-author attribute value was represented as 0.85. For the journal and conference attributes, since conferences of both the documents were “Journal of Bigdata,” they were identical, and a score of one was assigned. By adding the results from all the rules, the final weight was determined. The weight of the sample documents “Jang_0” and “Jang_1” was 3.08, indicating a high similarity.

Inst.	Chungbuk National Univ.	Chungbuk National Univ. Bigdata Depart.	$s_j = \frac{1}{3} * (\frac{13}{13} + \frac{13}{17} + \frac{13-8}{13})$ $s_w = s_j + (4*(0.1)*(1-s_j))$	0.83
Pub. Year	2018	2021	$-(\frac{ 2018 - 2021 }{5}) - 1$	0.4
Co-author (number)	S.H. Kim, Y.N. Kim, D.J. Choi, J. S. Yoo	Y.N. Kim, D.J. Choi, J. S. Yoo	$\frac{1 - e(-3)}{2}$	0.475
Co-author (ratio)	S.H. Kim, Y.N. Kim, D.J. Choi, J. S. Yoo	Y.N. Kim, D.J. Choi, J. S. Yoo	$(\frac{3}{4}) / 2$	0.375
Journal	Journal of Bigdata	Journal of Bigdata	Journal of Bigdata == Journal of Bigdata	1
Total (0.83+0.4+0.475+0.375+1)				3.08

Figure 3. Example of Name Disambiguation Rules.

3.4.2. Deep-Learning-based Scheme

The deep-learning-based analysis, similar to rule-based analysis schemes, uses major attributes from the metadata that can serve as distinguishing factors for individuals with the same name. These attributes were represented in a graph, and the GCN was used to learn the hidden features of the paper.

Deep-learning analysis uses various attributes, such as the title, keywords, abstract, co-authors, publication year, and journal data. As the document title and abstract were used as attributes, major keywords from these attributes were extracted using natural language processing packages, such as konlpy and NLTK. The extracted keywords were then converted into a vector form using FastText.

This converted vector was used as the input vector for the deep-learning model. In this study, the converted vectors were constructed in the form of a triplet network. The triplet network structure, as shown in Figure 4, represents vectors by placing vectors with similar and dissimilar values close to each other and further apart, respectively. In the (A) triplet network, anchor refers to a document of a unique author, positive refers to another document of that unique author, and negative refers to a document of an author with the same name but not the unique author. With the anchor document as a reference, the objective of the triplet network is to bring the documents corresponding to the positive closer and drive those corresponding to the negative far apart. As shown in Figure 4 (B), a pid triplet transformed the vector value of the converted paper into an optimal vector value using the vector value of another paper. Initially, P_{pj} referred to a paper similar to the pid paper (with author, title, publication year, keywords, journal info, etc.), and N_{pk} referred to a paper dissimilar to the pid paper. The aim of the triplet was to calculate the distance between vectors, bringing similar vectors closer and pushing dissimilar vectors further apart. Therefore, based on the similarity in the information of the papers, the vector values of all the input papers were converted into the form of a triplet network, as shown in Figure 4 (A).

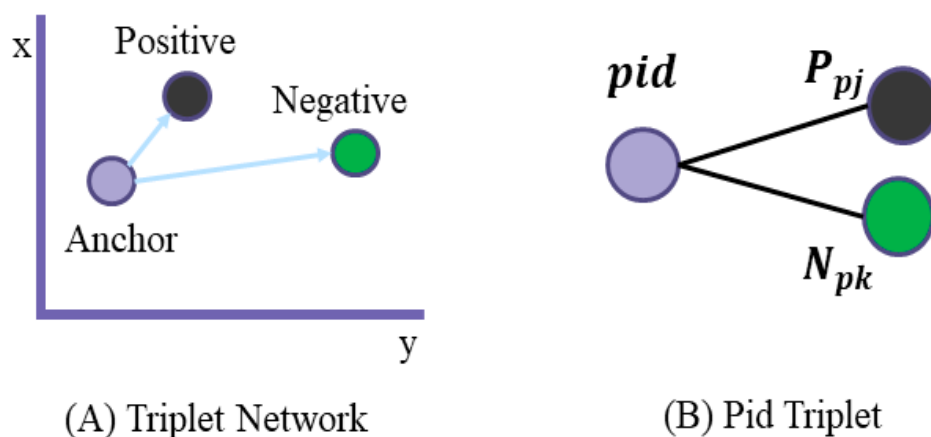


Figure 4. Triplet Network.

3.5. Clustering

To distinguish between individuals with the same name, documents in the pool of potential matches must be clustered by unique authors. In the clustering stage, the similarity values generated from the previous name disambiguation analyzer were used. The similarity values were converted into distance values to be used as input data for cluster analysis.

In this section, discerning between individuals with the same name using both rule-based and deep-learning schemes of the name disambiguation analyzer is discussed. Using the similarity between documents of authors with the same name, determining whether the author of a paper being compared is indeed the same author is necessary. First, to convert inter-document similarity into a distance value used in HAC, a distance matrix transformation was performed. Using the converted distance values, HAC was executed to distinguish between individuals with the same name.

HAC used in the name disambiguation scheme compared distances between clusters to perform clustering. For this, the document similarity generated by the name disambiguation analyzer was converted into a distance value. The inter-document similarity was represented in the form of a similarity matrix. The process of converting the similarity matrix into the distance matrix using the distance conversion formula, $sim2diss$, is explained next [20].

The similarity matrix is symmetrical in nature; hence, $N(N-1)/2$ pairs of a, b were generated. The distance value was computed for all pairs, and the results were organized in matrix form. Since distance values were calculated for every a, b pair, the matrix was a square matrix. The similarity values ranged between zero and four, based on a weighted application rule with four as the maximum score. The diagonal terms of the matrix were "0," as they represented the distance to

oneself, making the matrix symmetrical about its diagonal. Regarding the size of the similarity matrix: in case of d documents, it compared all documents of authors with the same name to generate inter-document similarities, representing them in a $d \times d$ matrix.

Figure 5 illustrates the method of constructing the similarity matrix using inter-document similarities. The input data A, B, and C have a total of three distance values, and the distances between all pairs, such as A-B and A-C, can be represented as a 3×3 matrix. Figure 5(a) displays a graph containing inter-document similarity values calculated from the name disambiguation analyzer. The distance between input data A and B is represented as one, A and C as three, and B and C as two. Figure 5 presents the similarity matrix in matrix form and shows all distance values from the graph data.



