Similarity Visualizer Using Natural Language Processing in Academic Documents of the DSpace in Ecuador

Chapter · March 2023

DOI: 10.1007/978-3-031-28032-0_28

Diego Vallejo
Universitat Politècnica de València
32 PUBLICATIONS **82** CITATIONS

Similarity Visualizer Using Natural Language Processing in Academic Documents of the DSpace in Ecuador

Diego Vallejo-Huanga[1](✉), Janneth Jaime[2], and Carlos Andrade[2]

[1] Universidad Politécnica Salesiana, IDEIAGEOCA Research Group, Quito, Ecuador
dvallejoh@ups.edu.ec

[2] Universidad Politécnica Salesiana, Department of Computer Science, Quito, Ecuador
{ljaimep,candradev5}@est.ups.edu.ec

Abstract. Due to the widespread use of the Internet, users have the ease of accessing collections of university academic documents stored in virtual libraries whose information is of an unstructured type. In recent years, the production and publication of scientific documents in Ecuador have increased considerably, so the search and classification of documents is a fundamental task within information retrieval computer systems. Intelligent search systems allow found information with a high degree of accuracy and similarity. For the development of this project, academic documents from the Ecuador Network of Open Access Repositories (RRAAE) were retrieved using a glossary of terms in the area of science and technology. For the recovery of documents, the web scraping technique was used and its results were stored in a cloud database in JSON format. In the recovered documents, NLP techniques were applied to clean and homogenize the unstructured information. Two similarity metrics were used to measure the divergence between the retrieved documents, and similarity matrices were generated based on the title, keywords, and abstract, which were then unified into a weighted matrix. The results of the system are displayed in a web interface that, through the use of graphs, shows the relationship between the linked documents. The operation of the similarity system was validated through functional tests through experimentation with a collection of 30 queries with indexed and non-indexed terms in the input of the information retrieval system. The experiments showed that for indexed terms, the system performs better.

Keywords: Vector cosine similarity · Web scraping · Connected papers · Jaccard · Artificial intelligence · University repositories · DSpace · Graphs

1 Introduction

During the last years, the volumes of information have grown due to the appearance of new technologies [25], which allow the user to access a large collection of documents through simplified search strings. The largest amount of information

I. Sserwanga et al. (Eds.): iConference 2023, LNCS 13972, pp. 1–17, 2023.
https://doi.org/10.1007/978-3-031-28032-0_28

disseminated in the network is of unstructured type [32], therefore, it is necessary to have intelligent search systems that allow finding correlated information, with a high degree of accuracy and similarity.

In generic form, similarity can be defined as the correlation between two or more objects in which common features can be found [18]. In computer science, a similarity metric is used as a measure of divergence between various instances [27]. Thus, several concepts in the field of Artificial Intelligence (AI) and Machine Learning (ML) incorporate this notion to be able to find patterns in various datasets that allow the prediction to be executed between two or more objects [15]. Natural Language Processing (NLP) is a subdiscipline of ML that combines computational linguistics using rule-based models, statistical models, and deep learning. [24]. NLP has made it possible to improve search systems, due to their inherent ability to assimilate the human colloquial language, allowing Information Retrieval (IR) systems to have more accurate results [5,16,22,35].

The massive use of the Internet meant that many collections of documents were published in digital format to facilitate their search. A survey conducted by Deloitte in 2019, evaluated 12 institutions in the United States and Canada on issues related to the adoption of AI as a service tool in social welfare. [6]. The results show that it exists an 80% of unstructured texts, which are the most significant standardized, generating a greater disorganization of information and difficulties in the search, collection, and analysis of textual data [1,10]. In this context, the International Federation of Library Associations (IFLA) [12] proposes to implement ML algorithms to improve document retrieval, with the aim of automating content classification using NLP.

Within the report carried out by Deloitte for Latin America, a comprehensive panorama was presented in the implementation of computational linguistics. This has allowed one to advance and automate the development of institutional university repositories, Registry of Open Access Repositories (ROAR), and Directory of Open Access Repositories (OpenDOAR) in the Ranking Web of Repositories. The report showed that several countries in the region have digital repositories to implement an open-access digital strategy that contributes to a network of virtual libraries for the storage of complete text files, scientific articles, books, and theses. Thus, Brazil has 45 repositories, Colombia with 26, Ecuador with 24, and Argentina with 19.

Ecuador is one of the 12 countries surveyed in Latin America that have digital repositories, with an approximate collection of 300000 documents [21]. Due to the high demand for digital documents from institutional repositories, Ecuador had to implement transversal policies to manage, organize, and store these documents. Academic and scientific information in Ecuador is not centralized and presents difficulties in access and search, as each institutional repository has its own policies, styles, and organization of information [20].

DSpace is a software that provides open-access administrative tools and facilitates the management of digital collections of documents. Generally, university communities use it for the management of their institutional academic repository [2]. In Ecuador, DSpace is used mainly as a de facto standard in university repositories, but most universities have worked independently on the construction of their institutional repositories, that is, through academic policies that establish each of the versions of DSpace. To try to group the information, the Ecuadorian government implemented the Ecuador Open Access Repositories Network (RRAAE) [26], used at the educational level to have access to academic work, where some of the documents hosted in the DSpace tool are retrieved at the national level. RRAAE is managed and administered by the Ecuadorian Corporation for the Development of Research and Academia (CEDIA).

