Scientific reference style using rule-based machine learning

Afrida Helen [a,1,*], Aditya Pradana [a,2], Muhammad Afif [a,3]

[a] Computer Science Departement, Padjadjaran University, Sumedang, Indonesia
[1] helen@unpad.ac.id; [2] aditya.pradana@unpad.ac.id; [3] afif_muhammad9910@yahoo.com
* corresponding author

ARTICLE INFO

Article history
Received March 18, 2023
Revised April 28, 2023
Accepted April 29, 2023
Available online November 30, 2023

Keywords
Regular expression
Reference writing style
Scientific paper
Levenshtein distance
Similarity ratio

ABSTRACT

Regular Expressions (RegEx) can be employed as a technique for supervised learning to define and search for specific patterns inside text. This work devised a method that utilizes regular expressions to convert the reference style of academic papers into several styles, dependent on the specific needs of the target publication or conference. Our research aimed to detect distinctive patterns of reference styles using RegEx and compare them with a dataset including various reference styles. We gathered a diverse range of reference format categories, encompassing seven distinct classes, from various sources such as academic papers, journals, conference proceedings, and books. Our approach involves employing RegEx to convert one referencing format to another based on the user's specific preferences. The proposed model demonstrated an accuracy of 57.26% for book references and 57.56% for journal references. We used the similarity ratio and Levenshtein distance to evaluate the dataset's performance. The model achieved a 97.8% similarity ratio with a Levenshtein distance of 2. Notably, the APA style for journal references yielded the best results. However, the effectiveness of the extraction function varies depending on the reference style. For APA style, the model showed a 99.97% similarity ratio with a Levenshtein distance of 1. Overall, our proposed model outperforms baseline machine learning models in this task. This study introduces an automated program that utilizes regular expressions to modify academic reference formats. This will enhance the efficiency, precision, and adaptability of academic publishing.

1. Introduction

In computer science, rule-based machine learning (RBML) refers to any machine learning method that identifies, learns, manipulate, or apply some data [1]. A rule-based is distinguished by the identification and application a set of relational rules that collectively represent the knowledge captured by the system [2], [3]. A specific Rule-Based is Regular Expression method (RegEx). RegEx is a learning method that can identify a pattern of string in a sentence [4], [5]. This study proposed the regular expressions to convert scientific reference style to another style, since regular expressions provide an automated method for processing a collection of string. The regular expressions method is the method that use symbols or other special characters to indicate patterns in sentences [6], [7]. Scientific articles use several different reference styles. Some are quite similar, while others are vastly different. Each proceeding or journal is given its own reference format style. Scientific reference style usually used including the Modern Language Association, also known as MLA, the Institute of Electrical and Electronics Engineers, also known as IEEE, the Harvard format, the American Psychological Association, also known as APA, the Springer reference format style, the Springer Lecture Notes in

Computer Science reference format style, and the Chicago format style. When authors compose journal or proceeding articles, they have to adjust the reference style based on the journal or proceeding requirements. Occasionally, the author already has a table of references, but the pattern is not appropriate for the intended use. As a result, author must convert the style. They have to a new style if they want to submit an article to a new propose. It is hardly task, so needed a help converting them automatically. The scientific reference sentence is not a complete sentence but it is a phrase, made up of words organized by syllables. Meanwhile, regex has the ability to extract the syllable in a sentence or phrase. A RegEx consist of symbols like characters, alphabet, punctuation and soon. These symbols can be used to match the syllables in scientific reference phrase and then restructure the reference to another style [8]. In some research RegEx is used to translate natural language text into meaning-representative regular expression. The natural language representation and regular expression representation differ in abstraction, making this a fresh and difficult topic. However, a regular expression can be written in many semantically similar forms, which use to facilitate translation by identifying a form that more closely matches the natural language [9].

In this paper we proposed RegEx method to reconstruct the pattern of scientific reference style writing to another style. First step is training process, we collect some scientific article style from various source. Then learn and classify into seven classes. When the testing process, the system recognize the style of the input, then convert them to the style target. It is expected that research conducted using the aforementioned strategy will yield solutions to current problems. The goal of this research is to incorporate regular expressions into software that can be used to manage writing forms, make it easier, classify, and convert references.

2. Method

The first step in developing this research is to conduct a literature review. A comparison of various studies on the subject of reference classification and reference information extraction is performed at this point. Dominika Tkaczyk's Citation Style Classifier's research goal is to assign a given reference to one of 17 reference formats or the format label "unknown". This supervised machine learning classifier employs a simple logistic regression model with TF-IDF feature representation. They generated the training and testing data using Crossref metadata. On the test set, the classifier's accuracy was determined to be 94.7% [10].