Figure 5. Example of Creating a Similarity Matrix.

An example of representing inter-document similarity values in the form of a similarity matrix is listed in Table 3. Each row and column, such as Jang_0, Jang_1, Jang_2, and Jang_3, represents candidate documents with the same name. As every document is compared 1:1 for similarity, it forms a symmetrical matrix. Each element in the symmetrical matrix signifies the similarity values between the compared documents. Documents Jang_0 and Jang_1 exhibited a similarity value of four; therefore, both the documents were concluded to have been written by the same author.

Table 3. Similarity matrix example.

	Jang_0	Jang_1	Jang_2	Jang_3
Jang_0	0	4	1.9	0.6
Jang_1	4	0	2.6	0.8
Jang_2	1.9	2.6	0	0.6
Jang_3	0.6	0.8	0.6	0

HAC calculates the distance between clusters and performs clustering based on these distance values. Therefore, in this study, we converted similarity values into inter-cluster distances. Equation 4 is a function provided in the statistical solution program R [28], which converts inter-document similarity into the inter-document distance value, known as the distance matrix, using the *sim2diss* formula.

$$1 - \left(\frac{\text{Similarity}}{\text{MAX}} \right) \quad (4)$$

The rule-based name disambiguation scheme represents the generated similarity as a distance value using rules based on document attribute values. Therefore, MAX in Equation (4) corresponds to the total number of attributes. When the similarity of two documents is indicated as a perfect score of four points, the distance value of these two documents can be expressed as $1 - (4/4) = 0$. Since it represents the distance between two documents, a higher similarity results in a smaller distance value. The range of the distance value is between zero and one. Contrary to when calculating similarity, a distance value closer to zero indicates higher similarity between the two documents.

The similarity matrix example from Table 3, which utilizes the *sim2diss* formula, was represented in the form of a distance matrix, as summarized in Table 4. After calculating the *sim2diss* formula for all similarities, it was represented in matrix form. Documents Jang Jun-hyuk_0 and Jang Jun-

hyuk_1 had a distance value of zero, indicating similarity. The values in the white cells represent the distance matrix values calculated using the *sim2diss* formula, while the values in the yellow cells are set to one as they compare the same documents.

Table 4. Distance matrix example.

	Jang_0	Jang_1	Jang_2	Jang_3
Jang_0	1	0	0.525	0.85
Jang_1	0	1	0.35	0.8
Jang_2	0.525	0.35	1	0.85
Jang_3	0.85	0.8	0.85	1

In the distance matrix phase, names were disambiguated based on the distance values obtained from the document similarities. The *AgglomerativeClustering* model in Python was used. When performing HAC, the number of formed clusters is uncertain. Hence, the hyperparameter value *n_clusters* determining the number of clusters was set to none. As previously described, the pre-calculated distance was used to represent the distance matrix; therefore, affinity was set to pre-computed. Additionally, distance measurement methods, such as single, complete, and average linkages, were used. After an intrinsic evaluation, the most suitable linkage method was selected. Finally, the *distance_threshold* value, which sets the stopping criterion for the clustering process, was also determined through intrinsic performance evaluation to determine the most suitable value.

Figure 6 shows an example of a dendrogram, using which the results of the HAC were visualized. Figure 6 shows the grouping of clusters. The red dotted line in the figure represents the stopping criterion, *distance_threshold*. Clusters grouped by the stopping criterion are marked in orange, while those not grouped are marked in blue. Clusters divided by the stopping criterion indicate individuals with the same name.

As an example from Table 4, when setting the stopping criterion to 0.4, the documents “Jang_0,” “Jang_1,” and “Jang_2” were grouped into one cluster, while “Jang_3” was represented in a different cluster. In other words, two authors were distinguished.

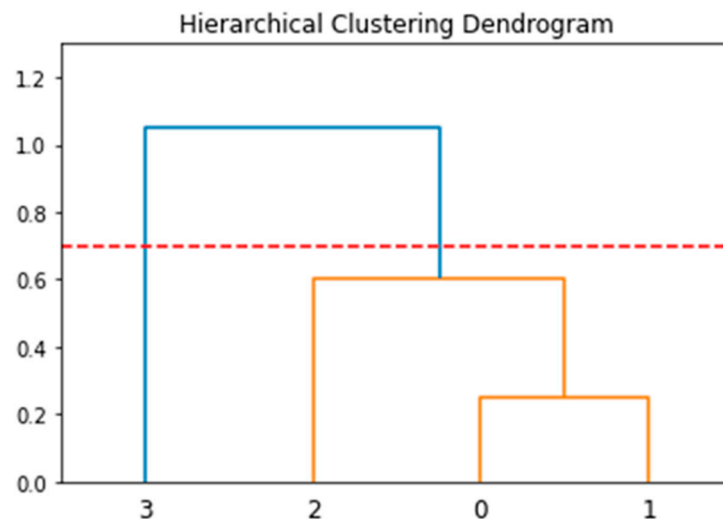


Figure 6. Example of a Dendrogram.

3.6. Multiclass classification.

In this study, name disambiguation was conducted by collecting data in real time from academic search websites. Metadata provided by heterogeneous academic search sites vary depending on the site characteristics. Therefore, name disambiguation needs to consider these varying characteristics. Not all academic search sites hold the metadata required for name disambiguation. Hence,

conducting name disambiguation using the available metadata from these academic search services was necessary.

The name disambiguation method proposed in this paper employed both rule-based and deep-learning schemes. The rule-based approach has the advantage of quickly disambiguating names when documents with the applicable metadata are input based on set rules. Conversely, although the deep-learning scheme requires training time, it can perform name disambiguation even in the absence of essential metadata. Therefore, even with the same metadata, diverse name disambiguation schemes can be applied to obtain results. In this paper, a multi-classifier that can select the appropriate name disambiguation scheme using the collected metadata is proposed. Using the multi-classifier, even if missing data are input, the appropriate name disambiguation scheme can be chosen to derive results. This method is scalable, that is, it can allow future addition of new academic search services or new name disambiguation classifiers based on the results from the multi-classifier.

Figure 7 displays the schematic of a multi-classifier that considers metadata from various academic search services and accordingly selects the appropriate name disambiguation method. When embedded name disambiguation data from name disambiguation candidates are input, the proposed multi-classifier selects between the presented rule-based and learning name disambiguation, considering the presence or absence of input metadata attributes.

