OJSR: Paquete de R para navegar y recuperar contenido de Open Journal System

» Gastón Becerra
CAETI – Universidad Abierta Interamericana, Universidad de Buenos Aires, CONICET, Argentina
gastonbecerra@sociales.uba.ar

Resumen

Presentamos un paquete para R, titulado ojsr, que permite navegar y recuperar metadatos de la aplicación para manejo editorial *Open Journal System* (OJS). El mismo se puede descargar e instalar desde *Comprehensive R Archive Network* (CRAN). Luego de introducir brevemente el sistema OJS y reseñar las necesidades que las funciones de ojsr buscan atender, así como sus limitaciones, mostramos un breve caso de uso.

KEYWORDS: PAQUETE DE R, WEB SCRAPING, OPEN JOURNAL SYSTEM

ojsr: R Package for Crawling and Retrieving Content from Open Journal System

Abstract

We present an R package, titled ojsr, that aims at crawling and retrieving metadata from Open Journal Systems (OJS), an open source application for editorial management of scientific journals. The package is available on CRAN. Here, after introducing OJS, and reviewing goals and limitations of ojsr functions, we show a brief example.

KEYWORDS: R PACKAGE, WEB SCRAPING, OPEN JOURNAL SYSTEM

1. Introducción

En esta breve comunicación presentamos un paquete (*package*) para lenguaje R (R Core Team, 2017), denominado ojsr, que permite navegar y recuperar metadatos de la aplicación para manejo editorial *Open Journal System* (OJS). El mismo se puede descargar e instalar desde *Comprehensive*

R Archive Network (CRAN)[1], y se encuentra documentado con una vignette[2] que introduce sus principales casos de uso, y un manual de referencia[3].

En las siguientes secciones se introduce brevemente el sistema OJS sobre el cual ojsr opera; se reseñan las necesidades que atiende y sus limitaciones, así como se mencionan sus principales funciones. Finalmente, se anexa un breve caso de uso con código comentado.

2. Acerca de OJS

Open Journal System (OJS) es una aplicación de código abierto diseñada para el manejo editorial de revistas científicas o *journals*. OJS se originó dentro de un proyecto de investigación de la University of British Columbia (Public Knowledge Project, PKP) cuyo objetivo era explorar herramientas y prácticas editoriales para facilitar el acceso abierto (Willinsky, 2006). En palabras de su autor:

If we were going to provide support for open access publishing, and more generally make the case for containing the cost of access, we needed to provide a way to reduce costs. Only by sharply containing costs could journals begin to look at reduced revenues, whether by offering open access to their online edition or by simply making their back issues free (forsaking reprint revenue). We could do this by creating open source software that was specifically developed to manage and publish journals online. The software could be designed so that it called for no greater technical skills on the part of journal editors than were commonly found among university faculty today, namely word-processing, e-mailing, and web-browsing. (Willinsky, 2005, p. 507)

Actualmente, el proyecto cuenta con varias instituciones participantes, aplicaciones similares para la publicación de libros y *pre-prints*, y una amplia comunidad que mantiene, desarrolla y extiende. OJS se distribuye como software de código abierto desde 2001 y, de acuerdo con su sitio web[4], hace 5 años que sostiene aproximadamente 10.000-11.000 revistas en todo el mundo, aunque, como una investigación realizada por miembros del proyecto sugiere, una gran parte de estas instalaciones son de "países en vías de desarrollo" (Edgar & Willinsky, 2010, p. 6).

3. Objetivos y limitaciones de las funciones de ojsr

ojsr fue diseñado para facilitar la tarea de navegar y recuperar contenido, particularmente los metadatos de artículos de un OJS de forma directa, es decir, sin recurrir a la consulta de indexadores o agregadores que acopian el contenido de las revistas, tales como DOAJ o Scopus. De esta manera, es posible incluso recuperar el contenido de revistas que no tienen ninguna indexación, que han indexado parcialmente su contenido, o que sólo se encuentran indexadas en portales pagos.

