Seminars in Medical Writing and Education. 2023; 2:43
doi: 10.56294/mw202343

ORIGINAL

Generation of graphs from scientific journal metadata with the OAI-PMH system

Generación de grafos a partir de metadatos de revistas científicas con el sistema OAI-PMH

Denis Gonzalez-Argote[1], Javier Gonzalez-Argote[2]

[1]Universidad Argentina de la Empresa, Facultad de Ingeniería y Ciencias Exactas, Carrera de Ingeniería Informática. Ciudad Autónoma de Buenos Aires, Argentina.
[2]Fundación Salud, Ciencia y Tecnología. Ciudad Autónoma de Buenos Aires, Argentina.

Cite as: Gonzalez-Argote D, Gonzalez-Argote J. Generation of graphs from scientific journal metadata with the OAI-PMH system. Seminars in Medical Writing and Education 2023;2:43. https://doi.org/10.56294/mw202343.

Submitted: 30-09-2023 **Revised:** 23-11-2023 **Accepted:** 28-12-2023 **Published:** 29-12-2023

Editor: Dr. José Alejandro Rodríguez-Pérez

ABSTRACT

Access to scientific and scholarly information has become more accessible than ever before, in large part due to the proliferation of scholarly journals that use metadata systems to expose their content. Notable examples of these systems include the Open Journal Systems (OJS) and the Objects of Learning Metadata Protocol (OAI-PMH), which have significantly simplified the dissemination of research online. However, as these platforms have become essential for research publication and dissemination, new needs arise for publishers and scholars. A technological innovation-type study was conducted by generating codes for primary data generation with the ultimate goal of generating graphs for co-authorship networks and co-occurrence of terms. This paper focuses on a solution to this growing demand for enriched information. We will explore how generating graphs from scholarly journal metadata using the OAI-PMH system can address these specific needs. The generation of graphs from scholarly journal metadata using the OAI-PMH system and Python codes offers a powerful and versatile approach to the analysis of scholarly output. This study demonstrates the applicability of this methodology in the generation of keyword co-occurrence networks and co-authorship networks, providing a deeper and more contextual view of scientific publications. The relevance of this application extends to editors of academic journals as well as to scholars and researchers. For publishers, this tool facilitates the effective presentation of their journals, the evaluation of the quality and content of publications, the selection of categories for indexing, and the identification of emerging trends. On the other hand, for academics, this methodology fosters collaboration, enables more advanced bibliometric analyses, facilitates the presentation of results, and supports informed decision-making in their research areas.

Keywords: Metadata; OAI-PMH; Python; Scientific Writing; Scientific Journals; Graphs; Co-Authorship; Co-occurrence Of Terms.

RESUMEN

El acceso a la información científica y académica se ha vuelto más accesible que nunca, en gran parte gracias a la proliferación de revistas académicas que utilizan sistemas de metadatos para exponer sus contenidos. Ejemplos notables de estos sistemas incluyen el Open Journal Systems (OJS) y el Protocolo de Metadatos de Objetos de Aprendizaje (OAI-PMH), que han simplificado significativamente la difusión de investigaciones en línea. Sin embargo, a medida que estas plataformas se han vuelto esenciales para la publicación y difusión de investigaciones, surgen nuevas necesidades para los editores y académicos. Se realizó un estudio del tipo innovación tecnológica, mediante la generación de códigos para la generación de datos primarios con el fin último de generar grafos para redes de coautoría y coocurrencia de téminos. Este artículo se centra en una solución a esta creciente demanda de información enriquecida. Exploraremos cómo la generación de grafos

a partir de los metadatos de revistas académicas utilizando el sistema OAI-PMH puede abordar estas necesidades específicas. La generación de grafos a partir de metadatos de revistas científicas utilizando el sistema OAI-PMH y códigos en Python ofrece un enfoque poderoso y versátil para el análisis de la producción académica. Este estudio demuestra la aplicabilidad de esta metodología en la generación de redes de coocurrencia de palabras clave y redes de coautoría, lo que proporciona una visión más profunda y contextual de las publicaciones científicas. La relevancia de esta aplicación se extiende tanto a los editores de revistas académicas como a los académicos e investigadores. Para los editores, esta herramienta facilita la presentación efectiva de sus revistas, la evaluación de la calidad y contenido de las publicaciones, la selección de categorías para la indexación y la identificación de tendencias emergentes. Por otro lado, para los académicos, esta metodología fomenta la colaboración, permite análisis bibliométricos más avanzados, facilita la presentación de resultados y respalda la toma de decisiones informadas en sus áreas de investigación.