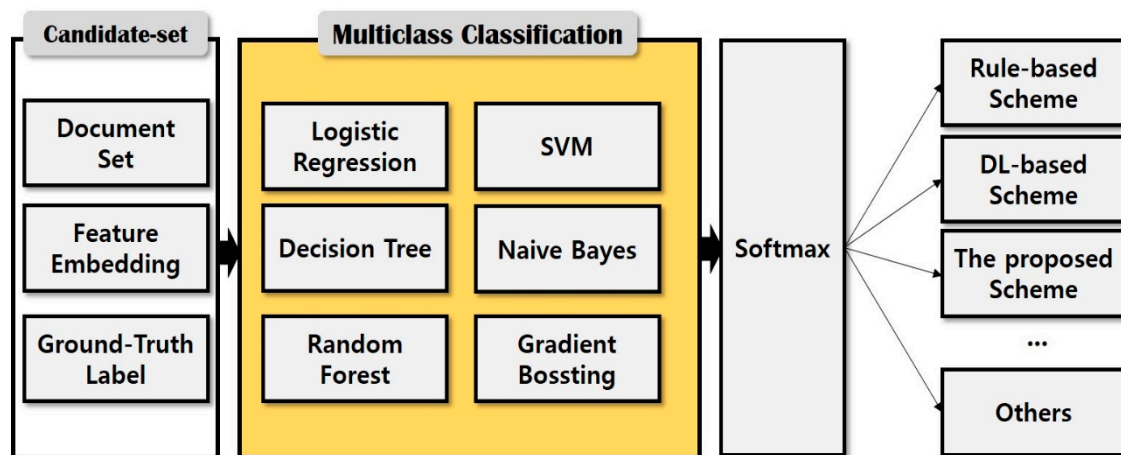


Figure 7. Overview of the multiclass classification structure.

The working process of the multi-classifier is as follows: First, the attributes of name disambiguation candidate documents were converted to vector form through the feature embedding process. Using the transformed feature values, the existing rule-based name disambiguation, existing deep-learning-based name disambiguation, and proposed scheme were executed, followed by performance evaluation. By comparing the F1-measure values obtained from the performance evaluation, the best identification scheme was designated as the correct label for the feature values. For example, labels were assigned based on features, such as zero for rule-based name disambiguation and one for deep-learning-based name disambiguation. After all the correct labels were generated, training was conducted using these data and various existing multi-classifiers. The trained multi-classifier then received input values that were converted to vectors through the feature embedding process of site-specific metadata. Ultimately, the multi-classifier produced the most appropriate identification scheme as its output. Thus, selecting the most suitable name disambiguation scheme in real-world environments becomes possible where various metadata are generated. Furthermore, even when a new identification scheme is introduced, the multi-classifier can be extended by adding one label, enabling the utilization of an expandable multi-classifier model.

4. Performance Evaluation

4.1. Performance Evaluation Environment

The performance of the proposed name disambiguation scheme was evaluated based on the termination criterion and linkage method of the HAC to validate its utility. In this study, the performance of the proposed and existing rule- and deep-learning-based name disambiguation schemes was comparatively evaluated [7,13]. Moreover, the performance of the multi-classifier, which selects a name disambiguation scheme based on attributes, was evaluated. The performance evaluation environment is summarized in Table 5. The performance evaluation was conducted on a system built with an Intel(R) Core(TM) i5-9600K CPU @ 3.70 GHz 64-bit processor and 32 GB of memory. The proposed scheme was implemented using Python 3.8.12 in the Python Anaconda environment, and machine-learning libraries, such as sklearn, keras, tensorflow, and the matplotlib library for data visualization, were used [21–25].

Table 5. Performance Evaluation Environment.

Name	Value
Processor	Intel(R) Core(TM) i5-9600K CPU @ 3.70 GHz
Memory	32 GB
OS	Window 10 Education
Language	Python 3.8.12.
Platform	Python Anaconda custom

The collected dataset is summarized in Table 6. The dataset used for the performance evaluation includes all research results published in the last 1-10 years. It includes a total of 23,563 entries from academic and project databases, such as NTIS, DBPIA, KCI, and SCIENCEON. These entries are based on search keywords like “database indexing,” “IoT applications,” “cloud computing,” “big data social network,” “AI verification,” “virtual reality,” and “steering control.” Among all the collected research materials, 2,460 name disambiguation candidate groups were created, targeting materials with the same author name appearing in two or more research outputs. The attributes of the collected data consisted of paper ID, co-authors, author ID, document title, academic journals and conferences, affiliation, publication year, etc. The performance evaluation of the proposed name disambiguation scheme consisted of its own performance evaluation, comparative performance evaluation with the existing name disambiguation schemes, and performance evaluation of the multi-classifier. To measure the accuracy of the proposed scheme, the precision, recall, and F1-measure were calculated. An intrinsic performance evaluation of the proposed scheme and a comparative evaluation with other name disambiguation schemes were conducted.

Table 6. Dataset.

Keyword	Period	NTIS	SCIENCEON	DBPIA	KCI	Total
Database & Index	10	208	47	966	76	1,297
IoT & Application	5	2,889	138	3,302	261	6,590
Cloud Computing	3	981	138	1,459	293	2,871
Bigdata & SNS	10	471	153	1,910	71	2,605
AI & Verification	2	2,826	104	2,176	227	5,333
AR / VR	1	540	85	1,870	335	2,830
Steering & Control	10	289	76	1,561	111	2,037
Total	-	8,204	741	13,244	1,374	23,563

4.2. Intrinsic Performance Evaluation

Clustering in HAC requires setting a termination criterion to distinguish unique clusters. In this section, the performance evaluation based on the termination criterion of HAC is discussed. In the proposed method, authors may be clustered differently depending on the termination criterion when using HAC. Therefore, in this study, various termination criterion values were set, and experimental evaluations were conducted as an intrinsic performance evaluation method.

Figure 8 displays the performance evaluation results based on the termination criterion. The results for the precision and F1-measure are represented in a bar graph. Performance evaluation was conducted by changing the termination criterion values from 0.2 to 0.6. The termination criterion of 0.2, which showed the highest F1-measure value of 0.95, was determined to be the most suitable termination criterion. A termination criterion of 0.2 means that if the distance between documents (or clusters) is closer than 0.2, they are clustered, and if not, they are not clustered. Through experimental evaluations, the termination criterion of 0.2 was used as the HAC standard in the proposed scheme.