The use of AI and NLP techniques in the field of scientific documents and institutional repositories is poor [7,28]. These techniques allow us to develop much more efficient and accurate search tools, which allow users to navigate comfortably and perform better searches for obtaining consistent results. The RRAAE does not have ML algorithms that automate search processes and ensure the quality of information retrieved.

Some of the university institutional repositories in Ecuador have missing files and the information retrieval processes, when entering a query, are not very relevant and with few similarities between the retrieved documents. This article presents a tool that allows the visualization of similarities between academic documents, that are recovered from topics related to the computational area, from several university repositories hosted within the RRAAE in Ecuador. After the information retrieval stage, the related documents are represented by a graph that allows one to visualize the relationship between them, making the information more intelligible to the user.

1.1 Previous Work

Several investigations focused on the classification and grouping of documents based on similarity metrics [19]. The similarity of documents is one of the factors intrinsically related to the field of AI and is often an expensive task due to the vast number of Internet documents. There are several algorithms and methodologies for searching and indexing similar documents, such as ML-based methods with classification tasks, clustering, and content-based systems [13,23].

Thus, the research of Yeh et al. [34] provides a review of machine learning approaches and document rendering techniques, using a training dataset of 862 articles drawn from FlyBase and 213 new articles were used for the test phase. The data were cleaned, except for the change in a format considered important for the definition of the characteristic of each token. A binomial classification task was executed, where the system returns a Boolean depending on the new article's membership in the already trained group of specific topics. The results of the classification evaluation had a recall of 85%.

Han et. al. [11] present a simple document classification clustering algorithm based on the centroid. The analysis shows that the measure of similarity allows for dynamically adjusting the documents belonging to different classes and densities, with the aim of grouping the documents according to the dependencies between the terms of the different classes. Then, allows the classification of a new document belonging to different areas based on the highest percentage of similarity, for which the clustering algorithm uses the cosine distance. The results show that better performance is obtained compared to the kNN and C4.5 algorithms.

Vijayarani et al. [31], evaluate the performance of the classification algorithms, Naïve Bayes and Lazy, with a dataset of 80,000 instances and four attributes, with the Weka software platform. The algorithms analyzed a collection of clean and standardized text documents. The classifiers had a performance of 85.7% accuracy for the Naïve Bayes algorithm and 89.9% for Lazy.

In [14], an empirical evaluation of similarity models is carried out in text documents, taking as ground truth peer review by 83 university students. The evaluation was carried out by 29 men and 54 women with a mean age of 19.6 years, who evaluated the documents with a range of similarity between 0 and 1. Around 350 documents were used, with a length of corpus variable between 51 to 126 words. The average recall between the predicted correlations of the similarity measures and the evaluation results for each university student was 0.605. Measures of similarity used for experimentation were: Jaccard, cosine distance, and the overlap coefficient.

Currently, there is a web tool developed by Alex Tarnavsky Eitan, Eddie Smolyansky, Itay Knaan Harpaz, and Sahar Perets, called *Connected Papers* [3], which analyzes approximately 50,000 scientific articles from different databases such as arXiv [17], Semantic Scholar [4,8], PubMed [33], etc. and look for similarities between them. In this tool, the user searches for a topic of scientific interest, and the application calculates the similarity with other documents. For the representation of relationships in graphical form, use the force-directed graph [9], which starts from the similarity metric of the documents, taking as a parameter the citation and bibliography of the articles to obtain a relationship between them. The recovered articles are mostly related to the area of medical sciences, and the tool can only retrieve documents that have a DOI as a unique identifier. On the other hand, our research proposes to develop a similar visualizer paper documents from the university repository DSpace of Ecuador, analogous to *Connected Papers*, taking into account the particularity of Ecuadorian articles, repositories, and research.

2 Methodology and Materials

2.1 Information Retrieval and Dataset Generation

To establish measures of similarity between different documents, it was necessary to generate a dataset of academic articles from the DSpace of Ecuadorian Universities. No academic institution nationwide has an API that allows direct communication and extraction of information from its repository. The CEDIA network, in collaboration with the Secretary of Higher Education, Science, Technology, and Innovation (Senescyt), through the RRAAE project, grouped several repositories of academic and scientific institutions in the same space. The RRAAE collects the documents from the university repositories periodically, which makes use of the DSpace software, and allows the administration of a large number of documents for better organization and grouping.

The collection of documents was carried out with web scraping. This technique initially extracts information from the HTML code of the RRAAE according to a set of terms that were pre-established with lexemes related to the area of science and technology. The set of lexemes was obtained from information sources from organizations that standardize the international nomenclature for the fields of science and technology, such as UNESCO, Incibe, NextU, and Oracle. Given that these organizations present a wide set of terminology, a sampling of specific terms was carried out, avoiding general lexemes that are usually used in other areas. Therefore, the glossary was limited to a total of 332 items, which can be consulted in the open source repository GitHub https://github.com/JanneJaime/visualizador_similitud/blob/main/palabras_permitidas.xlsx.

The dataset collected from the RRAAE is a collection of academic documents $D_i(i = 1, ..., n)$, where $n = 8378$ is the total number of retrieved documents. For each D_i, seven attributes were recovered, allowing for a broad and unambiguous description of each document. The three attributes used to describe the content of the academic document and on which similarity metrics will be applied are the *Title* T_i, the *Keywords* PC_i, and the *Abstract* R_i. For the variables T_i and PC_i similarity measure will be used Jaccard, while for R_i, which contains a greater amount of tokens repeated, the cosine similarity [29]. The remaining four attributes *ID*, *ISBN*, *Repository name*, *Publication date*, and *Author*. These attributes serve as informative fields in the document and are displayed in the results of our web tool.

