iJournals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 9 Issue 5 May 2021

A Stem-Based Classification Approach for Identifying Author Specialty

Sara Mohammed[1]; Tarek El-shishtawy[2]; Walaa Medhat[2,3]

Information System Department, Faculty of Computers & Artificial Intelligence, Benha University, Benha, Egypt[1];

Head, Department of Information System[2]; Information System Department, Faculty of Computers & Artificial Intelligence, Benha University, Benha, Egypt[2];

Information Technology and Computer Science, Nile University [3]

sara.ibrahim17@fci.bu.edu.eg[1]; t.shishtawy@bu.edu.eg[2]; WMedhat@nu.edu.eg[3]

DOI: 10.26821/IJSHRE.9.5.2021.9519

ABSTRACT

Researchers and readers of scientific articles face the problem with identifying the articles and scientific research papers categories and hence the difficulty in determining authors' specialty. Many researchers face the problem of selecting a journal that is suitable for publishing his/her scientific research paper. Many experiences assist researchers in choosing the appropriate journal. However, no one addresses the problem of determining the publisher's specialty of the scientific paper according to his / her article. This paper proposes a solution to identify the author's specialty through abstract comparison. Also, it suggests a new method to help choose the appropriate journal. That finds the appropriate journal according to the abstract of the article that is required to be published. A classification model designs to find the correct category of a given article. Accordingly, the author's specialty is determined. The classifier also finds the Scimago journal categories according to the journal's scope. We built the classifier using a vector space model based on a cosine similarity measure. Also, we use M-TF-IDF weight which is a TF IDF, but we have suggested a modified method that helps us with the measurement. After classifying the article category, a second classifier based on the Levenshtein algorithm selects the appropriate journal for publishing an article. Our dataset is divided into three groups: the scopes of journals, the abstract of articles, and the title of the journal and its scope datasets—all datasets in the main category fromthe Scimago website. The proposed measure shows good performance of results.

Keywords: Classification, Vector Space Model, Cosine Similarity, Modified TF-IDF, Levenshtein Edit Distance.

1. INTRODUCTION

Due to the diversity of scientific disciplines, the reader or researcher has a problem determining the publisher's specialty according to his published article, i.e., to any branch of science the article belongs to. There is no specific measure to determine the publisher's specialty just on reading the article. This is because of the convergence of phrases and words in most science branches. Also, the increasing of number of journals are available for researchers to publish their research made a problem in choosing the best journal for publishing. Therefore, the second problem addressed in this paper that faces the author is choosing the appropriate journal to publish the research papers and articles. This is because the process of publishing research papers and articles requires much time and effort to read and select the appropriate journal for publication.

This work focuses on solving the previous problems. Firstly, we propose a text classification technique that automatically classifies authors and

articles according to the similarity with other classified journal categories from the Scimago website. Second, according to journal's similarity to the article abstract to be published, the classifier sorts journals in a given category automatically.

Text classification is an essential and typical task in supervised machine learning (ML). Assigning categories to documents, which can be a web page, library book, media articles, gallery, etc., has many applications like, e.g., spam filtering, email routing, sentiment analysis, etc. [1]. Once transformed text into numbers, in a way that's machine learning algorithms can understand it is called term weight (TF-IDF), the TF-IDF score can be fed to the cosine similarity algorithm [2]. Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction [3].

Therefore, there is a need to build a system to classify articles then identify the specialties of authors. Also, classify journals to select the appropriate journal for publishing papers and scientific research.

The research main objective and contribution are to adopt a text classification method to build a system to identify the article's main category and the specialties of authors. Furthermore, building a classifier for selecting an appropriate journal to publish a scientific paper. By adopting well-known text classification techniques in which the vector space model based on cosine similarity measure and a modified TF-IDF (M-TF-IDF) method to match the article's abstract and vector spaces of the article abstract and journal scope in 24 categories from the Scimago website. The topic of authors' and publishers' classification not previously address. Therefore, we provide a method to make it easier to identify the author's specialty or scientific research. We tackled all 24 categories of the Scimago website, which helped improve the classification results compared to previous works that only tackled three or four categories. The proposed system facilitates selecting the appropriate journal for publishing a scientific research paper in all knowledge branches. Consequently, the classification results became higher and more accurate.