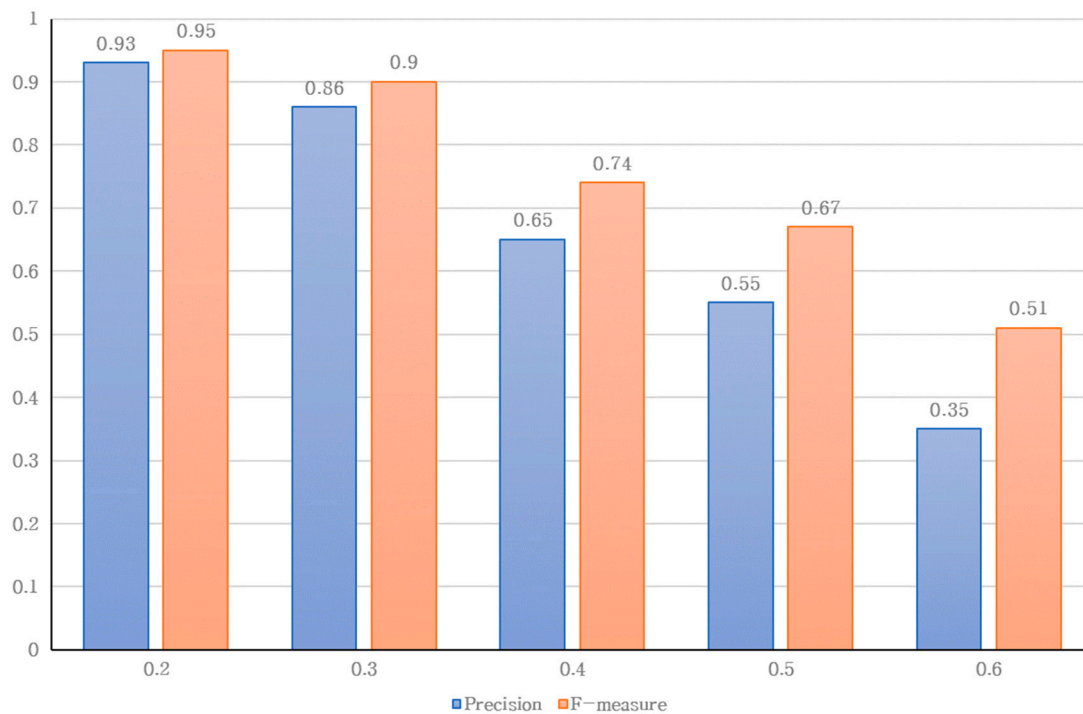


Figure 8. Precision and F1-measure According to the HAC Termination Criterion.

A performance evaluation was conducted based on the linkage method setting, which is one of the hyper-parameters of HAC. In the proposed scheme, when implementing the rule-based name disambiguation method using HAC, the clustering results can vary depending on the linkage method, similar to that with the termination criterion. Therefore, in this study, an experimental evaluation for each linkage method was conducted as an intrinsic performance evaluation method to determine the optimal linkage method. The termination criterion was set to 0.2, as determined through performance evaluation in the previous step. Figure 9 displays the performance evaluation results based on the linkage method. Three linkage methods were compared: single, complete, and average, excluding Ward's linkage, which cannot be used in HAC. According to the experimental results, the complete linkage method exhibited a higher F1-measure value than the other two methods. Therefore, the complete linkage method, which yielded the highest scores for precision and F1-measure, was adopted as the linkage method in the proposed scheme.

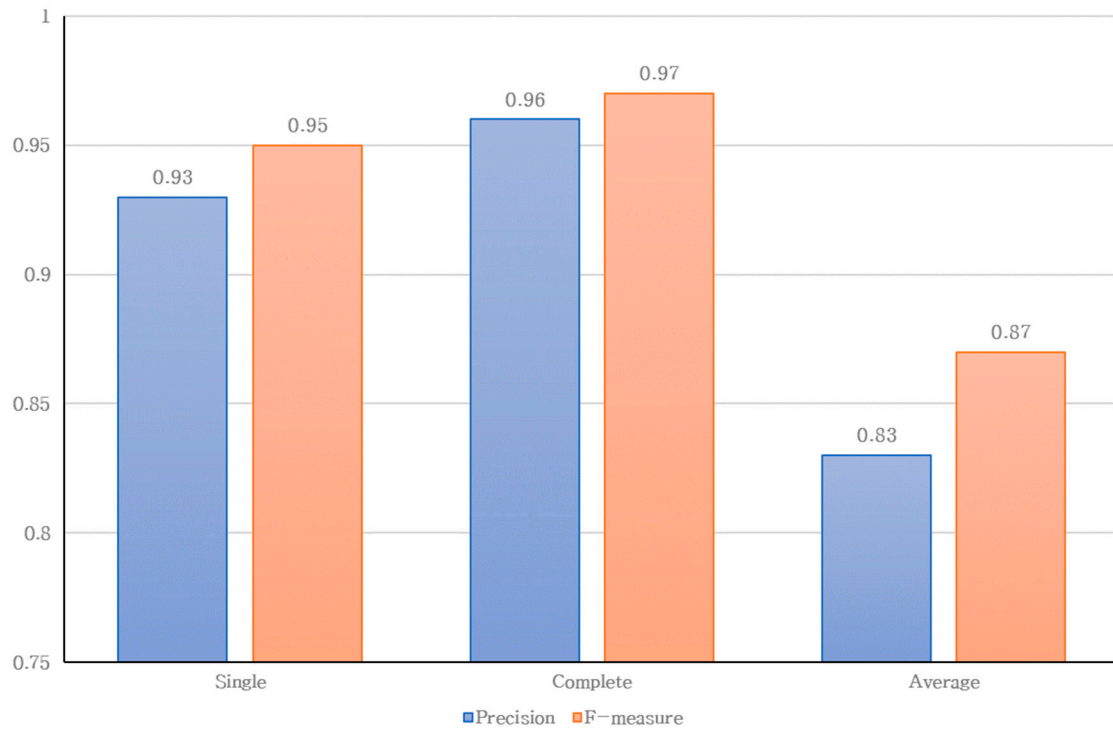


Figure 9. Precision and F1-measure According to the HAC Linkage Method.