The scientific research references dataset can be imported from unstructured dataset [11]. This proposal developed a web application that streamlines and partially automates the editing and formatting of references. They tried two methods to develop this research and use both. It as a rules-based and a machine-learning approach. They used Hidden Markov model (HMM), conditional random fields (CRF), and support vector machine (SVM) classifier. Scientific reference problem can be solved using metadata presenting hybrid rule based and Naive Bayes model. The Naive Bayes algorithm decrease the amount of classification rules and generate several rules for each attribute value. At the end, present an approach for the extraction and visualization of scientific metadata [12]. The research approach is a formally defined writing pattern. Regular expressions are used to define the rules for writing references, which are then extracted using a finite state machine [13], [14]. This method was successful in extracting automatic metadata from scientific papers. The extracted metadata is created, analyzed, and visualized using graphs and treemaps [15].

ParsCit, an open-source CRF reference string parsing package, can be used to extract string in reference sentences. This research reads a free reference string in a proceeding using ParsCit. ParsCit is built around the conditional random field (CRF) model. It is used to indicate the sequence of tokens in a reference string. The heuristic model extends the CRF model by allowing it to find reference strings in plain text files and extract context from citations [16].

Nowday, deep learning is becoming popular for processing text and image. Researcher proposed "Neural ParsCit" that is a reference based on deep learning. The deep learning method in ParsCit is

proposed by AnimeshP et al. The architecture of the ParsCit Neural Model based on the Long Short-Term Memory (LSTM) neural network and the conditional random field (CRF). The evaluation of Neural ParsCit showed that CRF-based parsing got a lot better (p = 0.01) [17]. Beyond that there is research that utilizes regular expression for packet classification solutions. How to detect application-layer protocols for monitoring, security, and network administration. Real-time traffic must be identified per-packet, hence traffic flow data and statistical approaches cannot be used. Clear text data used payload-based signature matching, however manual regular expression signature development is inefficient and humans can miss critical repeating patterns in unknown traffic. This research uses sequential pattern data mining with the apriori algorithm, frequency distribution tables from natural language processing, and pairwise sequence alignment from the Needleman-Wunsch scoring. This algorithm aimed to automatically generate regular expression signatures for targeted protocols [17]. Later, in software engineers research, reguler expression is used for many tasks [18]–[22]. Regular expression can be complex, making them hard for developers to write and understand. Regular expression can seeks comprehension-affecting code smells. To evaluate the understandability of various regular expression language features used golden standart [23]. In the biomedical domain normalization is considered more difficult than concept recognition, e.g. protein and gene recognition in scientific literature, and drugs, diseases, and treatments in electronic patient records, whereas normalization challenges are few. In the gene normalization task, the system must be able to identify all genes mentioned in a free-text article in a given organism. The identification of unique genes is done at the document level, not for individual gene mentions. In this case a wide variety of methods, including pattern matching [8], [24], machine learning [25]. and lexical resource search. Heuristic rules are mostly developed and implemented in an ad-hoc and customized manner. Hybrid use of rule-based and machine learning methods was observed in the system description [26], [27].

Fig. 1 shows the steps of this research. The first step, we collected scientific reference writing style from various source. Then, we annotated depend on their writing style. We found eight classes that had significant differences in writing style, shown in Table 1, and built a meta data using regEx. The metadata comes to be dataset. In the next step, dataset is trained using machine learning method

2.1. Data Collection

The first stage, data collection, this research collected reference writing style of guidebook, scientific articles from internet. Each reference style is learned to make the patterns in regular expressions method. The following sources are used as a reference.

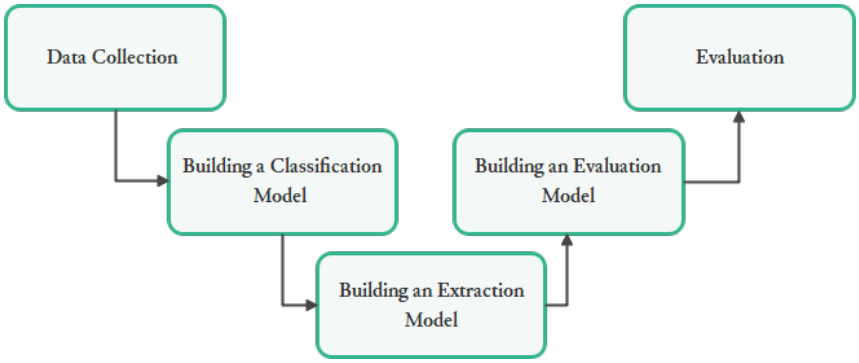

Fig. 1. Sequence of Reference Format Style Stages

The pattern will be used to recognize other reference style, then collect them as datasets. The reference style classifier dataset proposed by Dominika Tkaczyk's Gitlab repository [28] was taken to classify our dataset. The following steps kept the dataset to the repository. Our research had 5,000 documents and took from the collection on the Crossref website.