2.2 Development Methodology

The development of the project was divided into three phases with the aim of optimizing resources, segmenting specific processes, and reducing response times for the Web tool in the extraction, cleaning, processing, storage, and presentation of D_i. Development used Python 3.9.7 [30] as a programming language executed through an interpreter, i.e., there is no need for compilation. The experiments were carried out on a Microsoft Azure Server with 4 GB of RAM and 500 GB of storage.

For the first phase, the data set with D_i documents retrieved based on terms related to science and technology. Web scraping used the Python library *BeatifulSoup* to extract the information in HTML format from the RRAAE repository. The multiprocessing technique through threads was also applied. When querying the RRAEE, 20 results are obtained per page, and the implementation of threads allows one to retrieve all the results in a single execution time. In addition, it was synchronized with a method called *barrier*, which blocks the threads until they have completed each task. When these techniques were applied for the extraction of D_i, a shorter processing time was obtained that lasted approximately nine hours.

The collection of documents recovered from the RRAAE was stored in JSON format. This information, of unstructured type, is stored in the cloud database

Firebase, which has two NoSQL services. One of these services, called *Realtime Database*, allows reading and writing data in real-time, and in its free version, it has a storage limit of 1 GB with a data transfer of 360MB per day. The second service is *Firestore*, which is a flexible and scalable database service that offers several quotas of operations in its free version. Offer 50,000 operations for reading, 20,000 for writing, and 20,000 operations for data deletion, per day. This free service was used for this project, which allows the handling of information on a large scale and with cloud deployment.

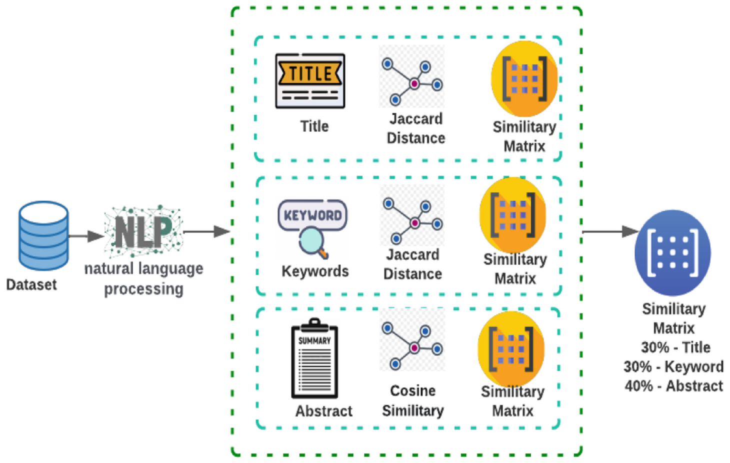


Fig. 1. Implementation of the NLP processes on the retrieved documents and calculation of the weighted similarity matrix.

Figure 1 shows the architecture of the second phase, which takes as input the query of D_i documents housed in *Firestore* and applies various NLP techniques. NLP processes allow normalization, cleaning of special characters, conversion to lowercase letters, tokenization, and elimination of stop words in the corpus texts of articles recovered in Spanish.

For each D_i were extracted T_i, PC_i, and R_i attributes. In the first two attributes, the level of similarity of the entire collection of documents is calculated using the coefficient Jaccard. This coefficient measures the similarity between two datasets and the operating range is $J \rightarrow \mathbb{R} \in [0, 1]$, where a value close to 0 implies low similarity between documents, while a value close to 1 indicates a high degree of similarity, with $j \neq i, (j = 1, ..., n)$. Eq. 1 shows the index particularization J for T_i y PC_i.

$$\begin{aligned} J(T_i, T_j) &= \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \\ J(PC_i, PC_j) &= \frac{|PC_i \cap PC_j|}{|PC_i|} \end{aligned} \tag{1}$$

To measure the divergence of the *Abstract* attribute, R_i, a bag of words is used that applies *Term Frequency Inverse Document Frequency* (TF-IDF) as a weighting system and uses the vector cosine similarity metric. In this model, documents are considered vectors in a multidimensional space, in which the Salton cosine measures their separation angle $cos \rightarrow \mathbb{R} \in [0,1]$. Thus, this coefficient for a pair of documents i y j is defined by Eq. 2.

$$\cos(\vec{R_i}, \vec{R_j}) = \vec{R_i} \cdot \vec{R_j} = \sum_{i=1}^{|n|} R_i R_j \ , \tag{2}$$

For each of the three attributes T_i, PC_i y R_i, a similarity matrix was generated, using its corresponding metric. To unify the three matrices, a weighting process is carried out in each matrix, thus, in the matrix of titles and keywords, a weighting of 0.3 and for the summary matrix a value of 0.4. This implies that in this model R_i provides more information, for the calculation of the similarity, than the other two attributes. As a result, a weighted similarity matrix is obtained MT. The Algorithm 1 refers to the process of generating MT.