4.3. Comparative Performance Evaluation of Name Disambiguation Schemes

A performance comparison with existing name disambiguation schemes was conducted to demonstrate the superiority of the proposed method. In this study, two schemes were compared: 1) the Protasiewicz method [7], an existing rule-based name disambiguation scheme, that uses the attributes of papers to create rules and runs HAC with the weights of these rules to disambiguate names. 2) The Chen Ya method [13], an existing deep-learning-based name disambiguation scheme, that learns paper attributes using a GCN and then runs HAC with the resulting weights to disambiguate names. The HAC of the proposed scheme was set with a termination criterion of 0.2 and used the complete linkage method. The dataset used for performance evaluation is summarized in Table 6. The superiority of the proposed method was demonstrated through a comparative performance evaluation of precision, recall, and F1-measure for name disambiguation accuracy between the proposed and existing methods.

Figure 10 shows the results of the comparative performance evaluation of precision based on the name disambiguation schemes. The precision of the proposed scheme exhibited a very high performance, with scores >0.99 for all keywords. However, the existing rule-based scheme, the Protasiewicz method, showed a decent average performance but underperformed for certain keywords. Additionally, the deep-learning-based scheme demonstrated the poorest average performance. The data used for the performance evaluation were collected from actual academic search services. This indicated that existing studies need more detailed preprocessing and analysis when performing name disambiguation analysis based on real data. Furthermore, it implies that these characteristics must be reflected since not all academic search services provide the same metadata.

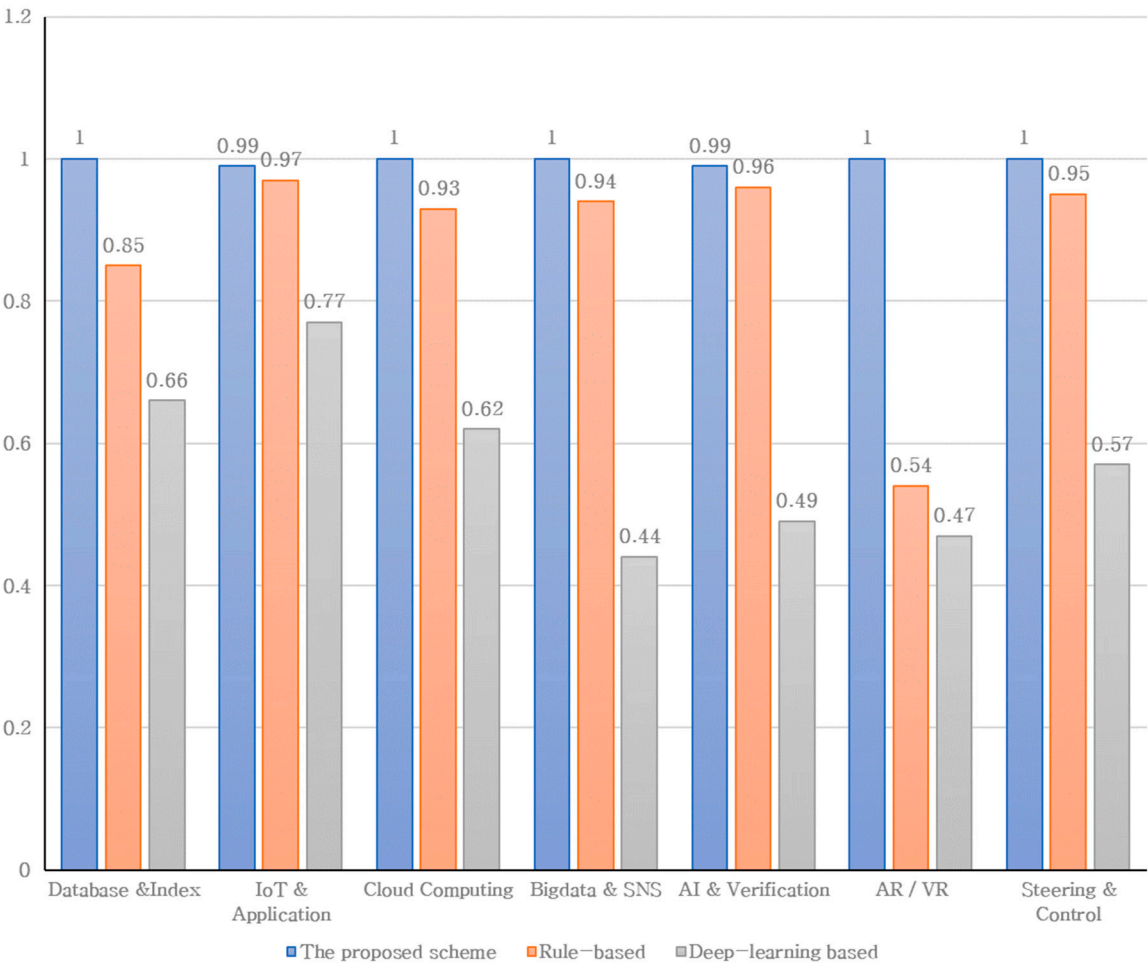


Figure 10. Precision According to the Name Disambiguation Schemes.

Figure 11 displays the results of the comparative performance evaluation of recall based on the name disambiguation schemes. The recall of the proposed scheme demonstrated a very high performance, with scores >0.97 for all keywords. The existing deep-learning-based Chen Ya method showed excellent performance for keywords “cloud computing,” “AI verification,” and “steering control;” however, the proposed method outperformed for all keywords.