Palabras clave: Metadatos; OAI-PMH; Python; Escritura Científica; Revistas Científicas; Grafos; Coautoría; Coocurrencia De Términos.

INTRODUCTION

In the digital age, accessing scientific and academic information has become easier than ever, largely due to the proliferation of academic journals using metadata systems to present their content. Major examples of these systems include the Open Journal Systems (OJS) and the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), both of which have significantly simplified the online dissemination of research. However, as such platforms have become essential for the publication and dissemination of research, giving rise to new needs for editors and academics.[1,2,3,4,5]

Academic journal editors often seek a deeper and more contextual understanding of their publications that goes beyond the basic reporting provided by journal web systems such as OJS (Open Journal Systems).[6,7]

Sometimes, it is useful to examine authorship networks or identify the co-occurrence of key terms in their articles. This aids in the effective presentation of their journals, conducting pre-inclusion evaluations for databases, selecting appropriate categories for journal submission for database indexing, identifying major research areas, or conducting bibliometric analyses of their contents.

However, detailed analyses are typically limited to journals indexed in renowned databases such as Scopus, PubMed, or Web of Science (WoS), which are capable of providing this metadata. Alternatively, they are performed on specialized platforms such as Wisdom. Even on these platforms, visualizing and analyzing relationships between metadata and terms is limited.

METHODS

A study was conducted on technological innovation, using code generation for primary data production, with the ultimate goal of generating graphs for co-authorship and term co-occurrence networks.

This article aims to address the increasing demand for enriched information. We will examine how the OAI-PMH system can generate graphs from academic journal metadata to meet specific needs. This methodology allows for the creation of authorship networks and visualization of key terms. It also offers a comprehensive approach to evaluating and improving the visibility, accessibility, and relevance of academic publications. We use examples and case studies to illustrate how this technique can be a valuable tool for editors, academics, and researchers interested in maximizing the impact and understanding of their research in the digital world.

RESULTS AND DISCUSSION

Using Python Code

Before discussing the code used in this study, it is important to understand how it can be implemented in a Python programming environment.

1. *Data Preparation*

Before running any code, make sure you have the data from scientific journals in an appropriate format. In this study, the data is stored in a text file named "datos.txt". Ensure that this file contains the metadata of the journals in a compatible format.

Each journal has a page corresponding to its OAI-PMH record. The “ListRecord” found within this page contains the metadata record for each article.

To systematize each journal, the following system should be used: https://(web de la revista)/index.php/(abreviatura-opcional)/ oai?verb=ListRecords&metadataPrefix=oai_dc

For example: The OAI-PMH page for the journal Community and Interculturality in Dialogue is the following

link: https://cid.saludcyt.ar/index.php/cid/oai, and the “ListRecord” is https://cid.saludcyt.ar/index.php/cid/oai?verb=ListRecords&metadataPrefix=oai_dc.

Save this web page as .XML. Then open the .XML file with Windows Notepad (or a similar program) and save it as plain text (.txt).

2. Python installation

Access the official Python website: https://www.python.org/downloads/. See the download page for the latest versions of Python available for various operating systems. Select the version that matches your operating system (e.g. Windows, MacOS, or Linux). Once the download is complete, run the downloaded file (e.g. "python-3.9.7.exe") to start the Python installer.

On the initial installer screen, make sure to select the option that reads “Add Python X.X to PATH'” (where “X.X” represents the version of Python being installed). This will add Python to the system PATH, making it easier to run Python commands from the command line. Click the “Install Now”' button to begin the installation. The installer will copy the required files and configure Python on your system. Upon completion, the message "Setup was successful” will appear. This confirms that Python has been properly installed.