1 https://cran.r-project.org/web/packages/ojsr/index.html (accedido el 30/06/2020)
2 https://cran.r-project.org/web/packages/ojsr/vignettes/ojsr-vignette.html (accedido el 30/06/2020)
3 https://cran.r-project.org/web/packages/ojsr/ojsr.pdf (accedido el 30/06/2020)
4 https://pkp.sfu.ca/ojs/ojs-usage/ojs-stats/ (accedido el 30/06/2020)

ojsr toma como input la URL de un OJS. Sus funciones de navegación (*crawling*) permiten listar números (*issues*), artículos, galeradas (*galleys*) y hasta resultados de búsquedas. En la mayoría de los escenarios de uso, el usuario no necesita más que conocer una URL del OJS para navegar el archivo completo de la revista utilizando alguna de las funciones de ojsr, tales como:

- get_issues_from_archive: navega y recupera las URLs de número (*issues*) desde la página del archivo (archive) de OJS;
- get_articles_from_issue: navega y recupera las URLs de artículos desde la tabla de contenidos de un número (*issue*) de OJS;
- get_articles_from_search: navega y recupera las URLs de artículos en los resultados de búsqueda (incluso cuando se devuelven varias páginas) de OJS;
- get_galleys_from_article: navega y recupera las URLs de las galeradas (*galleys*: generalmente, el contenido completo del artículo en formato PDF, XML o MP3, u otros archivos auxiliares) a partir de las páginas de los artículos de OJS.

Es importante destacar que ojsr no busca el contenido en la URL provista por el usuario sino que compone la URL requerida para cada tarea de recuperación siguiendo las convenciones de ruteo de OJS. Así, por ejemplo, si se quiere recuperar el listado de artículos de un número, ojsr compondrá la URL de la tabla de contenidos y buscará los links en esta página; o si se requiere recuperar el listado de artículos en los resultados de una búsqueda, ojsr generará la URL de la búsqueda y navegará sus distintas páginas de resultados. Por esta razón, ojsr depende de las convenciones de ruteo de OJS[5], y puede fallar en instalaciones con rutas customizadas.

Luego, es posible recuperar el contenido (*scraping*) de los metadatos de los artículos publicados con OJS, ya sea por la vía de leer el código HTML del artículo (*webscraping*), como de acceder a su registro OAI-PMH (*Open Archives Initiative Protocol for 'Metadata' Harvesting*), con otras funciones de ojsr:

- get_html_meta_from_article: Recupera el contenido de los metadatos de los artículos desde el código HTML de la página;
- get_oai_meta_from_article: Genera la dirección del registro OAI para un artículo de OJS y devuelve sus campos.

Desde su versión 3.1+, OJS se distribuye junto con una API Rest[6]. Estamos convencidos que, en un futuro, tareas de navegación y recuperación de contenido como las que aquí atendemos se facilitarían mucho con un una interfaz (*wraper*) de estas API. No obstante, por el momento esta API se encuentra en un estadío inicial de desarrollo, además de requerir autenticación, de modo que difícilmente pueda ser utilizada por alguien distinto a los administradores del OJS. ojsr puede ser una herramienta útil hasta que este desarrollo avance y sea adoptado masivamente.

Finalmente, ojsr tiene algunas funciones auxiliares que permiten analizar y manipular URLs de OJS, las cuales pueden servir para concatenar los resultados de otras operaciones, o para generar URLs que se puedan manipular desde otros paquetes.

5 https://docs.pkp.sfu.ca/dev/documentation/en/architecture-routes (accedido el 30/06/2020)
6 https://docs.pkp.sfu.ca/dev/api/ojs/3.1 (accedido el 30/06/2020)

4. Un caso de uso

En este breve anexo mostramos un caso de uso de ojsr: recuperamos el contenido de 6 revistas de psicología social de Argentina y Colombia, y exploramos sus principales *keywords*. Las tareas necesarias para el objetivo que nos proponemos son: (1) adquisición de datos (etapa en la que utilizaremos ojsr); (2) limpieza de datos; (3) análisis y visualización. Dado que sólo nos interesa ilustrar el contexto en el que se utiliza ojsr, comentamos el código relevante a la fase de adquisición de datos, y finalmente los gráficos resultantes del análisis. El código completo se encuentra disponible en el repositorio del paquete[7].

En primer lugar, debemos instalar ojsr y cargarlo, junto a los otros paquetes que utilicemos. Luego, aportamos las direcciones de los OJS que nos interesa explorar (aquí en formato de tabla con otra información, para luego incorporar en el análisis) (Cód. 1).