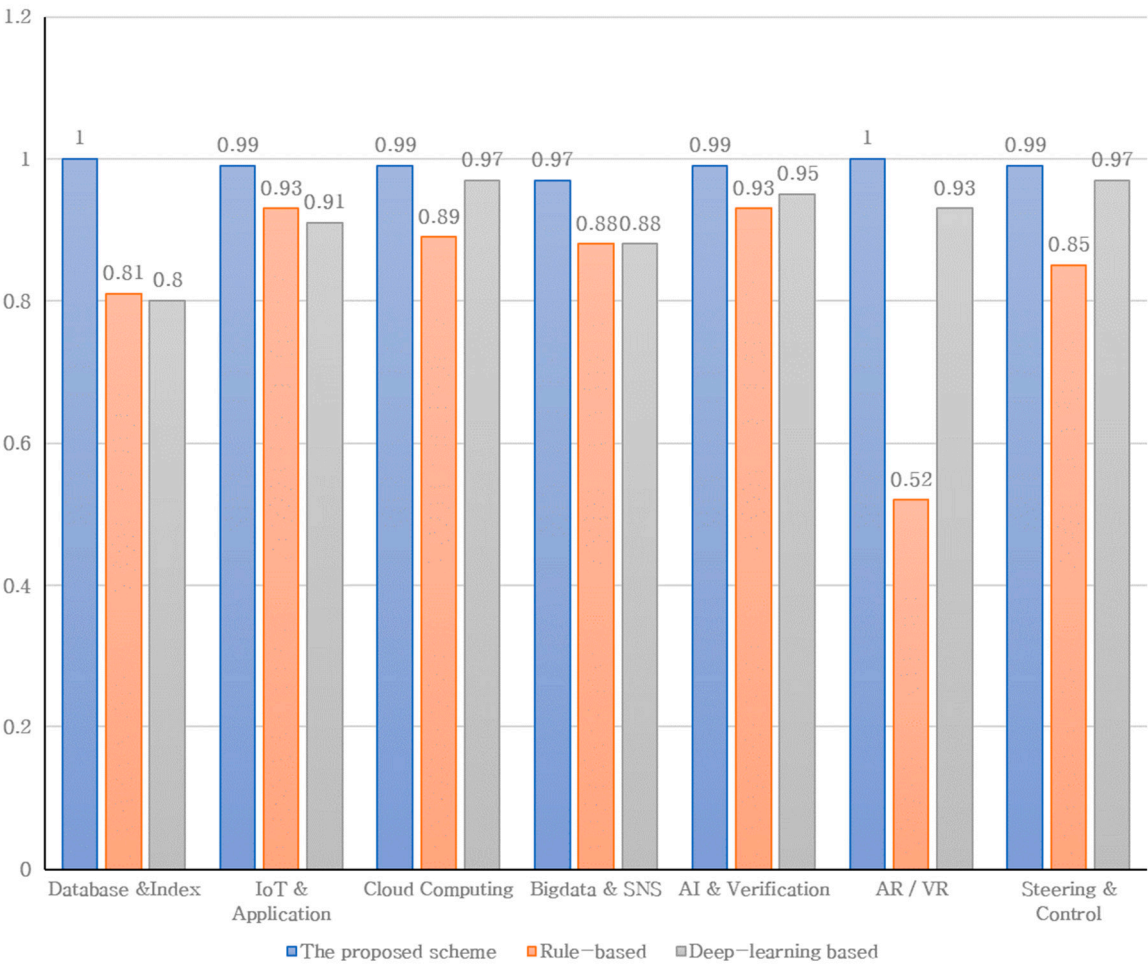


Figure 11. Recall Based on the Name Disambiguation Schemes.

Figure 12 displays the results of the comparative performance evaluation of F1-measure based on the name disambiguation schemes. The F1-measure of the proposed scheme demonstrated a very high performance, with scores >0.98 for all keywords. Compared with the existing rule-based Protasiewicz scheme and deep-learning-based Chen Ya scheme, the proposed scheme exhibited higher performance across all keywords, thereby proving its superiority.

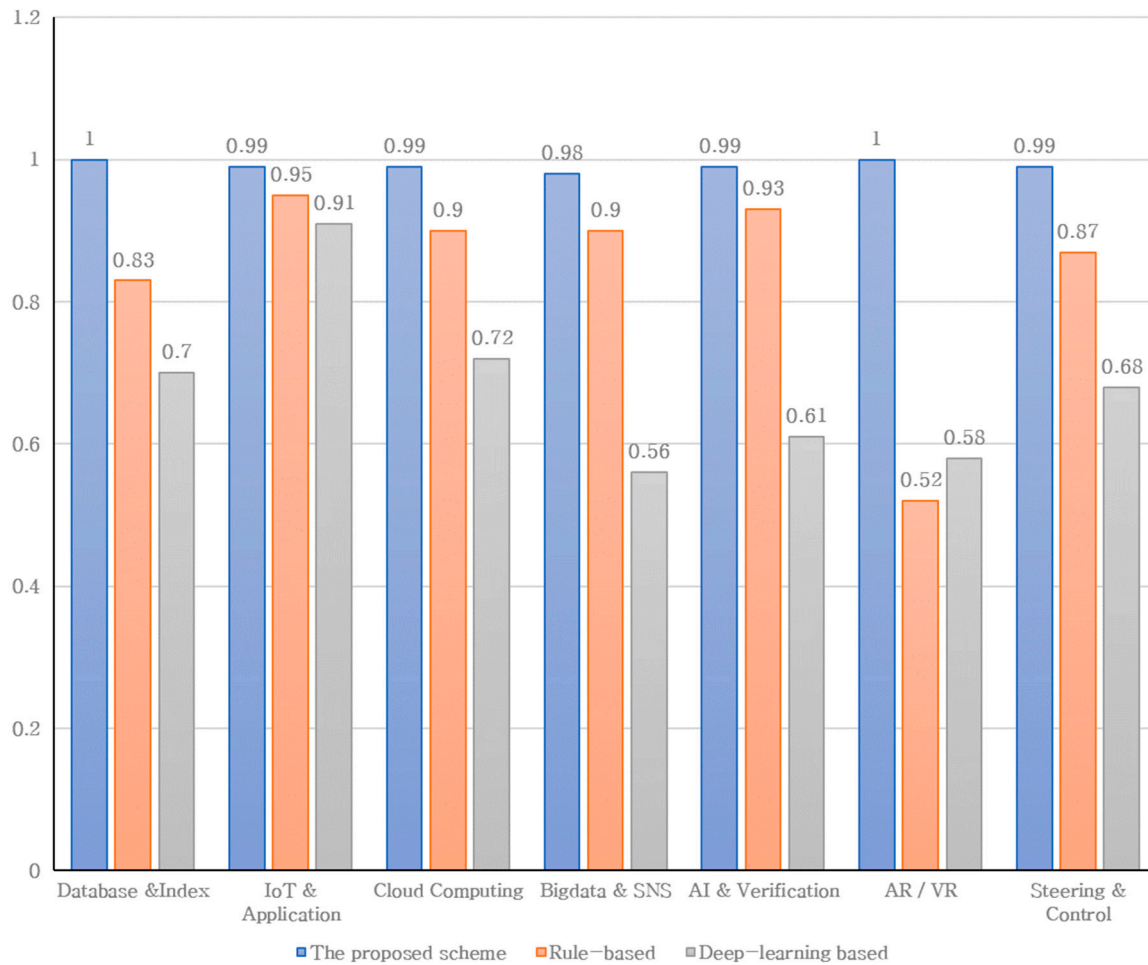


Figure 12. F1-measure Based on the Name Disambiguation Schemes.