3. Running a .py file

After installing Python, place the Python script and the saved file “datos.txt” in the same folder (directory).

To run a Python script (.py), open a terminal or command prompt on your operating system, navigate to the directory where the script file is located using the **cd** command, and then run the script using the **python** command followed by the name of the Python file (.py). Make sure the file name exactly matches the name of your script file, including the ".py" extension, and press "Enter". The Python script will run and display its output in the terminal.

The program will generate a file named “Resultados.txt”, which can be used to create co-occurrence of terms or collaborative networks.

Code for Co-occurrence of Spanish and English Keywords

This code uses regular expressions to identify and capture Spanish or English keywords from scientific journal metadata. It searches for the tags '<dc:subject xml:lang="en-US">' for English and '<dc:subject xml:lang="es-ES">' for Spanish. The identified keywords are presented in a results file along with their corresponding identification numbers.

English Keywords

```
import re

def get_palabras(text):
   dato=re.findall(r'<dc:subject xml:lang="en-US">(.+)</dc:subject>',text)

   return dato

with open("datos.txt", "r", encoding='utf-8') as archivo:
   # Lee todos los datos del archivo
   datos = archivo.read()
   datos=datos.split("<metadata>\n<oai_dc:dc\n")
   datos.pop(0)

for i in range(len(datos)):
   lista = get_palabras(datos[i])
   resultado = '\n'.join([f'{i+1}    {element}'  for element in lista])

   if i==0:
      with open("resultados.txt","w", encoding='utf-8') as arch:
            arch.write(resultado)
   else:
       with open("resultados.txt","at", encoding='utf-8') as arch:
            arch.write("\n"+resultado)

print("Tarea Terminada")
```

Spanish Keywords

```
import re

def get_palabras(text):
   dato=re.findall(r'<dc:subject xml:lang="es-ES">(.+)</dc:subject>',text)

   return dato

with open("datos.txt", "r", encoding='utf-8') as archivo:
   # Lee todos los datos del archivo
   datos = archivo.read()
   datos=datos.split("<metadata>\n<oai_dc:dc\n")
   datos.pop(0)

for i in range(len(datos)):
   lista = get_palabras(datos[i])
   resultado = '\n'.join([f'{i+1}    {element}'  for element in lista])

   if i==0:
      with open("resultados.txt","w", encoding='utf-8') as arch:
            arch.write(resultado)
   else:
       with open("resultados.txt","at", encoding='utf-8') as arch:
            arch.write("\n"+resultado)

print("Tarea Terminada")
```

Palabras clave en Portuguese

```
import re

def get_palabras(text):
   dato=re.findall(r'<dc:subject xml:lang=" pt-BR">(.+)</dc:subject>',text)

   return dato

with open("datos.txt", "r", encoding='utf-8') as archivo:
   # Lee todos los datos del archivo
   datos = archivo.read()
   datos=datos.split("<metadata>\n<oai_dc:dc\n")
   datos.pop(0)

for i in range(len(datos)):
   lista = get_palabras(datos[i])
   resultado = '\n'.join([f'{i+1}    {element}'  for element in lista])

   if i==0:
      with open("resultados.txt","w", encoding='utf-8') as arch:
            arch.write(resultado)
   else:
       with open("resultados.txt","at", encoding='utf-8') as arch:
            arch.write("\n"+resultado)

print("Tarea Terminada")
```

Code to generate the authors of the articles

This code uses regular expressions to identify and extract the names of authors from scientific journals. It then presents these author names along with an identification number in a results file.

```
import re

def get_palabras(text):
    dato=re.findall(r'<dc:creator>(.+)</dc:creator>',text)

    return dato

with open("datos.txt", "r", encoding='utf-8') as archivo:
    # Lee todos los datos del archivo
    datos = archivo.read()
    datos=datos.split("<metadata>\n<oai_dc:dc\n")
    datos.pop(0)

for i in range(len(datos)):
    lista = get_palabras(datos[i])
    resultado = '\n'.join([f'{i+1}    {element}'  for element in lista])

    if i==0:
        with open("resultados_autores.txt","w", encoding='utf-8') as arch:
            arch.write(resultado)
    else:
        with open("resultados_autores.txt","at", encoding='utf-8') as arch:
            arch.write("\n"+resultado)

print("Tarea Terminada")
```

Standardization of results

Please note that in the returned results, replace the 4 spaces (“….”) between the record number and the term/authors with a tab (“↹”).