The dataset consists of 17 references formats style. Some of source format style shows at Table 1. Concerning 5,000 unknown strings were also added. This string is made by switching around words from the original reference string in a random way. Journal proceeding are where most of the references

the dataset come from, but references in books come from places other than the books themselves. Reference style from books take from books that have already been written. In addition, we add 35 record reference style dataset manually including their classes. In this research, we are not too busy with preprocessing the data that has been collected, because our data is primary data and is well-structured, and almost all of the data is already in a condition according to its format.

Table 1. Source of References Format Style

Reference Style Type	Format Source
APA	*Publication Manual of the American Psychological Association, 7th ed.*
IEEE	*IEEE Reference Guide*
MLA	*MLA Handbook Ninth Edition*
Harvard	*The University of Adelaide: Harvard Referencing Guide*
Chicago (*Author -Date*)	*The Chicago Manual of Style*
Springer Basic	*Manuscript Guidelines*
Springer Lecture Notes on Computer Science	*Manuscript Guidelines* dan *Instructions for Authors of Papers to be Published in Springer Computer Science Proceedings*

2.2. Building a Classification Model

This stage created a classification model based on the unique pattern found in each reference and build another classification model, then compare the accuracy of the two models.We detected the unique pattern and read the pattern using regular expression method, and then compared to a references style in collected dataset. The general pattern of reference style consists of the name of the author and the year of publication. They are separated by periods and spaces, and the year of publication is written in parenthesis. Table 2 shows how the different parts of the APA format and other styles fit together.

Table 2. The pattern of special features in reference formats

Reference Style Type	Pattern
APA (Book)	<Author>._(<Year>)
APA (Journal)	<Author>._(<Year>)
IEEE (Book)	[<Reference Number>] &
IEEE (Journal)	<City>:_<Publisher>
MLA (Book)	[<Source Number>] &
MLA (Journal)	“<Judul>,” <Jurnal>,
Harvard (Book)	<Publisher>,_<Year>
Harvard (Journal)	<Jurnal|Volume|Issue>,_<Year>,
Chicago: *Author -Date* (Book)	<Author>_<Year>,
Chicago: *Author -Date* (Journal)	<Author>_<Year>,
Springer (Book)	<Author>._<Year>.
Springer (Journal)	<Author>._<Year>.
Spinger *Lecture Notes in Computer Science* (Book)	<Author>_(<Year>)
Spinger *Lecture Notes in Computer Science* (Journal)	<Author>_(<Year>)

2.3. Extraction Model

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

The pattern of the reference style in dataset is compiled according to the reference format that has known. The pattern of metadata will be used to parse the metadata value. Parsing aims to find the

metadata values in a reference and extract the metadata value into string pieces. The pieces of string will be inserted into a list, array, or Python dictionary. Metadata for book references show in Table 3.

Table 3. Metadata for book references style

Reference Style Type	Metadata Format
APA	Author._(Year)._Title._Publisher
IEEE	[ReferenceNumber]_Author,_Title,_Edition,_Vol._volume._City:_Publisher,_Year, _p/pp._Page
MLA	Author._Title._Edition,_Publisher,_Year
Harvard	Author_Year,_Title,_ Edition. Publisher:_City.
Chicago (*Author -Date*)	Author._ Year._Title._City: Publisher.
Springer	Author_(Year)_Title._Publisher,_City
Springer *Lecture Notes in Computer Science*	ReferenceNumber._Author:_Title._Publisher,_City_(Year)

To perform parsing, the format of the reference is transformed into a regular expression pattern. The pattern for APA, IEEE, MLA, Harvard, Chicago, Springer, and Springer Lecture Notes in Computer Science from book sources are listed in Table 3. Meanwhile, the reference pattern of journal sources can be seen in Table 4.

Table 4. Metadata for journal references style

Reference Style Type	Metadata Format
APA	Author._(Year)._Title._Publisher
IEEE	[ReferenceNumber]_Author,_Title,_Edition,_Vol._volume._City:_Publisher,_Year,_p/pp._Page
MLA	Author._Title._Edition,_Publisher,_Year
Harvard	Author_Year,_Title,_ Edition. Publisher:_City.
Chicago (*Author - Date*)	Author._ Year._Title._City: Publisher.
Springer	Author_(Year)_Title._Publisher,_City
Springer *Lecture Notes in Computer Science*	ReferenceNumber._Author:_Title._Publisher,_City_(Year)

As described in Table 3 and Table 4, the style of the reference is divided into metadata and then written as a regular expression pattern, so that it may be matched against the metadata values in the string fragments. The following is an example of a regular expression pattern from an IEEE reference: [29]. W.-K. Chen, Linear Networks and Systems. Table 5 displays the publication information for Belmont. This method will yield a match object that will be inserted into the Python dictionary data type.