This paper is organized as follows: Section 2 presents the research efforts that have been done for classification algorithms. Section 3 presents the methodology and steps to the proposed classifier. The experimental test is given in section 4. Section 5 presents a discussion of the results. Finally, the paper concludes in section 6.

2. RELATED WORKS

Text classification algorithms are the heart of many software systems that process text data at scale. Email software uses text classification to determine whether incoming mail is sent to the inbox or filtered into the spam folder. Discussion forums use text classification to determine whether comments should be flagged as inappropriate [4].

In [1] Muhammad Habibi and Praise WinarCahyo used the Cosine Similarity algorithm and Support Vector Machine in the classification process using the TF-IDF weighting method for Journal Classification Based on Abstract. This study used abstract journals as a dataset that consisted of only four science classes. This study aimed to create a classification model that can classify journals automatically using the Cosine Similarity algorithm and Support Vector Machine in the classification process using the TF-IDF weighting method. Based on their experimental results, the Support Vector Machine method produced better performance accuracy than the Cosine Similarity method. Cosine Similarity had an average accuracy value of 61% [5].

In [2] Putri YuniRistanti, AjiPrasetyaWibawa, &UtomoPujianto used Vector Space Model Approach to represent terms in spatial dimensions for Cosine Similarity for Title and Abstract of Economic Journal Classification. They used the Cosine Similarity Method to calculate the two documents' similarities. Also, they used TF-IDF weighting. Their goal was to adopt a classification system of journals' articles to select journals for their articles. The average accuracy results were 57,79% [6].

In [3] PiskaDwiNurfadilaa, AjiPrasetyaWibawaa, Ilham Ari ElbaithZaeni a, & Andrew Nafalski used VSM (Vector Space Model) approach and the Cosine Similarity method and TF-IDF weighting for Journal Classification Using Cosine Similarity Method on Title and Abstract with Frequency-Based Stop word Removal. This study focused on the phase of word elimination by adding frequency-

based stop word removal. Their goal was to adopt a classification system of journals based on Title and Abstract with Frequency-Based Stop word Removal. This study resulted in an accuracy value of 64.28% [7].

3. METHODOLOGY

In this work, we address two main problems. The first is finding unknown article category, and hence the author's main category and specialty. The second is determining the best journal for publishing articles and scientific papers. Our methodology begins with collecting datasets from the Scimago website, divided into three groups. The first group is the scope of journals, while the second group is the abstract of the articles, and the third group is the title of the journals and their scopes. These groups of datasets are in the main categories in all branches of knowledge. The first phase is the pre-processing of data, which is the process of cleaning and preparing the text for classification [8].

The second phase is building modified TF-IDF (M-TF-IDF) classifier. The third phase is building a vector space for each dataset. The first vector is the vector space of journal scopes in 24 categories. The second vector is the article abstract's vector space in 24 categories.

The fourth phase applies the cosine similarity method to classify articles and journals. Cosine similarity measurement is the testing phase. The similarity of an unclassified article abstract to the vector space of journal scope is measured, which shows the law accuracy of results. The highest accuracy appeared with the similarity between an unclassified article abstract to the article abstract's vector space. That is to determine the specialty and main category of the author of this paper.

The last phase applies the Levenshtein (edit) distance method for selecting the best journal by comparing the article's abstract to be published with the group of journal titles and their scopes available to us. See Figure 1 it shows the phases of the methodology:

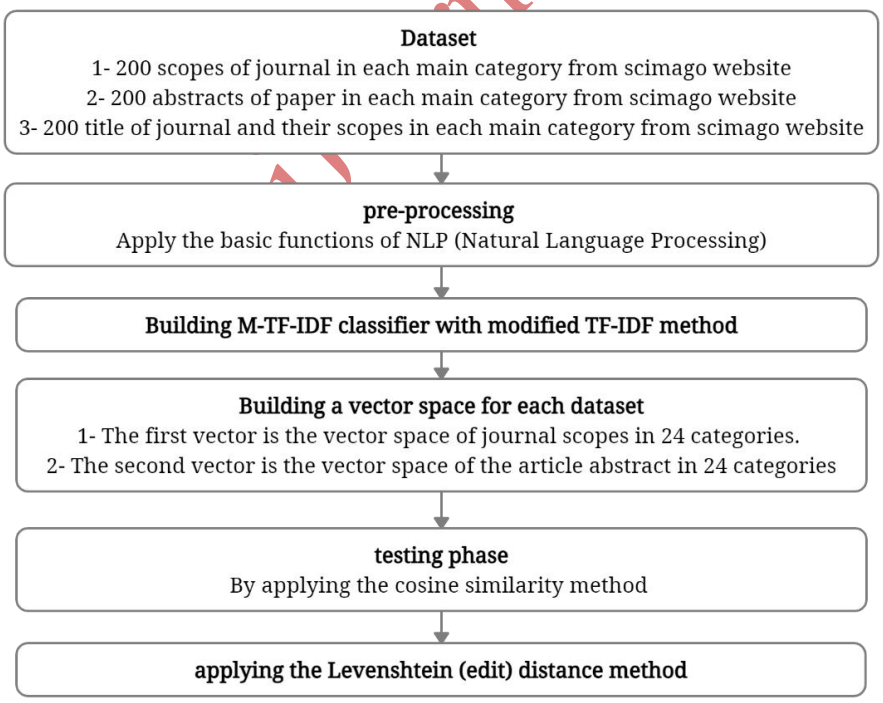

Fig 1: Phases of methodology

3.1 Dataset

The dataset used was extracted from the Scimago[1] website (journal rankings branch). Scimago'swebsite contains enormous data that organize in a structured way. It divides into categories and subcategories on which our system depends. It includes all branches of knowledge (computer science, energy, chemistry, art ..., etc.), all subject categories (computational theory and mathematics, communication, analytical chemistry, …., etc.), and many journals in both category and subcategory branches. All subject areas, which we refer to in our experiment as the main category, contain 24 branches of knowledge.

The datasets were collected in October 2020, consisting of 9600 data containing abstracts and scopes. The datasets divide into three groups. The first is the journals' scope, which includes 4800 journal scopes belonging to 24 different categories. We got the scopes from the journals published on the Scimago website. We applied this method with 24 branches available on the Scimago website. In each category, we took 200 scopes. The second is the abstracts of the article. It contains 4800 article abstracts belonging to 24 different categories. We got the abstracts from the articles published in the journals which is published on Scimago website. We applied this method with 24 branches available on the Scimago website. In each category, we took 200 abstracts. As the journals, which can classify on the Scimago website, may belong to more than one branch, we targeted journals classified in a single category to improve classification and testing in our adopted method. The third is the title of the journal and its scope. We got the titles and scopes from the journals published on the Scimago website. We applied this method with several branches available on the Scimago website. In each category, we took 200 titles and scopes.

3.2 Pre-processing

Pre-processing aims to change previously unstructured documents into structured documents [9]. Pre-processing performed by applying the essential functions of NLP (Natural Language Processing), which are:

a. Remove punctuation and number to improve performance. A case folding, converting all uppercase characters in the document to lowercase, has been implemented at this stage.

b. Tokenization: is the process of transforming a paragraph of text into a token.

c. Remove stop words: It includes eliminating common language articles, pronouns, and prepositions such as “and,” “the” or “to” in the English language.

d. Remove the most frequent words such as verbs and some common and undefined words. The verbs are such as “add,” “read,” or “educate” in the English language. Also, the familiar and undefined words are such as “research,” “article,” or “journal” to improve performance.

e. Stemming: is the process of slicing the ending of or the beginning of the word, using a list of common prefixes and suffixes like (-ing, -ed, -es).