Algorithm 1. Weighted Similarity Matrix

```
Input: $D_i$
Output: $MT$
 1: Step 1: Initialization
 2: $T_i \leftarrow [1, ..., n]$;
 3: $R_i \leftarrow [1, ..., n]$;
 4: $PC_i \leftarrow [1, ..., n]$;
 5: for $i \leftarrow 1, n$ do
 6: Step 2: Attribute Extraction
 7:     $T_i \leftarrow D_i[Title]$;
 8:     $PC_i \leftarrow D_i[Keywords]$;
 9:     $R_i \leftarrow D_i[Abstract]$;
10: Step 3: NLP
11:     $T_i \leftarrow NLP(T_i)$;
12:     $PC_i \leftarrow NLP(PC_i)$;
13:     $R_i \leftarrow NLP(R_i)$;
14: end for
15: Step 4: Similarity Matrix
16: $M_{T_i} \leftarrow Jaccard(T_i)$;
17: $M_{PC_i} \leftarrow Jaccard(PC_i)$;
18: $M_{R_i} \leftarrow Cosine(R_i)$;
19: Step 5: Weighing
20: $M_{T_i} \leftarrow (M_{T_i} * 0.3)$;
21: $M_{PC_i} \leftarrow (M_{PC_i} * 0.3)$;
22: $M_{R_i} \leftarrow (M_{R_i} * 0.4)$;
23: Step 6: Total Matrix
24: $MT \leftarrow (M_{T_i} + M_{PC_i} + M_{R_i})$;
```

For the execution of the second phase, the *Google Collaboratory* Jupyter Notebook was used to execute Python code. This development environment provides free access to storage and processing resources in the cloud. Since the collection of documents is from $n = 8378$, the resulting weighted similarity matrix, MT, is a square matrix of $n \times n$. The generation of this matrix requires computational resources and processing time for its calculation. The virtual machine assigned by *Google Colaboratory*, is elastic in nature, that is, it assigns computational resources in a variable way depending on the process to be executed. For matrix generation MT a virtual machine was allocated up to 12.68 GB in RAM, and a 107.72 GB disk and the approximate time execution was three hours. In this sense, the RI web system will not calculate the matrix MT in real time but will use a plain text file .TXT that contains the pre-calculated matrix and is stored on the web application server. Before a new query, the system will only extract a submatrix of MT, therefore, the latency to search for similarities between documents is low.

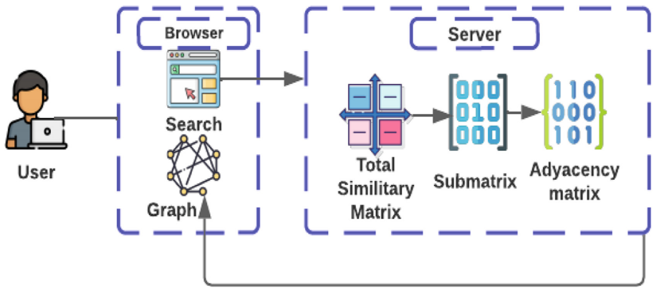


Fig. 2. Process of presentation of similarity results between documents linked by means of a graph.

The scheme of Fig. 2 shows the processes executed in the third phase to obtain the relationship between the linked documents by means of a graph. In this stage, a viewer was implemented to show the relationship between D_i. The process invokes MT and computes a submatrix SM, which satisfies a query Q made by the user. The submatrix SM calculates the graph of the relative neighborhood, with the Python function *returnRNG*, and returns an adjacent array MA which has the same dimension as SM. Algorithm 2 summarizes the process for generating the adjacency matrix. For one Q it is verified that the value of each D_i, which corresponds to the node N in the graph, is greater than 0, and thus it is guaranteed that there is a link E between the nodes. Finally, it is sent to the web front-end, in a file with extension JSON, the information of the nodes obtained from each of the links that are assigned in a dictionary with the keys origin-destiny. The front-end is responsible for displaying the similarity results between Q and D_i to the users through an interactive graph. In addition, its informative attributes can be displayed in each document.

2.3 Development of the Back-end and Front-end of the Application

For the development of the web page, the client-server model is used, made up of three blocks: API, back-end, and front-end, as shown in Fig. 3. The API performs web scraping operations on the RRAAE page to retrieve academic documents. A parameter was implemented that limits the response time to 15 s, and when this is exceeded, the next document continues. The result of the entire operation is a set of documents in JSON format.

The back-end was developed with the Python programming language using the Flask framework. For the design of the software, the Model View Controller (MVC) architecture pattern was used, which facilitates the connection with the database *Firestore*, where the collection of documents obtained by the API is housed. Finally, within the back-end the *MT* and *SM* of the documents are calculated for consultation by the user.

For the development of front-end languages, *HTML5*, *CSS3*, and *JavaScript* were used to structure, design, and give interactivity to the system. Furthermore,

Algorithm 2. Adjacency Matrix

```
Input: $MT, D_i, Q$
Output: $N, E$
 1: Step 1: Initialization
 2: $SM \leftarrow SM \subset MT$;
 3: $MT \leftarrow read(MT)$;
 4: $Q \leftarrow read(Q)$;
 5: Step 2: NLP
 6: $Q \leftarrow NLP(Q)$;
 7: Step 3: Query Database
 8: $D \leftarrow searchD_i(Q)$;
 9: Step 4: Generate Sub-array
10: for $i \leftarrow 1, n$ do
11:     for $j \leftarrow i+1, n$ do
12:         $SM_{i,j} \leftarrow MT.item((D_i, D_j))$;
13:     end for
14: end for
15: Step 5: Adjacency Matrix
16: $MA \leftarrow returnRNG(SM)$;
17: Step 6: Node Links
18: for $i \leftarrow 1, n$ do
19:     for $j \leftarrow i+1, n$ do
20:         if $MA_{i,j} > 0$ then
21:             $E(Origin : D_i, Destiny : D_j)$;
22:         end if
23:     end for
24: end for
25: Step 7: Response
26: $N \leftarrow D$;
27: $E \leftarrow E$;
```

the multi-platform library was used *Bootstrap* to generate the adaptive design of the Web and the library *D3* to visualize links and nodes in the form of a graph. The interaction of the HTML elements are executed with the library *JQuery* and *AJAX*, which make requests to the server, in the background, with the aim of obtaining documents to complement the graph.