Table 5. Regular expression of each metadata attribute

Metadata Attribute	Regular Expression	Example
Source Number	^\[(\d+)\]	[10]
Author's Name	(.+?)\,\s	W.-K. Chen,
Title	(.+?)\.\s	Linear Networks and Systems.
City of Publisher	(.+?)	Belmont, CA
Publisher	(.+?)	Wadsworth
City: Publisher	(.+?)\:\s(.+?)	Belmont, CA: Wadsworth,
Year	(\d+)\,\s	1993,
Pages	pp\.\s(\d+\-\d+)\.\s	pp. 123-135

The comparison model extraction algorithm is Bayesian. Bayesian has a good track record for probability-based classification. Bayesian is often used for text-based datasets. The first stage determines the feature classes that contribute to the classification. Each reference is parsed into 5 feature classes namely name, year, title, location/publisher, paper page. The labelling of feature classes on the training data is done manually. An example of feature parsing on the following references;

N. Veljkovic, D. Puflović and L. V. Stoimenov, "Scientific References Import from Unstructured Data," Facta Universitatis Series Automatic Control and Robotics, pp. 031-040, 2019. The parsing result is as follows:

Author : N. Veljkovic, D. Puflović and L. V. Stoimenov.

Year : 2019

Title : "Scientific References Import from Unstructured Data,"

Publisher : Facta Universitatis Series Automatic Control and Robotics.

p_p : 031-040

We combine edition and city information in the publisher feature because most of them become one in the writing. The amount of training data for each feature class is almost the same at 1526 data except paper page is less because not all references mention the paper page.

2.4. Building an Evaluation Model

In this stage, the metadata value is put into a data structure and then compiled and put together with string concatenation operations to make a new reference string that was intended. The match objects turn the metadata values in the regular expression into data structures. A Python list, an array, a Python dictionary, or a Pandas DataFrame can be used as the data structure. In this study, metadata values are kept in a Python dictionary before being turned into Pandas DataFrames. Because of both the Python dictionary and the Pandas DataFrame can hold both metadata values and the metadata attributes followed them. A pandas DataFrame is made from a Python dictionary with attributes and metadata values because metadata values are easier to see in a DataFrame. Then, the metadata values in the data structure are put in order based on the rules for reference writing style for the destination.

2.5. Evaluation

Before the system implemented, it was evaluated first. Three elements of the primary model were evaluated, including the classification model, the extraction model, and the conversion model. A classification evaluation model is performed to assess the classification model accuracy in classifying the reference format. The evaluation is performed by calculating the classification model accuracy on the dataset. Thus, it will be easy to determine if the classification results match the dataset's reference format labels.

The evaluation does not make of each and every piece of data contained in the dataset. Only data with a reference format compatible with APA, IEEE, MLA, Harvard, Chicago, Springer, and Springer Lecture Notes in Computer Science that utilized for this study. Therefore, there are 35,000 references utilized in order to gauge how effectively the classification model operates. In order to determine how many changing need to be made to the extracted metadata values before the metadata is converted, an evaluation of the extraction model is carried out with the goal of gathering this information. The Levenshtein distance and the Levenshtein similarity ratio are calculated for each metadata value that was extracted by the model and compared to the metadata value that was supplied explicitly in order to perform the evaluation. It is anticipated that the assessment will be able to demonstrate the correctness of the regular expressions that were utilized in the parsing of the metadata values thanks to this method.

Two sources provided the information that was used for the analysis the model. Sample data from the dataset used to determine the classification accuracy, model were utilized to assess the extraction model from a journal style reference source. If the dataset has a reference that doesn't result in an error when parsed, a sample is randomly chosen from the top data in the dataset. Additional data obtained

from sources outside the dataset is used to assess the extraction function from books used as references. References from new data as well as evaluation data are copied as new data. The new references will be manually divided based on the metadata attribute. The two original datasets will be put into the extraction procedure, and the separated references will be utilized as validation data.

The goal of evaluating the conversion model is to figure out how similar the reference forms that have been rearranged into reference texts with different formats are. As with the evaluation of the extraction function, the evaluation is done by calculating the distance and the Levenshtein similarity ratio from the reference conversion results to the same reference and in the same format as the conversion format. The process of figuring out, how well the extraction model works, we need two datasets. The first dataset is used as an input because it has the metadata that needs to be changed. The second dataset is the reference data that will be used to check the results of the conversion. In the evaluation of the extraction model, the metadata values of the top five references are used as input data. This is because the other 30 references are the same reference written in a different way. Metadata values from references are split according to metadata attributes.

3. Results and Discussion

In this section we divide the discussion into two parts. The first discusses the regular expression model and the second discusses the Bayesian classification model.

3.1. Regular Expression Classification Model

The analysis and assessment of the classification model, the evaluation of parsing metadata in the extraction function, and the evaluation of the reference conversion model make up the discussion of the research findings. The reference style classification model is assessed using the precision of two classification functions, namely the reference writing format classification function for books and the reference writing format classification function for journals.