See Figure2 it shows a flowchart of pre-processing phases:

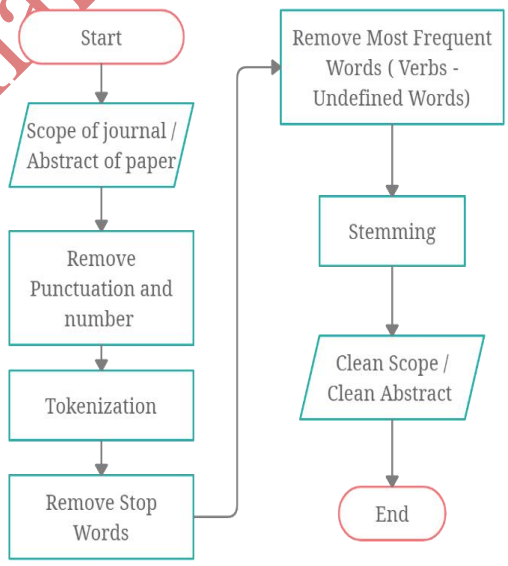

Fig 2: Flowchart of pre-processingThe Used Algorithms

This study uses the Vector Space Model (VSM), an algebraic model for representing text documents as vectors of identifiers [10]. The VSM has proved an effective and sound framework in retrieving documents in different languages, on different subjects, of different sizes, and of different media, thanks to a number of proposed and tested weighting schemes and applications [11]. The VSM approach chose because it allows

[1]Scimago website: https://www.scimagojr.com

measurement of similarity degree between queries and documents and arranges documents according to their potential significance.

One of the VSM-based classification methods is Cosine similarity. Cosine similarity uses widely to classify text. For example, Journal Classification Based on Abstract Using Cosine Similarity and Support Vector Machine [5]. Also, they have used it for Title and Abstract of Economic Journal Classification [6] and Journal Classification on Title and Abstract with Frequency-Based Stop word Removal [7].

In addition to using the cosine similarity algorithm, TF-IDF is used in this study but we proposed it in a modified method as it is broadly used as an essential technique in the text classification process. For example, Text Mining Use of TF-IDF to Examine the Relevance of Words to Documents [12].

Levenshtein (edit) distance is also used in this study to help determine the appropriate journal. The Levenshtein distance is a string metric for determining the difference between two sequences. Levenshtein (edit) distance uses widely. For example, A Hybrid Cross-Language Name Matching Technique Using Novel Modified Levenshtein Distance [13].

3.3 Building Classifier with Modified TF-IDF Method (M-TF-IDF)

At this stage, the document was displayed as a vector using the VSM approach. The function of VSM is to convert documents into numbers so that we can calculate the weight [14]. Weight calculated from each term in the training document and test documents using the TF-IDF weighting method. The TF-IDF stands for term frequency-inverse document frequency, and the TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus [15]. We proposed a modified method to calculate TF-IDF we called it as M-TF-IDF. There were four steps to determine the value of M-TF-IDF weighting, which are:

Step 1: We listed all words in 24 categories and find the frequency of each term (word) in each category, which is considered the term frequency (TF). After this, we find universe frequency (UF). It is the sum of frequencies of one word in all categories and the same way for the rest of the words. It can represent mathematically asfollows in Eq. (1):

$$UF\ (i) = \sum_{cat\ 1}^{cat\ 24} TF(cat, i) \qquad (1)$$

Step 2: We find normalized term frequency (NTF) by dividing term frequency of word on universe frequency of the same word. It can represent mathematically as follows in Eq. (2):

$$NTF = \frac{TF(cat\ ,i)}{UF\ (i)} \qquad (2)$$

Step 3: we find the vector (category) length by calculating the sum of the squares of each NTF of each word in the same category. After this, take the square root of the addition results. It can represent mathematically as follows in Eq. (3):

$$length(cat_1) = \sqrt{NTF(1)^2 + NTF(2)^2 + \cdots + NTF(n)^2} \qquad (3)$$

Step 4: we find a normalized vector (Unit vector) for each category (NV) by dividing normalized term frequency (NTF) by the length of the vector. It represents M-TF-IDF for each vector. So, we will have 24 vectors. It can represent mathematically as follows in Eq. (4):

$$NV(cat_{1\to24}) = \frac{NTF\ (cat\ _{1\to24})}{Lengt\ h\ (cat\ _{1\to24})} \qquad (4)$$

3.4 Finding Article Category and Author Specialty

The first problem is based on building two vector spaces. The first vector is the vector spaces of the journal scope category, while the second vector is the vector spaces of the article abstract. See Figure 3it shows the first problem of finding article category:

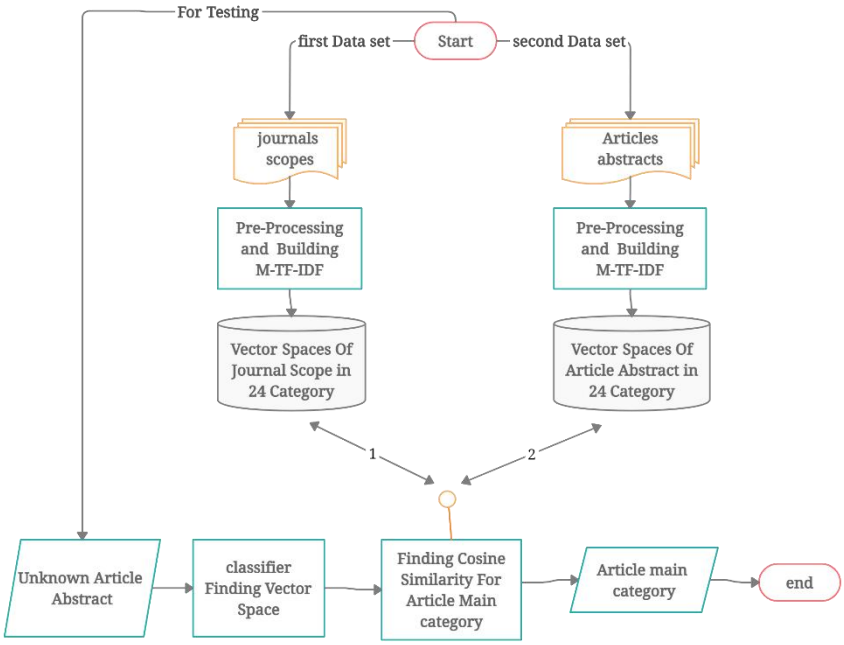

Fig 3: The first problem of finding article category

3.4.1 Building journal scope vector space

We start by collecting 200 scopes of the journals in each category from the Scimago website. The pre-processing function is applied by starting with remove punctuation until executing the stem. After this process, the M-TF-IDF score calculates. So, the vector space of the journal scopes is building.

3.4.2 Building article abstract vector space

We start by collecting 200 abstracts of the article in each category from the Scimago website. Pre-processing is applied by starting with remove punctuation until executing the stem. After this process, the M-TF-IDF score calculates. So, the vector space of the article abstracts is building.

3.5 Test Phase by Applying Cosine Similarity Measurement

After the TF-IDF calculation, the cosine similarity score calculates. Cosine similarity is a measure of similarity that can be used to compare documents or give a ranking of documents with respect to agiven vector of query words. Let x and y be two vectors for comparison. The measure computes the cosine of the angle between vectors x and y. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors [3]. We have represented it mathematically as follows in Eq. (5):

$$\cos(v_{a\,\&\,j}) = NV(article\ or\ journal) * NV(cat_{1\to 24}) \quad (5)$$

*NV(article or journal)*is a normalized (or unit) vector for an article or journal under test, and$NV(cat_{1\to 24})$is a normalized (or unit) vector for each category to determine the article or journal under testing closest to any of these categories. See Figure 4 it shows the proposed algorithm steps:

Input:
Training documents,
Test documents
Output:
The smallest similarity angle of each vector
Steps:
1. Creating index term;
2. Words weighting (TF-IDF)
-Calculating term frequency (TF) of training and testing document;
-Calculating normalized term frequency of training and testing document;
-Calculating the length of the training and testing vector;
-Calculating normalized vector for each category and tested journal;
3. Cosine similarity scores;
4. Calculating the similarity of documents;

Fig 4: Proposed Algorithm steps

3.6 Selecting the Best Journal

The second problem is selecting the best journal. After the authors and articles are classified and the author's specialty determined, we know which journal in this category is most suitable for publication. We apply this by using the Levenshtein edit distance method. The Levenshtein (edit) distance is defined to be the smallest number of edit operations (insertions, deletions, and substitutions) required to change one string into another. The greater the Levenshtein distance, the more different the strings are. By default, the Levenshtein distance can be calculated between words from the same writing script, so in order to compare two names from different scripts, one of them should be initially transliterated to the other script [13]. We have selected the appropriate journal to publish the article and the scientific paper by measuring the similarity between the article's abstract and a group of journal titles and their scopes in the specified branch. The journal is chosen based on the highest likelihood we get. See Figure 5 it shows the process steps of selecting the best journal:

