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.

Citar como: 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.

Enviado: 30-09-2023 **Revisado:** 23-11-2023 **Aceptado:** 28-12-2023 **Publicado:** 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.

INTRODUCCIÓN

En la era digital, 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.[1,2,3,4,5]

En muchas ocasiones, los editores de revistas académicas anhelan obtener una visión más profunda y contextual de sus publicaciones más allá de los informes básicos que proporcionan los sistemas web de las revistas como el OJS).[6,7]

En ocasiones resulta oportuno explorar las redes de autoría o identificar la coocurrencia de términos clave en sus artículos, con la finalidad de realizar una presentación efectiva de sus revistas, llevar a cabo evaluaciones previas a la inclusión en bases de datos, seleccionar categorías adecuadas para sus revistas con vistas a su postulación para indexación en bases de datos, identificar las principales áreas de investigación o llevar a cabo análisis bibliométricos de sus contenidos.

Sin embargo, este tipo de análisis detallados a menudo se limita a revistas indexadas en bases de datos de renombre como Scopus, Pubmed o Web of Science (WoS) que son las que pueden proveer estos metadatos, o se lleva a cabo en plataformas especializadas como Wisdom. Incluso en estas plataformas, la capacidad para visualizar y analizar relaciones entre los metadatos y términos es limitada.

MÉTODOS

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. Esta metodología no solo permite la creación de redes de autoría y visualización de términos clave, sino que también ofrece un enfoque integral para evaluar y mejorar la visibilidad, accesibilidad y relevancia de las publicaciones académicas. A través de ejemplos y casos de estudio, ilustraremos cómo esta técnica puede ser una herramienta valiosa para editores, académicos e investigadores que buscan maximizar el impacto y la comprensión de sus investigaciones en el mundo digital.

RESULTADOS Y DISCUSIÓN

Uso de los Códigos en Python

Antes de adentrarnos en los detalles de los códigos utilizados en este estudio, es importante comprender cómo pueden ser implementados en un entorno de programación Python.

1. *Preparación de los Datos*

Antes de ejecutar cualquier código, asegúrese de tener los datos de las revistas científicas en un formato adecuado. En este estudio, los datos se almacenan en un archivo de texto llamado "datos.txt". Asegúrese de que este archivo contenga los metadatos de las revistas en un formato compatible.

Cada revista posee una página de su registro OAI-PMH, dentro de esta página está el “ListRecord” que es el registro de metadatos de cada uno de los artículos.

Para sistematizar cada revista tendía la siguiente sistemática:

https://(web de la revista)/index.php/(abreviatura-opcional)/ oai?verb=ListRecords&metadataPrefix=oai_dc

Ej: Para la revista Community and Interculturality in Dialogue la página del OAI-PMH es el siguiente enlace: https://cid.saludcyt.ar/index.php/cid/oai, y el "ListRecord" es el siguiente: https://cid.saludcyt.ar/index.php/cid/oai?verb=ListRecords&metadataPrefix=oai_dc.

Esta web se debe guardar como .XML, y luego abrir el .XML con el Block de Notas de Windows (o similar) y guardar como un texto simple (.txt).

2. Instalación del Python

Deberá acceder al sitio web oficial de Python en https://www.python.org/downloads/. En la página de descargas, se encontrarán las últimas versiones de Python disponibles para diferentes sistemas operativos. Seleccionar la versión que corresponde a tu sistema operativo (por ejemplo, Windows, macOS o Linux). Una vez que se complete la descarga, ejecuta el archivo descargado (por ejemplo, "python-3.9.7.exe") para iniciar el instalador de Python.

En la primera pantalla del instalador, asegúrate de marcar la casilla que dice "Add Python X.X to PATH" (donde "X.X" representa la versión de Python que estás instalando). Esto agregará Python al PATH del sistema, lo que facilitará la ejecución de Python desde la línea de comandos. Haz clic en el botón "Install Now" para iniciar la instalación. El instalador copiará los archivos necesarios y configurará Python en tu sistema. Cuando termine, verás un mensaje que indica "Setup was successful". Esto confirma que Python se ha instalado correctamente en tu computadora.