Figure 1. Co-occurrence of English keywords

Generation of co-occurrence of terms and co-authorship networks

The information in the "datos.txt" file can be used to create keyword co-occurrence or authorship networks in specific bibliometric analysis tools (e.g., Bibexcel). These tools enable to import data and configure analysis to create the desired networks.

After generating the networks using these tools, visualize the resulting graphs in network visualization software like Gephi, Sci2 Tool, Cytoscape, Palladio, VantagePoint, UCINet, NodeXL, NetworkX, PNet, NetMiner, Carrot2, TouchGraph, Pajek, BibExcel or VOSviewer.

The authors suggest using VOSviewer to import data and customize network visualizations, including layout, colors, and labels. On the other hand, VOSviewer's visualization is interactive, making it easier to explore and analyze the network. Users can zoom in, search for specific nodes, highlight groups of nodes, and perform more detailed analysis. The VOSviewer-generated network visualization can be exported in various formats, including images and data files, for future presentations or inclusion in research papers.

Application of these codes to the journal Community and Interculturality in Dialogue

Figure 1 shows the graph with the co-occurrence of English keywords in the articles of the journal Community and Interculturality in Dialogue.

Figure 2 displays the co-authorship network graph for the articles published in the journal Community and Interculturality in Dialogue.

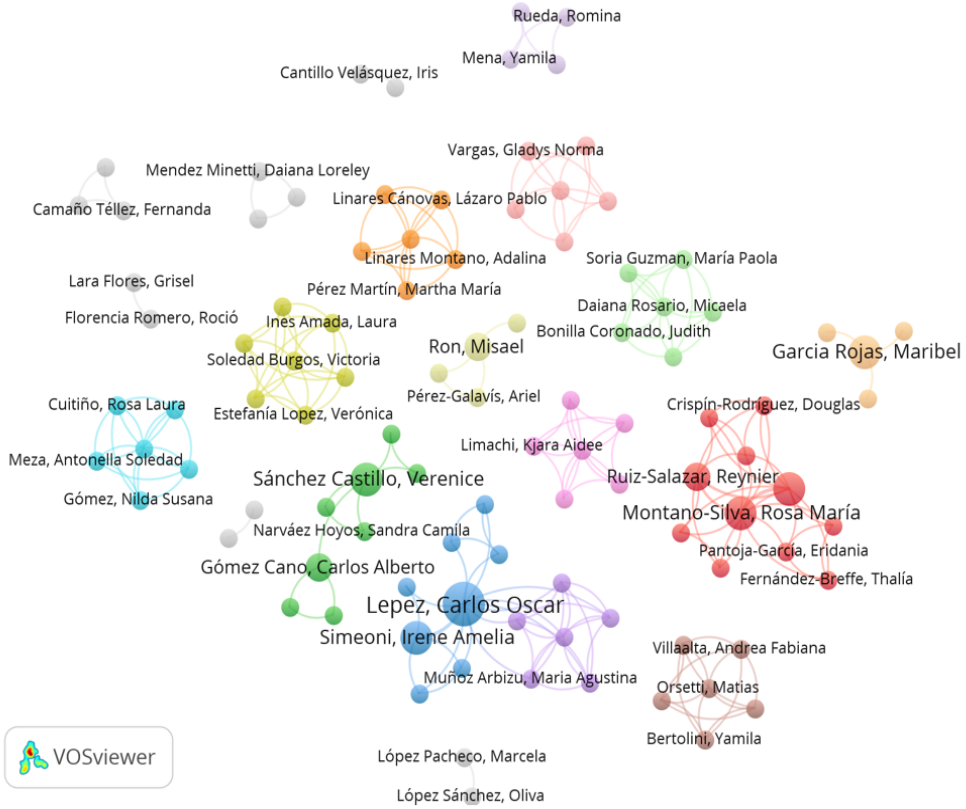

Figure 2. Co-authorship networks

Table 1. Most productive authors

Authors	Number of documents
Lepez, Carlos Oscar	5
Garcia Rojas, Maribel	3
Sánchez Castillo, Verenice	3
Gómez Cano, Carlos Alberto	3
Simeoni, Irene Amelia	3
Abraham-Millán, Yoneisy	3
Montano-Silva, Rosa María	3

https://doi.org/10.56294/mw202343

Cirulli, Alan	2
Silva-Sánchez, Camilo	2
Ruiz-Salazar, Reynier	2
Godoy, Abigaíl	2
Ron, Misael	2
Fuentes Milián, Yangel	2
Veloz Montano, María de las Nieves	2

CONCLUSIONS

Creating graphs from metadata of scientific journals using the OAI-PMH system and Python code offers a robust and flexible approach to academic production analysis. This study has proven the applicability of this method to generate keyword co-occurrence and co-authorship networks, providing a deeper and more contextual insight into scientific publications.