2.4 Functional Description of the Web Tool

The web application consumes academic articles from the RRAAE, with the aim of providing the user with the possibility of searching and viewing, through a graph, all the documents related to a query regarding the technological area.

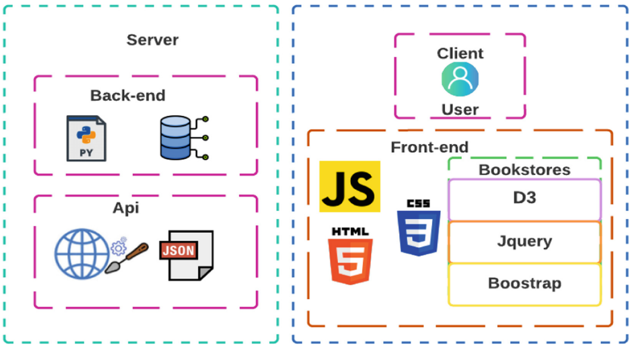


Fig. 3. Web architecture of the client-server model for viewing academic documents.

The home page, as shown in Fig. 4, allows the user to enter a specific query. To execute this process, the user must press the button *Search*, and the system will display similar documents recovered for the query. If no results are found, a message indicating this fact will be displayed on the screen. With the indexes (ID) of the retrieved documents, we proceed to generate the submatrix SM, from the global similarity matrix MT and calculate the adjacency matrix MA to identify the links that each node has in relation to each document. This information is sent to front-end for displaying the results obtained.

Search results are listed on the left side of a new window with their respective attributes. Meta-data displayed on the Web include the unique identifier of the document (ID), the full name of the document, the authors of the manuscript, the year of publication, and the university repository where it is hosted. The user can access the complete document through a hyperlink that will redirect him to the RRAAE repository. In the right section of this same page, a canvas is available, where the relationship graph between documents will be displayed. When the user interacts with the list of results and clicks on one of them, the graph with similar linked nodes is displayed. This last graph is of a dynamic type, ergo, the user can interact with the graph and when passing the cursor

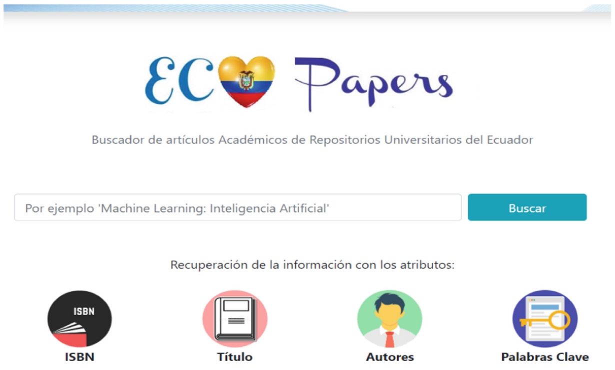
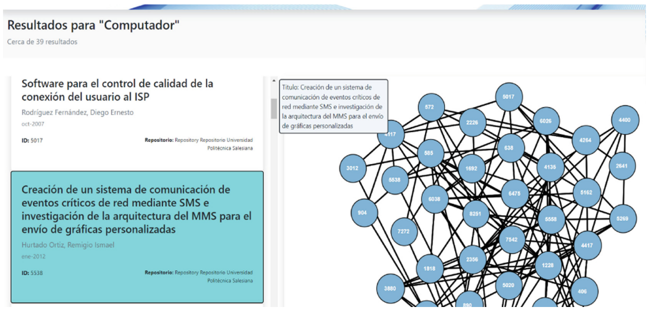

EC Papers
Buscador de artículos Académicos de Repositorios Universitarios del Ecuador
Por ejemplo 'Machine Learning: Inteligencia Artificial'
Buscar
Recuperación de la información con los atributos:
ISBN
ISBN
Título
Autores
Palabras Clave

Fig. 4. Search interface on the main web page for the information retrieval system

over each node, some attributes of the document are displayed. Figure 5 displays the result of a search and retrieval of academic documents related to that entry.



Resultados para "Computador"
Cerca de 39 resultados

Software para el control de calidad de la conexión del usuario al ISP
Rodríguez Fernández, Diego Ernesto
oct-2007
ID: 5017 **Repositorio:** Repository Repositorio Universidad Politécnica Salesiana

Creación de un sistema de comunicación de eventos críticos de red mediante SMS e investigación de la arquitectura del MMS para el envío de gráficas personalizadas
Hurtado Ortiz, Remigio Ismael
ene-2012
ID: 5538 **Repositorio:** Repository Repositorio Universidad Politécnica Salesiana

Titulo: Creación de un sistema de comunicación de eventos críticos de red mediante SMS e investigación de la arquitectura del MMS para el envío de gráficas personalizadas

Fig. 5. Visualization page of the links between nodes of each academic document given a specific search.

3 Experiments and Results

The experiments aim to validate the operation of the Web tool. Therefore, functional tests were carried out, using a heuristic method, which does not explore all the possible outputs of the system. To begin with, certain lexemes were selected to be entered as search parameters for the retrieval of academic documents. Two criteria were used for the selection of terms; the first criterion is to select certain lexemes from the glossary of the 332 terms compiled above, to be extracted from the RRAAE. The set of elements is made up of 215 elements of a token that represent the 65% of the entire glossary and 117 for two or more tokens, which represent the 35% of the dataset. The second search criterion took into account lexemes that are not found within the established glossary but related to the technological area.