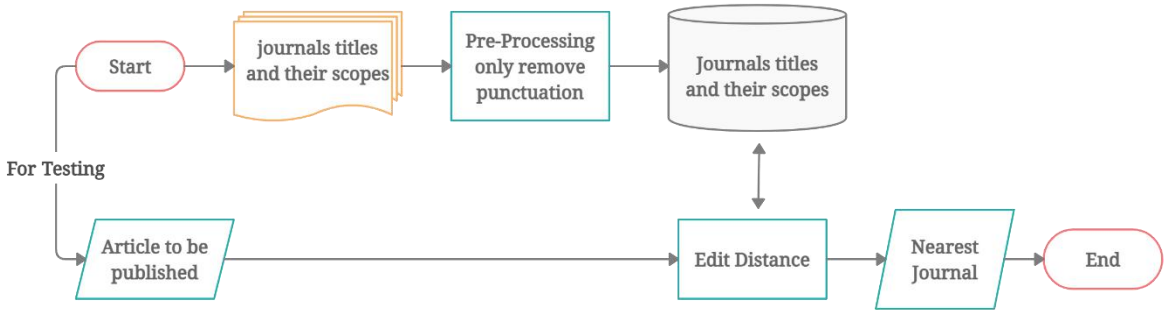

Fig 5: The process steps of selecting the best journal

4. EXPERIMENTAL TEST

For the first problem, which is finding the article category, and hence, the author's specialty, we tested the article abstract on two databases. The first database is the vector spaces of journal scope, while the second is the vector space of the article abstract. For the second problem of selecting the best journal, we test on the last database that is the journal titles and their scopes. We explain the test method as follows:

4.1 Finding Unknown Article Category

A varying number of test cases from 10 to 30 are examined, and the result of each category is the average result. The data is fed to the previous first vector space, which is the journal scope's vector space. The test case is the abstract of a published paper within a journal in an unknown category. We measure the similarity between the article abstract and the journal scope's vector space. This test shows the low accuracy of results. See Table 1 it shows the averages results of accuracy in each case study with different samples:

Table 1. Results by using the vector space of journal scope

	Success percentage			
Cases	Case 1 – 10 samples	Case 2 – 20 samples	Case 3 – 30 samples	***Accuracy***
Average	10%	10%	8%	9%

Therefore, the previous second vector space, the article abstract vector space, is built and used to feed the system. We measure the similarity between the article abstract and the article abstract's vector space. This test shows the high accuracy of results. The test measures are the trained system's success in classifying the article to the correct category. Based on it, the author's specialty is determined. See Table 2 it shows the averages results of accuracy in each case study with different samples:

Table 2. Results by using the vector space of article abstract

	Success percentage			
Cases	Case 1 – 10 samples	Case 2 – 20 samples	Case 3 – 30 samples	***Accuracy***
Average	69%	69%	70%	70%

4.2 Testing the Journal Scope to Show How the Accuracy of Finding Journal Scope

A varying number of test cases from 10 to 30 are examined, and the result of each category is the average result. The data is fed to the previous first vector space, which is the journal scope's vector space. The test case is the scope of a new journal not used in the unknown category's training set. The test measures the trained system's success in classifying the journal to the correct category. Accordingly, the appropriate journal category was chosen. See Table 3 it shows the averages results of accuracy in each case study with different samples:

Table 3. Results of journal classification

	Success percentage			
Category	Case 1 – 10 samples	Case 2 – 20 samples	Case 3 – 30 samples	***Accuracy***
Average	73%	71%	72%	72%

4.3 Selecting the Best Journal

We have examined 30 test cases in the specified category; the result is determined based on the highest likelihood. The data is fed from the previous third trained system, the journal titles, and scopes. The test case is the article abstract in a specified category detected from the test of finding article category. A measure of test success is selecting the best journal. Based on it, the appropriate journal for publishing is determined. See Table 4 it shows the results in a specific category:

Table 4. Results of selecting the best journal in the computer science category

Paper number	*Abstract*	*Journal title*	*Similarity ratio*
1	soft robots promise an exciting design trajectory in the field of	Neuro-Informatics	19%
2	we introduce physics informed neural networks neural networks	Neural Networks Journal	17%
3	we investigated how product attributes average consumer ratings	Systems Control Letters	15%

4.4 Evaluation

This study's evaluations depend on identifying the smallest similarity angle for each class and the highest likelihood we get for choosing an appropriate journal. The similarity is between two vectors: the category vector and the vector of tested article or journal. We repeat the testing process three times. Each time we test several articles and journals. In the second problem of choosing the appropriate journal, we test a specific number of articles. While verifying the results, we produce a test case results table. At the end, we determine accuracy by applying the following equation represented in Eq. (6):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \quad (6)$$

5. RESULTS AND DISCUSSION

In our proposed method, we repeat the testing process three times. These results depend on the cosine similarity method for the classification of articles in all categories of knowledge to classify the articles to the correct category and the classification of journals for determining the correct journal category. Also, the results depend on using Levenshtein (edit) distance for choosing the appropriate journal for publishing the article and a scientific paper.

5.1 Finding unknown article category

Comparing the results using journal scopes vector space and article abstract vector space, the result using article abstracts vector space is better than the result using journal scopes vector space. Whereas the average success while using journal scopes vector space ranges between 0% to 20%. These results led to the test using the second vector space, which is the article abstract's vector space, which showed better results. The test results using the article abstract's vector space were high performance, as the average accuracy was 70%. See Figure 6 it shows the averages articles success in 3 cases:

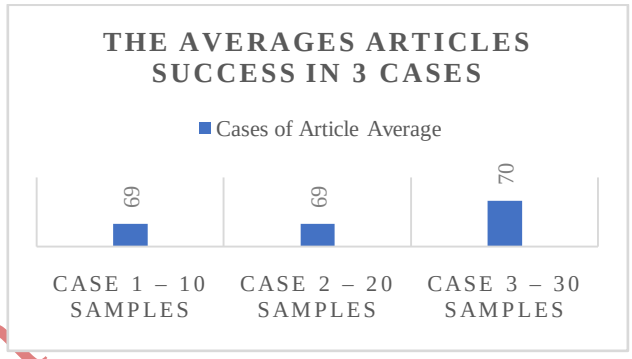

Fig 6: Averages articles success in 3 cases

5.2 Finding journal category

The results of the classification journals determine the correct journal category. These test results showed high performance, as the average accuracy was 72%. The moderate success ranges between 60% to 80% in all test cases. In case 1, The success factor ranges from 60% to 80%. In case 2, the success factor ranges from 50% to 85%. In case 3, the success factor ranges from 65% to 80%. See Figure 7 it shows the averages journals success in 3 cases:

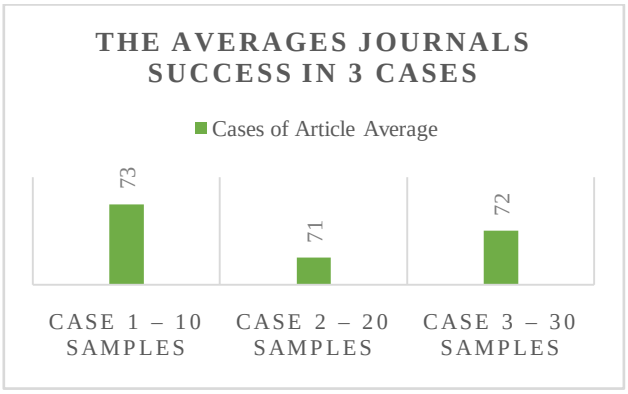

Fig 7: Averages journals success in 3 cases

5.3 Selecting the Best Journal

These results show selecting the best journal for publishing a scientific paper by comparing the article's abstract and a group of journals' titles and scopes. We show three cases from 30 cases in the computer science category, as shown in table 4. In paper 1, after classifying and knowing the category of paper, the appropriate journal for publishing was Neuro-Informatics with a similarity ratio of 19%. In paper 2, after classifying and knowing the category of paper, the appropriate journal for publishing was Neural Networks Journal, with a similarity ratio of 17%. In paper 3, after classifying and knowing the category of paper, the appropriate journal for publishing was Systems Control Letters with a similarity ratio of 15%. The average results range in all cases range between 19% to 10%.