3. Ejecución de un Archivo .py

Una vez el Python está instialado, se deberá colocar en la misma carpeta (directorio), el Script de Python y el archivo que se guardó llamado "datos.txt".

Para ejecutar un script de Python (.py), abre una terminal o línea de comandos en tu sistema operativo, navega al directorio donde se encuentra el archivo de script utilizando el comando **cd**, y luego ejecuta el script utilizando el comando **python** seguido del nombre del archivo Python (.py). Asegúrate de que el nombre del archivo coincida exactamente con el nombre de tu archivo de script, incluyendo la extensión ".py", y presiona "Enter". El script de Python se ejecutará y mostrará su salida en la terminal.

Este devolverá un archivo titulado Resultados.txt el que se podrá utilizar para generar la coocurrencia de términos o redes de colaboración.

Código para la Coocurrencia de Palabras Clave en Español e Inglés

Este código utiliza expresiones regulares para identificar y capturar palabras clave en español o inglés de metadatos de las revistas científicas ("<dc:subject xml:lang="en-US">" (inglés) o "<dc:subject xml:lang="es-ES">" (español)) de los. Luego, presenta estas palabras clave junto con un número de identificación en un archivo de resultados.

Palabras clave en Inglés

```
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")
```

Palabras clave en Español

```
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 Portugués

```
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")
```

Código para generar los autores de los artículos

Este código utiliza expresiones regulares para identificar y extraer los nombres de los autores de las revistas científicas. Luego, presenta estos nombres de autores junto con un número de identificación en un archivo de resultados.