In this study, 35,000 reference data points were used for the evaluation, with 5,000 reference data for each type of reference style. The number of accurate classifications divided by the total amount of data is used to calculate accuracy. The dataset does not include specific information about the object types it references. The majority of the dataset's references are taken from journals and proceedings. Table 6 demonstrates that there is not much of a difference in the two accuracy models. The accuracy value produced by the classification model for book references is 57.25%, and the accuracy value produced by the classification function for journal references is only somewhat different, at 57.5%.

Table 6. Accuracy scores for classification functions

Classification functions	Accuracy (%)
Book references	57.25
Journal references	57.56

Based on how well the classification model works, it is clear that it has not yet reached a high level of performance. For each type of reference, an evaluation was done to find out more. Table 7 and Table 8 show the results of figuring out the accuracy values for each reference format.

Table 7. Accuracy scores for reference format that referring to the book

Reference Style Type	Accuracy (%)
APA	85.32
MLA	1.84
Chicago	87.26
Harvard	0.02
IEEE	100
Springer	59.28
Springer Lecture Notes in Computer Science	66.74

Table 8. Accuracy scores for reference format that referring to the journal

Reference Style Type	Accuracy (%)
APA	89.04
MLA	0.26
Chicago	89.04
Harvard	0.52
IEEE	100
Springer	55.78
Springer Lecture Notes in Computer Science	68.30

Table 8 shows that the MLA and Harvard citation styles have the lowest accuracy scores, while the APA, Chicago, and IEEE citation styles have the greatest. Low accuracy in classifying MLA and Harvard-style citations is due to the fact that these styles have evolved over time and are no longer identical to the classification scheme. Because publication date serves as the primary pattern in the classification, the distinction between Table 9 and Table 10 is clearly visible when looking at the years in question.

Table 9. Harvard format differences in dataset and classification function

Harvard style according to dataset format	Harvard style according to classification function format
Lecouvey, G. et al., 2012. Les apports de la réalité virtuelle en neuropsychologie : l'exemple de la mémoire prospective. Revue de neuropsychologie, 4(4), p.267.	Lecouvey, G, Gonneaud, J, Eustache, F & Desgranges, B 2012, 'Les apports de la réalité virtuelle en neuropsychologie : l'exemple de la mémoire prospective', Revue de Neuropsychologie, vol. 4, no. 4, p. 267.
Gifford, S.R., 1920. Recurrent Iritis Associated with Dermatitis Exfoliativa. American Journal of Ophthalmology, 3(6), p.433.	Gifford, SR 1920, 'Recurrent Iritis Associated with Dermatitis Exfoliativa', American Journal of Ophthalmology, vol. 3, no. 6, p. 433.

Table 10. MLA format differences in dataset and classification function

MLA style according to dataset format	MLA style according to classification function format
Lecouvey, Grégory et al. "Les Apports de La Réalité Virtuelle En Neuropsychologie : L'exemple de La Mémoire Prospective." Revue de neuropsychologie 4.4 (2012): 267. Crossref. Web.	Lecouvey, Grégory, et al. "Les Apports de La Réalité Virtuelle En Neuropsychologie : L'exemple de La Mémoire Prospective." Revue de Neuropsychologie, vol. 4, no. 4, 2012, p. 267
Gifford, Sanford R. "Recurrent Iritis Associated with Dermatitis Exfoliativa." American Journal of Ophthalmology 3.6 (1920): 433. Crossref. Web	Gifford, Sanford R. "Recurrent Iritis Associated with Dermatitis Exfoliativa." American Journal of Ophthalmology, vol. 3, no. 6, June 1920, p. 433

An evaluation of the performance of the extraction model was done by calculating the distance and the Levenshtein. To carry out the evaluation, the extracted metadata values and the metadata used for validation were respectively converted into two lists, where each list contains a list of each field from a reference. Then, the distance and Levenshtein will be calculated from the element values in the list inside. Since the Levenshtein distance represents the number of edits that must be made to the string to turn it into a string, each Levenshtein for one reference in the list is summed. For the Levenshtein, each Levenshtein for a single reference in the list is averaged. The results of calculating the distance and Levenshtein for 10 references that reference journals with the highest similarity ratio can be seen in Table 11, while the results of calculating the distance and Levenshtein for 10 references that reference books can be seen in Table 12.

Table 11. Levenshtein distance and ratio of similarity for extraction function for journal source

Rank	Reference Style Type	Levenshtein Distance	Similarity Ratio
1	APA	2	0.978
2	APA	12	0.978
3	IEEE	15	0.941
4	Harvard	8	0.922
5	Harvard	19	0.91
6	APA	9	0.867
7	IEEE	38	0.865
8	Harvard	10	0.862
9	IEEE	26	0.861
10	Springer Lecture Notes in Computer Science	7	0.856

Table 12. Levenshtein distance and ratio of similarity for extraction function for book source

Rank	Reference Style Type	Levenshtein Distance	Similarity Ratio
1	APA	69	0.757
2	APA	131	0.753
3	APA	113	0.752
4	Harvard (Book)	110	0.752
5	Springer	132	0.75
6	Springer	111	0.746
7	MLA (Book)	158	0.745
8	MLA (journal)	76	0.737
9	MLA	126	0.735
10	Harvard	164	0.729

Evaluation of the conversion model are being carried out in the same way as the evaluation of the extraction model, specifically by calculating the Lavenstein distance and the similarity ratio. The converted reference text will be stored in a list, Levenshtein distance and similarity ratio will be calculated from string in the list against validation dataset. The results of calculating the Levenshtein distance and similarity ratio for 10 references that referenced journals with the highest similarity ratio can be seen in Table 13.