4.4. Multiclass Classification Performance Evaluation

In the performance evaluation of the multiclass classification, the proposed scheme selected either the rule-based or deep-learning discrimination method based on attributes using machine learning. The machine-learning models used in the multiclass classification include a total of four classification schemes: support vector classification (SVC), linear SVC, random forest, and naive Bayes. Through the multiclass classification performance evaluation, the most appropriate multiclassification scheme was determined. To measure the accuracy of the proposed method, performance was evaluated by calculating precision, recall, and F1-measure.

For the input of the multiclass classification, two methods were compared: one that represents attributes in a binary form (1 and 0) and another that embeds attributes into vector values using word2vec(W2V) for each attribute. The first method transforms values based on the presence or absence of an attribute. If the attribute is present, it is represented as one, and if absent, it is represented as zero for classifier training. Figure 13 depicts the results of the multiclass classification performance evaluation based on binary attribute embedding. All four classification schemes displayed similar precision; however, the F1-measure of SVC and random forest showed values >0.7 , indicating approximately 8% higher performance than linear SVC and naive Bayes schemes.

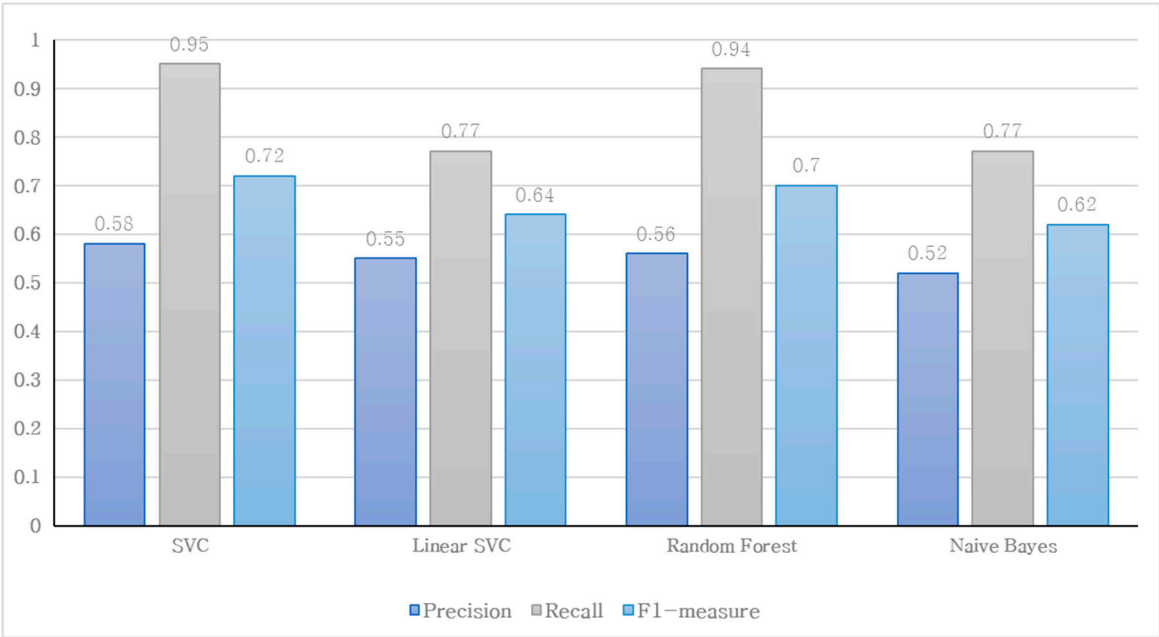


Figure 13. Comparison of Multiclass Classification Performance Based on Binary Attribute Embedding.

The second method involves evaluating the performance of a multi-classifier based on W2V, one of the most renowned feature embedding schemes. Figure 14 shows the results of the multiclass classification performance evaluation based on W2V. The performance evaluation results showed that the random forest scheme exhibited outstanding precision and an F1-measure value of 0.98. This was 28% higher than the results of the random forest scheme embedded using 1s and 0s. Additionally, SVC also displayed an F1-measure value of 0.98, which was 23% better than that of the previous method. In conclusion, for multiclass classification, use of values transformed through W2V for training is more suitable rather than relying solely on the presence or absence of attributes.

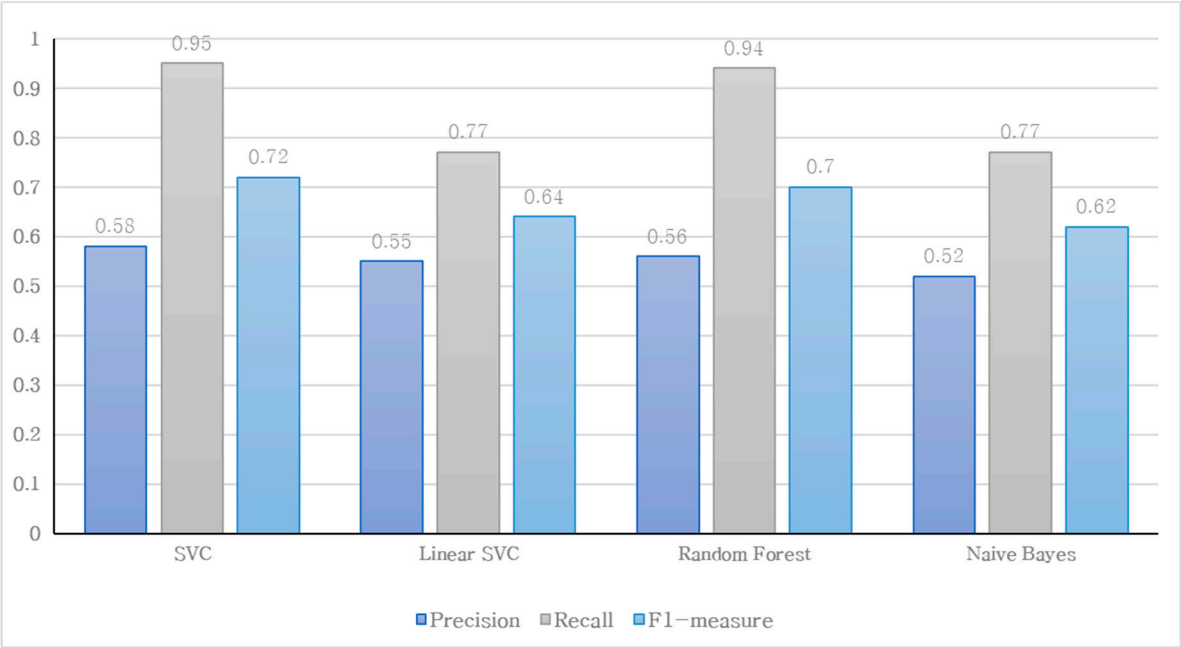


Figure 14. Comparison of Multiclass Classification Performance Based on W2V.