For the first experiment, 30 indexed terms were randomly selected from the set of established elements, representing 10% of the dataset. In Table 1 the number of documents retrieved is displayed for each word N_{D_i}, the range, and the mean μ_{SM_i}, of each submatrix of similarities generated by the input lexeme. The range of similarities in the documents is expressed by means of the minimum (Min) and maximum (Max) of the resulting SM. For example, for the word *Metadata*, 32 documents were recovered, where the submatrix SM of similarities has a range between the minimum of 0.007 and the maximum of 0.357, with a mean value of 0.182.

Table 1. Lexemes, number of retrieved documents, measures of dispersion, and central tendency of the similarity sub-matrices for the first experiment.

Lexeme	N_{D_i}	Min	Max	μ_{SM_i}
Modem	39	0.006	0.329	0.167
Driver	30	0.004	0.263	0.133
Router	33	0.005	0.145	0.075
Authenticity	30	0.004	0.6	0.302
Metadata	32	0.007	0.357	0.182
Pentest	9	0.009	0.076	0.042
OSI	37	0.006	0.356	0.181
USB	30	0.007	0.56	0.283
FAT	34	0.002	0.327	0.164
Switch	16	0.009	0.567	0.288
Cracker	13	0.012	0.15	0.081
Export	32	0.008	0.6	0.304
Heuristics	34	0.002	0.335	0.168
IO	32	0.009	0.176	0.092
Machine	30	0.01	0.127	0.068
Domain	39	0.008	0.6	0.304
Bit	32	0.008	0.6	0.304
Browser	34	0.003	0.55	0.276
URL	35	0.009	0.6	0.304
HTTP	30	0.001	0.488	0.244
Computer system	32	0.016	54	0.278
Deep learning	30	0.007	0.6	0.303
Bandwidth	35	0.007	0.6	0.303
Smart machine	36	0.007	0.55	0.278
Business to business	30	0.016	0.533	0.24
Information search	32	0.009	0.6	0.304
Public key	35	0.003	0.3	0.151
Reputation system	30	0.009	0.172	0.09
Linear programming	31	0.008	0.6	0.304
Computerized system	32	0.009	0.565	0.287
	31 $\pm$7	0.007 $\pm$ 0.003	0.428 $\pm$ 0.1821	0.227 $\pm$ 0.1474

Figure 6 represents the sub-matrix of similarities of the term *Metadata*, where the intensity of the color indicates the similarity that each instance has. It is observed that all recovered documents are related to the established Q, and SM will be the input parameter that will allow the adjacency matrix MA, to be created as a graph.

On the other hand, the second experiment aims to observe the result of words that are not found in the pre-established glossary. The experiment was performed with 10 non-indexed terms, as shown in Table 2, with the same attributes as Table 1. It is important to remember that these items were not used as input parameters of the information retrieval system, ergo, it would be expected to obtain more generic documents related to other areas.

In Table 1, the functional tests for the first experiment are shown, where the mean for the 30 values of μ_{SM_i} is 0.227 with a deviation of 0.147. For the second experiment, in Table 2, the mean value of μ_{SM_i} is 0.307 with a deviation of 0.012. These results indicate that the documents retrieved in the first experiment generate matrices with higher similarities, probably because the lexemes used for this case were inputs from the IR system.

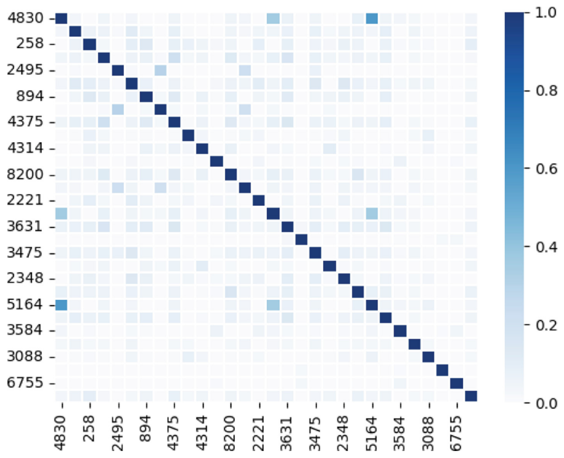


Fig. 6. Similarity sub-matrix of word *Metadata*

The results of the contrast of the means are visualized by the box-and-whisker plot in Fig. 7. The top of the diagram shows the average μ_{SM_i} of the Di indexed and their quartile dispersion. At the bottom is the group of non-indexed documents, where it is observed that the distribution is low between the papers.

In Fig. 8, the number of documents retrieved was considered, both for indexed and non-indexed terms and evaluated in a diagram that allows for observing the differences of means and interquartile dispersions. In the first experiment, an average of 31 documents were recovered; in the second experiment, the average was 476. The latter implies that the information retrieval system obtains more

Table 2. Lexemes, number of retrieved documents, measures of dispersion, and central tendency of the similarity sub-matrices for the second experiment.