Table 13. Levenshtein distance and ratio of similarity for conversion function for journal source

Rank	Reference Style Type	Levensthein Distance	Similarity Ratio
1	Springer Lecture Notes in Computer Science	2	0.992
2	Springer Lecture Notes in Computer Science	4	0.987
3	Springer Lecture Notes in Computer Science	3	0.983
4	Springer Lecture Notes in Computer Science	6	0.961
5	Springer Lecture Notes in Computer Science	14	0.956
6	Springer	14	0.954
7	Harvard	14	0.939
8	Harvard	11	0.933
9	IEEE	20	0.932
10	IEEE	22	0.932

The results of calculating the Levenshtein distance and similarity ratio for 10 references that reference books can be seen in Table 14.

Table 14. Levenshtein distance and ratio of similarity for conversion function for book source

Rank	Reference Style Type	Levensthein Distance	Similarity Ratio
1	APA	1	0.997
2	APA	1	0.996
3	APA	2	0.987
4	Chicago	6	0.977
5	Harvard	6	0.976
6	MLA	7	0.973
7	Springer	7	0.97
8	Harvard	6	0.969
9	Harvard	5	0.967
10	Springer Lecture Notes in Computer Science	8	0.964

3.2. Bayesian classification model

This stage builds two classification models. The first model is to recognize the features that distinguish the writing styles of the seven proposed references, the prediction results in the form of Confusion Matrix are shown in Fig. 2. Our next research step entails constructing a classification model to identify and categorize established reference style forms. Six diverse samples of reference writing styles were chosen for this purpose. The construction of this categorization model adheres to a stepwise methodology: Initially, we ascertain the essential attributes required for the categorization process. Subsequently, every citation is analyzed based on these characteristics categories and assigned distinct identifiers such as name, year, title, publisher, and paper_page (p_p) through manual labeling. The process of labeling is carried out by hand in order to guarantee precision. Subsequently, the learning process is conducted with WEKA tools, employing the Bayesian method to facilitate the analysis. In order to train the model effectively, a dataset consisting of 1500 items is used, which offers a significant and diverse range of data for robust model training.

	Predicted					
Actual	1306	0	156	66	0	A_name
	0	1390	0	0	138	Year
	161	0	1143	224	0	Title
	173	0	246	1109	0	Pub
	0	183	0	0	873	p_p
	A_name	Year	Title	Pub	p_p	

Fig. 2. The accuracy for each Author_name class is 85.47%, Year = 90.9%, Title = 74.8%, Publisher = 62.2%, and p_p = 82.7%.

The research explores the intricacies of different reference writing formats, emphasizing that although these styles have commonalities, each has distinct qualities. An anomaly is observed in the Reference Springer Lecture Notes in Computer Science. It markedly deviates from standard journal references, indicating its structure does not conform to conventional journal citing norms. In order to comprehend and evaluate these disparities comprehensively, two confusion matrices were generated. Fig. 2 displays a Confusion Matrix that examines five feature classes: name, year, title, publisher (pub), and paper page (pp). This matrix clarifies how the classification model differentiates between individual properties across various referencing methods, offering insights into the algorithm's capacity to identify and categorize each attribute precisely.

	Predicted						
Actual	4085	700	15	100	50	50	APA
	1445	1125	623	203	789	815	MLA
	475	500	3769	87	38	131	Chicago
	657	1009	946	1219	396	773	Harvard
	89	68	311	8	4478	46	IEEE
	577	423	248	395	252	3105	Springer
	APA	MLA	Chicago	Harvard	IEEE	Springer	

Fig. 3. Confusion Matrix classification for six reference classes

Fig. 3 presents a Confusion Matrix illustrating six distinct writing style groups. This matrix is crucial in evaluating the model's accuracy in correctly identifying and categorizing the overall referencing style rather than simply its constituent elements. The accuracy rates for each writing style are highly informative: The accuracy of the APA style is 81.7%, suggesting a significant conformity with the model's parameters. The MLA style, on the other hand, exhibits a considerably lower accuracy rate of 23.7%, indicating that its structure may present difficulties for the model's current setup. The Chicago style demonstrates a comparatively high level of correctness, reaching 77.6%, while the Harvard style achieves an accuracy rate of 24.5%. The IEEE style is the most precisely recognized, boasting a better accuracy rate of 88.1%. The Springer style demonstrates a commendable level of accuracy at 62.1%; however, it is lower than several other styles. The accuracy rates demonstrate the classification model's efficiency in identifying and distinguishing between distinct reference styles and highlight the intricate and varied nature of academic referencing forms. A comprehensive understanding of these nuances is essential for further improving the model and increasing its usefulness in academic and research contexts..