6. CONCLUSIONS

This study uses a cosine similarity algorithm and a modified TF-IDF weight to apply classification for articles and determine author specialty according to his / her publication. The classification was applying to the journals to determining the correct journal category. Articles, authors, and journals are classifying according to the classification of categories on the Scimago website. After applying the article and the scientific research paper classification, there is a need to know the appropriate journal to publish this article or scientific research paper. Therefore Levenshtein (edit) distance method was applying. System performance testing produces a score of accuracy at 70% for classifying articles and authors and 72% for classifying journals, While the results of previous research dealing with the issue of classification of journalswere from 57% to 64%. In the case of applying the Levenshtein (edit) distance method, the similarity ratio results range from 19% to 10%. While the results of the previous application through which we examined the same scientific research papers range between 8 to 16. Accordingly, the uses of the cosine similarity method and Levenshtein (edit) distance method led to better results and tremendous success of the system.

7. REFERENCES

[1] J. Shaikh, “Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK.,” Towards Data Science, 30-Oct-2017.

[2] B. Stecanella, “What is TF-IDF,” MonkeyLearn Blog, 10-May-2019.

[3] J. Han, M. Kamber, and undefined undefined undefined, “Getting to Know Your Data,” in Data mining: concepts and techniques, Third edition., Burlington, MA: Elsevier, 2012, pp. 39–82.

[4] jolasa Iñaki, “Text Classification: Data Science and Machine Learning,” Kaggle, 17-Jul-2019.

[5] M. Habibi and P. W. Cahyo, “Journal Classification Based on Abstract Using Cosine Similarity and Support Vector Machine,” JISKA (Jurnal Informatika Sunan Kalijaga), vol. 4, pp. 185–192, Jan-2020.

[6] P. Y. Ristanti, A. P. Wibawa and U. Pujianto, "Cosine Similarity for Title and Abstract of Economic Journal Classification," 2019 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 2019, pp. 123-127.

[7] P. D. Nurfadila, A. P. Wibawa, I. A. E. Zaeni, and A. Nafalski, “Journal Classification Using Cosine Similarity Method on Title and Abstract with Frequency-Based Stopword Removal,” International Journal of Artificial Intelegence Research, vol. 3, pp. 28–37, Dec-2019.

[8] E. Haddi, X. Liu, and Y. Shi, “The Role of Text Pre-processing in Sentiment Analysis,” Procedia Computer Science, vol. 17, pp. 26–32, 2013.

[9] D. M. Eler, D. Grosa, I. Pola, and R. E. Garcia, “Analysis of Document Pre-Processing Effects in Text and Opinion Mining,” Information, vol. 9, p. 100, Apr-2018.

[10] Chris I, “Let’s Understand the Vector Space Model in Machine Learning by Modelling Cars,” Towords Data Science, 04-Nov-2019.

[11] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, vol. 24, no. 5, pp. 513–523, 19-Jul-2002.

[12] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” International Journal of Computer Applications, vol. 181, no. 1, pp. 25–29, 16-Jul-2018.

[13] D. Medhat, A. Hassan and C. Salama, "A hybrid cross-language name matching

technique using novel modified Levenshtein Distance," 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 2015, pp. 204-209.

[14] V. C. Trejo, G. Sidorov, S. M. Jiménez, and M. Moreno, "Latent Dirichlet Allocation complement in the vector space model for Multi-Label Text Classification," International Journal of Combinatorial Optimization Problems and Informatics, vol. 6, pp. 7–19, Apr-2015.

[15] K. A. R. E. N. S. P. A. R. C. K. JONES, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," Journal of Documentation, vol. 28, no. 1, pp. 11–21, 01-Jan-1972.