Lexeme	N_{D_i}	Min	Max	μ_{SM_i}
File	249	0.001	0.602	0.301
Logic	1045	0.001	0.619	0.31
Circuits	82	0.002	0.6	0.301
Sentences	14	0.012	0.6	0.306
Information	2983	0.001	0.685	0.343
Memory	205	0.001	0.602	0.301
Buffer	24	0.009	0.6	0.304
Flowchart	10	0.012	0.6	0.306
Machine	121	0.001	0.6	0.300
Source code	31	0.007	0.6	0.303
	476 ± 933	0.0046 ± 0.0048	0.6108 ± 0.0267	0.3072 ± 0.0119

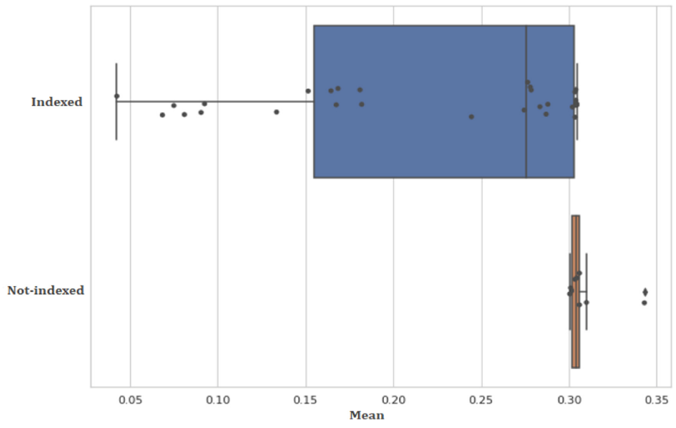


Fig. 7. Box-and-whisker plot of μ_{SM_i} for indexed and non-indexed terms in the glossary

accurate and less generic results when the query Q contains some lexeme of the pre-established in the glossary of terms. A logarithmic scale was used in the box-and-whisker diagram because of the significant difference in magnitudes to be compared.

Finally, in Fig. 9, queries are made using the terms of Tables 1 and 2 of one or more tokens. The retrieval of documents consisting of lexemes of a token with non-indexed terms presents a more significant number of papers collected than queries made with two or more tokens not indexed. Querying the terms of one or more indexed tokens results in a more uniform number of documents collected.

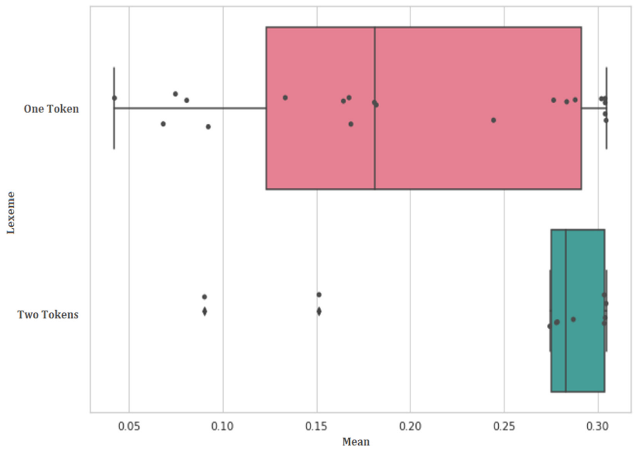


Fig. 8. Box-and-whisker plot of the number of documents retrieved by indexed and non-indexed terms.

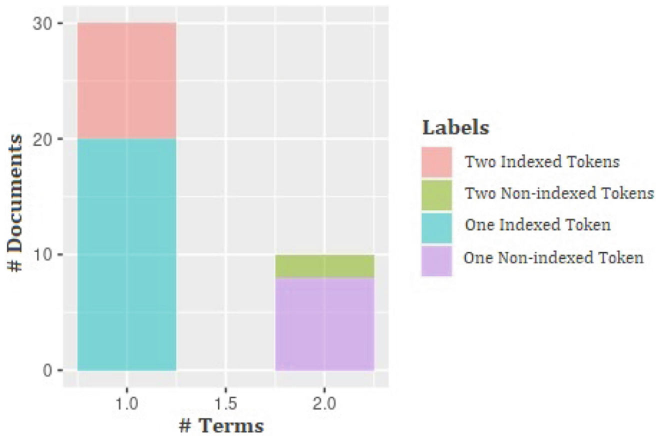


Fig. 9. Frequency diagram of one or more tokens for the experimental process

4 Conclusions and Future Work

This research project collected academic documents from repositories of universities in Ecuador stored in the RRAAE. With this information, a Web search engine was built to find the most similar documents according to the search carried out by the user. Documents were collected using Web scraping and NLP to perform data cleansing and calculate Jaccard and cosine similarity metrics. The work is limited to using a search engine based on divergence metrics. However, as prospects, the intelligent recommendation method could be used to recommend relevant documents to users while retrieving documents, to improve users' retrieval satisfaction. In addition, the system in future versions could use large-scale processing and distributed computing of massive texts in the software architecture, to improve the performance of the tool.

This project's scope is limited to the search for documents related to technical terms, and the matrix of similarities must be updated every certain period since the computational cost of its generation is high.