```
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")
```

Normalización de resultados

Es necesario tener en cuenta que en los resultados devueltos es hay que reemplazar los 4 espacios (“ ”) entre el número de registro y el término/autores, por una tabulación (“ ”).

Generación de coocurrencia de términos y redes de coautoría

Los datos contenidos en el archivo “datos.txt” pueden ser utilizados para generar coocurrencia de palabras clave o redes de autoría en herramientas específicas de análisis bibliométrico (Ej. Bibexcel u otro). Estas herramientas permiten importar los datos y configurar análisis para crear las redes deseadas.

Una vez generadas las redes en estas herramientas, los grafos resultantes pueden ser visualizados en herramientas de visualización de redes como Gephi, Sci2 Tool, Cytoscape, Palladio, VantagePoint, UCINet, NodeXL, NetworkX, PNet, NetMiner, Carrot2, TouchGraph, Pajek, BibExcel o VOSviewer.

Los autores recomiendan el uso del VOSviewer que permite importar estos datos y personalizar la visualización de la red, incluyendo aspectos como el diseño, colores y etiquetas. Por otro lado visualización en VOSviewer es interactiva, lo que facilita la exploración y el análisis de la red. Los usuarios pueden hacer zoom, buscar nodos específicos, resaltar grupos de nodos y realizar análisis más detallados. La visualización de la red generada en VOSviewer puede ser exportada en el formato deseado, como imágenes o archivos de datos, para su posterior presentación o inclusión en trabajos de investigación.

Aplicación de estos códigos a la revista Community and Interculturality in Dialogue

La figura 1 muestra el grafo con la coocurrencia de palabras clave en inglés de los artículos de la revista Community and Interculturality in Dialogue.

Figura 1. Coocurrencia de términos de palabras clave en inglés

La figura 2 muestra el grafo con la red de co-autoría de los artículos de la revista Community and Interculturality in Dialogue.

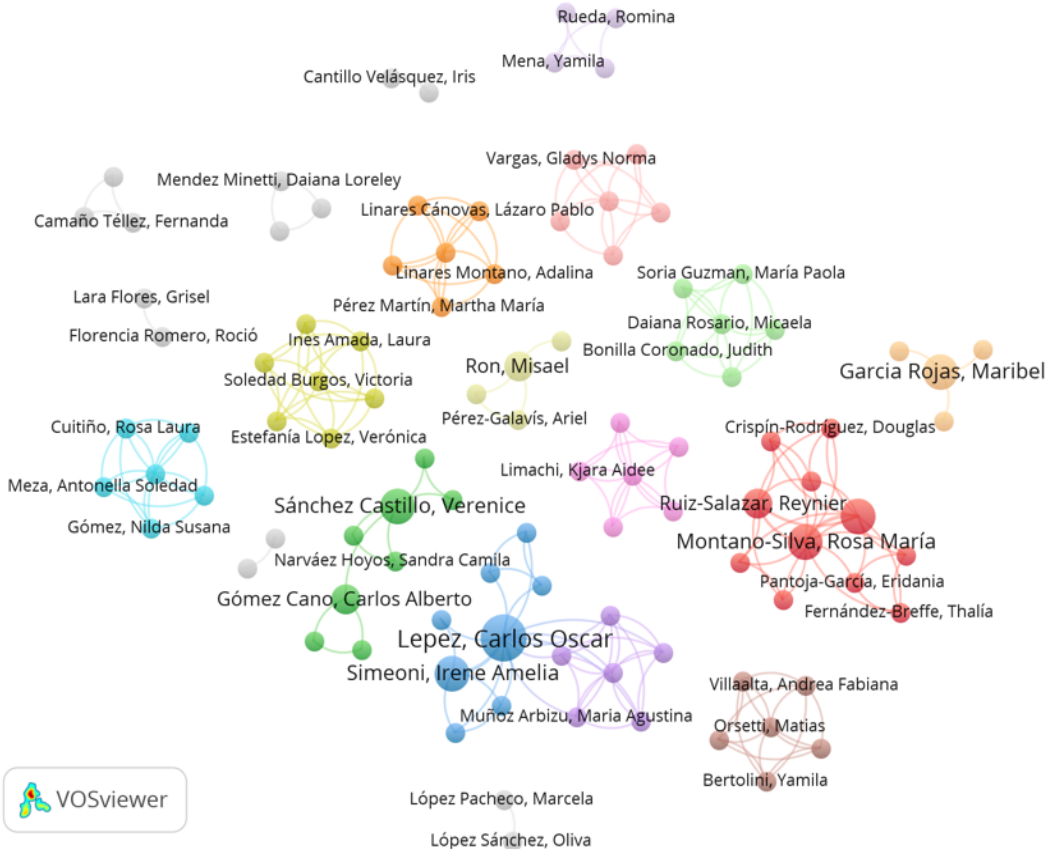

Figura 2. Redes de co-autoría

https://doi.org/10.56294/mw202343

Tabla 1. Autores más productivos

Autores	Ndoc
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
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

CONCLUSIONES

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 ha demostrado 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.

En un entorno académico y científico cada vez más competitivo y dinámico, la capacidad de acceder y analizar datos de manera efectiva se ha convertido en una necesidad crucial. Los códigos y técnicas presentados en este estudio ofrecen una solución efectiva para satisfacer estas demandas crecientes de información enriquecida, mejorando la visibilidad, accesibilidad y relevancia de las publicaciones académicas en la era digital. En última instancia, esta aplicación contribuye al avance del conocimiento y la colaboración científica, impulsando la investigación hacia nuevos horizontes y descubrimientos.

REFERENCIAS BIBLIOGRÁFICAS

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.

FINANCIACIÓN

Ninguna.

CONFLICTO DE INTERESES

Ningún conflicto de intereses.

CONTRIBUCIÓN DE AUTORÍA

Conceptualización: Javier González Argote.
Curación de datos: Denis González Argote.
Análisis formal: Denis González Argote.
Investigación: Javier González Argote.
Metodología: Javier González Argote.
Recursos: Denis González Argote.
Software: Denis González Argote.
Supervisión: Denis González Argote.
Validación: Denis González Argote.
Visualización: Denis González Argote.
Redacción – borrador original: Denis González Argote, Javier González Argote.
Redacción – revisión y edición: Denis González Argote, Javier González Argote.