4. Conclusion

On the basis of the several topics discussed in the preceding section and the research design, and evaluation conducted, the following conclusions can be drawn: Using regular expressions, reference conversion can be achieved by searching for specific patterns that are present in the majority of reference texts. During testing, the classification model achieves an accuracy of 57.25 percent for book references and 57.5 percent for journal references. This is because the categorization function's pattern criteria contradict with the dataset's varied MLA and Harvard citation styles. The reference style conversion is evaluated by evaluating the distance between the two relevant functions, the extraction model and the conversion model, and the Levenshtein similarity ratio. For the extraction function, the APA reference format with a journal reference source having a similarity ratio of 0.978 and a Levenshtein distance of two produces the best results. The APA reference format, which employs a book reference source with a similarity ratio of 0.99 and a Levenshtein distance one, produces the most accurate conversion function results. When using the Bayesian algorithm, the training data automatically becomes a bag of words, so that the model can learn each feature class properly. However, this technique will eliminate the relationship between related feature classes and it is difficult to recognize the features back into a complete reference. In addition, the writing standard is generally modified by the author, causing the accuracy value to be not high. Therefore, the accuracy for RegEx algorithm is better because it reads each character while Bayesian uses the conditional probability principle.

Acknowledgment

The authors thanks DRPM Padjadjaran University for supporting research and publication.

Declarations

Author contribution. The first author generates research ideas, models, and evaluates. The second author transferred knowledge to program code and modeled. The 3rd author built the dataset and model wrote the program code.
Funding statement. Research and publication are supported by Padjadjaran University.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The datasets and program codes used in this study are available from the corresponding author on reasonable request

References

[1] G. Carleo *et al.*, "Machine learning and the physical sciences," *Rev. Mod. Phys.*, vol. 91, no. 4, p. 045002, Dec. 2019, doi: 10.1103/RevModPhys.91.045002.

[2] R. Pradhan, "Rule based Approach to convert abbreviation into Phrases," in *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, Oct. 2021, pp. 1–5, doi: 10.1109/ISCON52037.2021.9702404.

[3] M. D. Drovo, M. Chowdhury, S. I. Uday, and A. K. Das, "Named Entity Recognition in Bengali Text Using Merged Hidden Markov Model and Rule Base Approach," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, Jun. 2019, pp. 1–5, doi: 10.1109/ICSCC.2019.8843661.

[4] D. Stammbach and E. Ash, "DocSCAN: Unsupervised Text Classification via Learning from Neighbors," *KONVENS 2022 - Proc. 18th Conf. Nat. Lang. Process.*, no. Konvens, pp. 21–28, 2022, [Online]. Available at: https://aclanthology.org/2022.konvens-1.4.pdf.

[5] V. G, H. R, and J. Hareesh, "Relation Extraction in Clinical Text using NLP Based Regular Expressions," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Jul. 2019, pp. 1278–1282, doi: 10.1109/ICICICT46008.2019.8993274.

[6] Z. Fu and J. Li, "High speed regular expression matching engine with fast pre-processing," *China Commun.*, vol. 16, no. 2, pp. 177–188, Feb. 2019. [Online]. Available at: https://ieeexplore.ieee.org/document/8663561.

[7] H. Liu, A. Gegov, and F. Stahl, "Categorization and Construction of Rule Based Systems," in *Communications in Computer and Information Science*, vol. 459 CCIS, Springer Verlag, 2014, pp. 183–194, doi: 10.1007/978-3-319-11071-4_18.

[8] C. Chapman, P. Wang, and K. T. Stolee, "Exploring regular expression comprehension," in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Oct. 2017, pp. 405–416, doi: 10.1109/ASE.2017.8115653.

[9] M. Uma, V. Sneha, G. Sneha, J. Bhuvana, and B. Bharathi, "Formation of SQL from Natural Language Query using NLP," in *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, Feb. 2019, pp. 1–5, doi: 10.1109/ICCIDS.2019.8862080.

[10] D. Tkaczyk, "What's your (citations') style?," *Crossref*. Accessed Aug. 12, 2021. [Online]. Available at: https://www.crossref.org/blog/whats-your-citations-style/.

[11] N. Veljković, D. Puflovic, and L. Stoimenov, "Scientific References Import from Unstructured Data," *Facta Univ. Ser. Autom. Control Robot.*, vol. 18, no. 1, p. 031, Sep. 2019, doi: 10.22190/FUACR1901031V.