References

1. Assessing the quality of unstructured data: an initial overview
2. DSpace: An open source dynamic digital repository. https://doi.org/10.1045/january2003-smith. https://dspace.mit.edu/handle/1721.1/29465
3. Eitan, A.T., Smolyansky, E.: Connected papers. https://www.connectedpapers.com/ (2019)
4. Ammar, W., et al.: Construction of the literature graph in semantic scholar. In: NAACL (2018)
5. Cambria, E., White, B.: Jumping NLP Curves: a review of natural language processing research. IEEE Comput. Intell. Mag. **9**(2), 48–57 (2014). https://doi.org/10.1109/MCI.2014.2307227
6. Gómez Mont, C., Martinez Pinto, C.: La inteligencia artificial al servicio del bien social en América Latina y el Caribe: Panorámica regional e instantáneas de doce países
7. Ekanayaka, S.: Combining institutional repositories and artificial intelligence: AI in Academia is Poised to Induce an Unfaltering Growth Stance in Research and Innovation. Research Information, pp. 40–41 (2020)
8. Fricke, S.: Semantic scholar. J. Med. Libr. Assoc. JMLA **106**(1), 145 (2018)
9. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. Softw. Pract. Exper. **21**(11), 1129–1164 (1991)
10. Ghazi, A.N., Petersen, K., Reddy, S.S.V.R., Nekkanti, H.: Survey research in software engineering: problems and mitigation strategies. IEEE Access **7**, 24703–24718 (2018)
11. Han, E.-H.S., Karypis, G.: Centroid-based document classification: analysis and experimental results. In: Zighed, D.A., Komorowski, J., Żytkow, J. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45372-5_46
12. Irvall, B., Nielsen, G.S.: Access to libraries for persons with disabilities: checklist. IFLA Professional Reports, No. 89. International Federation of Library Associations and Institutions (2005). https://eric.ed.gov/?id=ED494537 iSSN: 0168-1931 Publication Title: International Federation of Library Associations and Institutions (NJ1)
13. Kurian, S.K., Mathew, S.: Survey of scientific document summarization methods. Comput. Sci. **21**, 3356 (2020)
14. Lee, M.D., Pincombe, B., Welsh, M.: An empirical evaluation of models of text document similarity. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 27 (2005)
15. Mair, C., et al.: An investigation of machine learning based prediction systems. J. Syst. Softw. **53**(1), 23–29 (2000). https://doi.org/10.1016/S0164-1212(00)00005-4. https://www.sciencedirect.com/science/article/pii/S0164121200000054
16. Mayr, P., et al.: Introduction to the special issue on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL). Int. J. Digit. Libr. **19**(2), 107–111 (2018)

17. McKiernan, G.: arXiv. org: The los alamos national laboratory e-print server. Int. J. Grey Literat. **1**(3), 127–138 (2000)
18. Medin, D.L., Goldstone, R.L., Gentner, D.: Respects for similarity. Psychol. Rev. **100**(2), 254–278 (1993). https://doi.org/10.1037/0033-295X.100.2.254. http://doi.apa.org/getdoi.cfm?doi=10.1037/0033-295X.100.2.254
19. Mohammed, A.J., Yusof, Y., Husni, H.: Document clustering for knowledge discovery using nature-inspired algorithm (2014)
20. Pazmiño-Maji, R., Naranjo-Ordoñez, L., Conde-González, M., García-Peñalvo, F.: Learning analytics in Ecuador: an initial analysis based in a mapping review. In: Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 304–311 (2019)
21. Saltos, W.R.F., Barcenes, V.A.B., Benavides, J.P.C.: Una mirada a los repositorios digitales en ecuador. RECIAMUC **2**(1), 836–863 (2018)
22. Sánchez, D., Martínez-Sanahuja, L., Batet, M.: Survey and evaluation of web search engine hit counts as research tools in computational linguistics. Inf. Syst. **73**, 50–60 (2018)
23. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. (CSUR) **34**(1), 1–47 (2002)
24. Sintoris, K., Vergidis, K.: Extracting business process models using natural language processing (NLP) techniques. In: 2017 IEEE 19th Conference on Business Informatics (CBI), vol. 01, pp. 135–139 (2017)
25. Sosin, A., et al.: How to increase the information assurance in the information age. J. Defense Resour. Manage. (JoDRM) **9**(1), 45–57 (2018)
26. Sumba, F.: Red de repositorios de acceso abierto del ecuador-rraae. In: X Conferencia Internacional de Bibliotecas y Repositorios Digitales (BIREDIAL-ISTEC) (Modalidad virtual, 25 al 29 de octubre de 2021) (2021)
27. Suryakant, Mahara, T.: A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment. Procedia Comput. Sci. **89**, 450–456 (2016). https://doi.org/10.1016/j.procs.2016.06.099. https://www.sciencedirect.com/science/article/pii/S1877050916311644
28. Tonon, L., Fusco, E.: Data mining as a tool for information retrieval in digital institutional repositories. Proceed. CSSS **2014**, 180–183 (2014)
29. Vallejo-Huanga, D., Morillo, P., Ferri, C.: Semi-supervised clustering algorithms for grouping scientific articles. Procedia Comput. Sci. **108**, 325–334 (2017)
30. Van Rossum, G., et al.: Python programming language. In: USENIX Annual Technical Conference, vol. 41, pp. 1–36. Santa Clara, CA (2007)
31. Vijayarani, S., Muthulakshmi, M.: Comparative analysis of Bayes and lazy classification algorithms. Int. J. Adv. Res. Comput. Commun. Eng. **2**(8), 3118–3124 (2013)
32. Weiss, S.M., Indurkhya, N., Zhang, T., Damerau, F.: Text mining: predictive methods for analyzing unstructured information. Springer Science & Business Media (2010). https://doi.org/10.1007/978-0-387-34555-0
33. White, J.: Pubmed 2.0. Med. Ref. Serv. Quart. **39**(4), 382–387 (2020)
34. Yeh, A.S., Hirschman, L., Morgan, A.A.: Evaluation of text data mining for database curation: lessons learned from the KDD challenge cup. Bioinform. **19**(suppl_1), 331–339 (2003)
35. Yue, X., Di, G., Yu, Y., Wang, W., Shi, H.: Analysis of the combination of natural language processing and search engine technology. Procedia Eng. **29**, 1636–1639 (2012)