5. Conclusions

In this paper, we proposed a name disambiguation scheme based on heterogeneous academic search sites. The proposed scheme integrated and collected research outcomes provided by heterogeneous academic search sites. Using the collected data, name disambiguation was performed using clustering schemes based on necessary attributes. Moreover, the proposed method was compared with and evaluated against traditional rule-based name disambiguation schemes and deep-learning-based name disambiguation schemes. Considering the metadata provided by academic search sites, we proposed a multiclass classification capable of selecting a more accurate name disambiguation scheme. The proposed multi-classifier selects a more precise name disambiguation scheme. The performance evaluation of the proposed method showed an exceptionally high F1-measure value of 0.99, confirming its suitability as the most apt scheme for name disambiguation. In future, we plan to expand the proposed method to a multi-language-based name disambiguation scheme.

Author Contributions: Conceptualization, D.C., J.J., S.S., H.L., J.L., K.B., and J.Y.; methodology, D.C., J.J., S.S., H.L., J.L., K.B., and J.Y.; validation, D.C., J.J., H.L., J.L., and K.B.; formal analysis, D.C., J.J., S.S., H.L., J.L., and K.B.; writing—original draft preparation, D.C., J.J., S.S., and K.B.; writing—review and editing, J.Y.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT). (No. 2022R1A2B5B02002456, No. RS-2022-00166906), the Korea Association of University, Research Institute and Industry (AURI) grant funded by the Korean Government (Ministry of SMEs and Startups; MSS). (No. S3047889, HRD program for 2021), and by “the Ministry of Science and ICT (MSIT), Korea, under the Grand Information Technology Research Center support program (IITP-2023-2020-0-01462) supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Reference

1. Smalheiser, N. R. ; Torvik, V. I. Author Name Disambiguation. *Annual review of information science and technology* **2009**, 43(1), pp. 1–43.
2. Bhattacharya, I. ; Getoor, L. Collective Entity Resolution in Relational Data. *ACM Transactions on Knowledge Discovery from Data* **2007**, 1(1), pp. 1–36.
3. Ferreira, A. A.; Gonçalves, M. A.; Laender, A. H. A Brief Survey of Automatic Methods for Author Name Disambiguation. *Acm Sigmod Record* **2012**, 41(2), pp. 15–26.
4. Levin, M.; Krawczyk, S.; Bethard, S.; Jurafsky, D. Citation-based Bootstrapping for Large-Scale Author Disambiguation. *Journal of the American Society for Information Science and Technology*, **2012**, 63(5), pp. 1030–1047.
5. Louppe, G.; Al-Natsheh, H.T.; Susik, M.; Maguire, E.J. Ethnicity Sensitive Author Disambiguation using Semi-Supervised Learning. In *Knowledge Engineering and Semantic Web: 7th International Conference, Prague, Czech Republic, 21-23 September, 2016*.
6. Veloso, A.; Ferreira, A. A.; Gonçalves, M. A.; Laender, A. H.; Meira, W. Cost-Effective On-Demand Associative Author Name Disambiguation. *Information Processing & Management* **2012**, 48(4), pp. 680–697.
7. Protasiewicz, J.; Dadas, S. A Hybrid Knowledge-Based Framework for Author Name Disambiguation. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9-12 October 2016.
8. Hermansson, L.; Kerola, T.; Johansson, F.; Jethava, V.; Dubhashi, D.P. Entity Disambiguation in Anonymized Graphs using Graph Kernels. In: *22nd ACM International Conference on Information and Knowledge Management, CIKM 2013, San Francisco, CA, USA, 27 October - 01 November 2013*.
9. Zhang, B.; Hasan, M.A. Name Disambiguation in Anonymized Graphs using Network Embedding. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 06–10 November 2017*.
10. Zhang, Y.; Zhang, F.; Yao, P.; Tang, J. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in The Loop. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, 19–23 August 2018*.

11. Qiao, Z.; Du, Y.; Fu, Y.; Wang, P.; Zhou, Y. Unsupervised Author Disambiguation using Heterogeneous Graph Convolutional Network Embedding. 2019 IEEE international conference on big data (Big Data). Los Angeles, CA, USA, 09-12 December 2019.
12. Yan, H.; Peng, H.; Li, C.; Li, J.; Wang, L. Bibliographic Name Disambiguation with Graph Convolutional Network. *Web Information Systems Engineering* **2019**, 11881, pp. 538– 551.
13. Chen, Y.; Yuan, H.; Liu, T.; Ding, N. Name Disambiguation Based on Graph Convolutional Network. *Scientific Programming* **2021**, pp. 1-11.
14. Ma, C.; Xia, H. Author Name Disambiguation Based on Heterogeneous Graph. *Journal of Computers* **2023**, 34(4), pp. 41-52.
15. Rettig, L.; Baumann, K.; Sigloch, S.; Cudré-Mauroux, P. Leveraging Knowledge Graph Embeddings to Disambiguate Author Names in Scientific Data. In 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17-20 December 2022.
16. Protasiewicz, J. A Support System for Selection of Reviewers. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 05-08 October 2014.
17. Li, J.; Shao, H.; Sun, D.; Wang, R.; Yan, Y.; Li, J.; Abdelzaher, T. Unsupervised Belief Representation Learning with Information-Theoretic Variational Graph Auto-Encoders. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11-15 July 2022.
18. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *Advances in neural information processing systems*, Barcelona, Spain, 5-10 December 2016.
19. Dreßler, K.; Ngonga Ngomo, A. C. On The Efficient Execution of Bounded Jaro-Winkler Distances. *Semantic Web* **2017**, 8(2), pp. 185-196.
20. sim2diss. Available online : <https://rdrr.io/cran/smacof/src/R/sim2diss.R> (accessed on 02/11/2023)
21. SNAP. Available online : <https://snap.stanford.edu/data/> (accessed on 11/12/2022).
22. Anaconda. Available online : <https://www.anaconda.com/> (accessed on 11/12/2022).
23. scikit-learn. Available online : <https://scikit-learn.org/stable/> (accessed on 11/12/2022).
24. keras. Available online : <https://keras.io/> (accessed on 11/12/2022).
25. tensorflow. Available online : <https://www.tensorflow.org/> (accessed on 11/12/2022).
26. matplotlib. Available online : <https://matplotlib.org/> (accessed on 11/12/2022).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.