[12] S. Cvetković, M. Stojanović, and M. Stanković, "An Approach for Extraction and Visualization of Scientific Metadata," in *ICT Innovations 2010, Web Proceeding*, 2010, pp. 161–170, [Online]. Available: http://www.ictinnovations.org/htmls/papers/ictinnovations2010_submission_1.pdf.

[13] Y. Xu *et al.*, "Detecting premature departure in online text-based counseling using logic-based pattern matching," *Internet Interv.*, vol. 26, p. 100486, Dec. 2021, doi: 10.1016/j.invent.2021.100486.

[14] S. Arts, B. Cassiman, and J. C. Gomez, "Text matching to measure patent similarity," *Strateg. Manag. J.*, vol. 39, no. 1, pp. 62–84, Jan. 2018, doi: 10.1002/smj.2699.

[15] M. Cui, R. Bai, Z. Lu, X. Li, U. Aickelin, and P. Ge, "Regular Expression Based Medical Text Classification Using Constructive Heuristic Approach," *IEEE Access*, vol. 7, pp. 147892–147904, 2019, doi: 10.1109/ACCESS.2019.2946622.

[16] I. G. Councill, C. Lee Giles, and M. Y. Kan, "ParsCit: An open-source CRF reference string parsing package," in *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, 2008, no. 3, pp. 661–667, [Online]. Available at: https://citeseerx.ist.psu.edu/viewdoc/download;jsessionid.

[17] A. Prasad, M. Kaur, and M.-Y. Kan, "Neural ParsCit: a deep learning-based reference string parser," *Int. J. Digit. Libr.*, vol. 19, no. 4, pp. 323–337, Nov. 2018, doi: 10.1007/s00799-018-0242-1.

[18] M. Kapoor, G. Fuchs, and J. Quance, "RExACtor: Automatic Regular Expression Signature Generation for Stateless Packet Inspection," in *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)*, Nov. 2021, pp. 1–9, doi: 10.1109/NCA53618.2021.9685959.

[19] P. Wang, G. R. Bai, and K. T. Stolee, "Exploring Regular Expression Evolution," in *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Feb. 2019, pp. 502–513, doi: 10.1109/SANER.2019.8667972.

[20] A. A. Jalal, "Text Mining: Design of Interactive Search Engine Based Regular Expressions of Online Automobile Advertisements," *Int. J. Eng. Pedagog.*, vol. 10, no. 3, p. 35, May 2020, doi: 10.3991/ijep.v10i3.12419.

[21] I. Onyenwe, S. Ogbonna, E. Onyedimma, O. Ikechukwu-Onyenwe, and C. Nwafor, "Developing Smart Web-Search using Regex," *Int. J. Nat. Lang. Comput.*, vol. 11, no. 3, pp. 25–30, Jun. 2022, doi: 10.5121/ijnlc.2022.11303.

[22] C. M. Frenz, "Introduction to Searching with Regular Expressions," in *Proceedings of the 2008 Trenton Computer Festival*, 2008, pp. 1–13. [Online]. Available at: https://arxiv.org/abs/0810.1732.

[23] D. Riaño, R. Piñon, G. Molero-Castillo, E. Bárcenas, and A. Velázquez-Mena, "Regular Expressions for Web Advertising Detection Based on an Automatic Sliding Algorithm," *Program. Comput. Softw.*, vol. 46, no. 8, pp. 652–660, Dec. 2020, doi: 10.1134/S0361768820080162.

[24] C. A. Flores, R. L. Figueroa, and J. E. Pezoa, "Active Learning for Biomedical Text Classification Based on Automatically Generated Regular Expressions," *IEEE Access*, vol. 9, pp. 38767–38777, 2021, doi: 10.1109/ACCESS.2021.3064000.

[25] V. Olago, M. Muchengeti, E. Singh, and W. C. Chen, "Identification of Malignancies from Free-Text Histopathology Reports Using a Multi-Model Supervised Machine Learning Approach," *Information*, vol. 11, no. 9, p. 455, Sep. 2020, doi: 10.3390/info11090455.

[26] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, May 2021, doi: 10.1007/s42979-021-00592-x.

[27] H.-S. Shin, D. Turchi, S. He, and A. Tsourdos, "Behavior Monitoring Using Learning Techniques and Regular-Expressions-Based Pattern Matching," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1289–1302, Apr. 2019, doi: 10.1109/TITS.2018.2849266.

[28] Y. Tang, W. Le, X. Chen, Z. Gu, L. Yin, and X. Yi, "Automatic Classification of Matching Rules in Pattern Matching," in *2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC)*, Jul. 2020, pp. 302–306, doi: 10.1109/DSC50466.2020.00053.

[29] C. A. Flores and R. Verschae, "A Generic Semi-Supervised and Active Learning Framework for Biomedical Text Classification," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Jul. 2022, vol. 2022-July, pp. 4445–4448, doi: 10.1109/EMBC48229.2022.9871846.