The relevance of this application reaches both academic journal editors and academics/researchers. This tool helps editors effectively present their journals, evaluate the quality and content of publications, select categories for indexing, and identify emerging trends. Instead, this methodology promotes collaboration among academics, enables more advanced bibliometric analyses, facilitates result presentation, and supports informed decision-making in their research areas.

In today's highly competitive and rapidly changing academic and scientific environment, the ability to effectively access and analyze data has become a critical need. The codes and techniques in this study provide an effective solution to meet the growing demands for enriched information. They improve the visibility, accessibility, and relevance of academic publications in the digital age. In the end, this application contributes to the advancement of knowledge and scientific collaboration, taking research to new horizons and discoveries.

REFERENCES

1. Subirats Coll I, Barrueco Cruz JM. Open archives initiative. Protocol for metadata harvesting (OAI-PMH): descripción, funciones y aplicaciones de un protocolo. El profesional de la información 2003;12:99-106.

2. Van de Sompel H, Nelson M, Lagoze C, Warner S. Resource Harvesting Within the OAI-PMH Framework. D-Lib Magazine 2004;10. https://doi.org/10.1045/december2004-vandesompel.

3. Devarakonda R, Palanisamy G, Green JM, Wilson BE. Data sharing and retrieval using OAI-PMH. Earth Sci Inform 2011;4:1-5. https://doi.org/10.1007/s12145-010-0073-0.

4. Jackson A, Han M-J, Groetsch K, Mustafoff M, Cole T. Dublin Core Metadata Harvested Through OAI-PMH. Journal of Library Metadata 2008;8:5-21.

5. McCown F, Nelson ML, Zubair M, Liu X. Search engine coverage of the OAI-PMH corpus. IEEE Internet Computing 2006;10:66-73. https://doi.org/10.1109/MIC.2006.41.

6. Ledesma F, González BEM. Bibliometric indicators and decision making. Data and Metadata 2022;1:9-9. https://doi.org/10.56294/dm20229.

7. Castillo JIR. Identifying promising research areas in health using bibliometric analysis. Data and Metadata 2022;1:10-10. https://doi.org/10.56294/dm202210.

FINANCING

None.

CONFLICT OF INTEREST

No conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Javier González Argote.
Data curation: Denis González Argote.
Formal analysis: Denis González Argote.
Research: Javier González Argote.

Methodology: Javier González Argote.
Resources: Denis González Argote.
Software: Denis González Argote.
Supervision: Denis González Argote.
Validation: Denis González Argote.
Visualization: Denis González Argote.
Writing - original draft: Denis González Argote, Javier González Argote.
Writing - proofreading and editing: Denis González Argote, Javier González Argote.

https://doi.org/10.56294/mw202343