Código 1. *Instalación del paquete y preparación de la tabla de datos a procesar:*

```
install.packages('ojsr', 'tidyverse') # instala desde CRAN
library('ojsr', 'tidyverse') # carga el paquete en memoria
revistas <- data.frame(
  stringsAsFactors = FALSE,
  url = c(
          "https://publicaciones.sociales.uba.ar/index.php/
psicologiasocial",
          "https://revistas.unc.edu.ar/index.php/revaluar",
          "https://dspace.palermo.edu/ojs/index.php/psicodebate",
          "https://revistas.ucc.edu.co/index.php/pe/about",
          "https://revistas.javeriana.edu.co/index.php/revPsycho/index/",
          "https://revistas.unbosque.edu.co/index.php/CHP/" ),
  nombre = c(
             "PSocial (UBA, Arg.)",
             "Revista Evaluar (UNC, Arg.)",
             "Psicodebate (UP, Arg.)",
             "Pensando Psicología (UCC, Col.)",
             "Universitas Psychologica (PUJ, Col.)",
            "Cuadernos Hispanoamericanos de Psicología (UBosque, Col.)" ),
  pais = c( "Arg.", "Arg.", "Arg.", "Col.", "Col.", "Col." ))
```

Todas las funciones de navegación y recuperación de ojsr toman como input una lista de URLs provista por el usuario, y en todos los casos devuelven una tabla con 2 columnas: input_url, la URL provista por el usuario, y output_url, las URLs recogidas. Por lo general, cada *input* (por ejemplo, revista) devuelve más de un resultado (varios números), generando una tabla larga, al estilo *tidy* (Wickham & Grolemund, 2016). Con las URLs de las revistas, podemos recuperar las URLs de los números, y subsecuentemente, de los artículos. El proceso es el mismo si quisiéramos recuperar galeradas.

7 https://github.com/gastonbecerra/ojsr/tree/master/paper (accedido el 04/07/2020)

Luego, con las URLs de los artículos podemos recuperar los metadatos, con los que vamos a hacer nuestros análisis. Las funciones de recuperación de contenido de ojsr devuelven una tabla con 5 columnas: input_url (la URL provista), meta_data_name (el nombre del metadato, e.g., "DC.Date.created"), meta_data_content (el contenido o valor del metadato), meta_data_scheme (el estándar seguido), meta_data_xmllang (idioma). Aquí, particularmente, nos interesan los *keywords*, de modo que podemos armar una nueva tabla filtrando sólo estos datos.

Código 2. *Recupero de información acerca de números, artículos y metada:*

```
numeros <- ojsr::get_issues_from_archive(input_url = revistas$url)
articulos <- ojsr::get_articles_from_issue(input_url = numeros$output_url)
metadatos <- ojsr::get_html_meta_from_article(input_url = articulos$output_url)
keywords <- metadatos %>% filter(meta_data_name %in% c("citation_keywords",
"keywords") )
```

Antes de continuar podemos explorar los datos devueltos, agrupados al nivel de las revistas. Para ello anotaremos cada URL registrada utilizando la función parse_base_url().

Código 3. *Preparación de tabla de números, artículos y metadata:*

```
revistas$base_url <- ojsr::parse_base_url(revistas$url)
numeros$base_url <- ojsr::parse_base_url(numeros$input_url)
articulos$base_url <- ojsr::parse_base_url(articulos$input_url)
metadatos$base_url <- ojsr::parse_base_url(metadatos$input_url)
keywords$base_url <- ojsr::parse_base_url(keywords$input_url)
revistas %>% # vinculamos las tablas entre si
     left_join( numeros %>% group_by( base_url ) %>%
summarise(numeros=n()) ) %>%
     left_join( articulos %>% group_by( base_url ) %>%
summarise(articulos=n()) ) %>%
     left_join( metadatos %>% group_by( base_url ) %>%
summarise(metadata=n()) ) %>%
     left_join( keywords %>% group_by( base_url ) %>%
summarise(keywords=n()) ) %>%
  select(nombre, pais, numeros, articulos, metadata, keywords)
```

La tabla 1 muestra el resultado de esta exploración a julio del 2020:

Tabla 1. *Números, artículos y metadata de revistas:*

Nombre	País	Issues	Art.	Meta	Keyw.
PSocial (UBA)	Arg.	11	69	3176	227
Revista Evaluar (UNC)	Arg.	23	139	7708	357
Psicodebate (UP)	Arg.	25	182	9669	728
Pensando Psicología (UCC)	Col.	17	182	9756	816
Universitas Psychologica (PUJ)	Col.	10	172	10392	773
Cuadernos Hisp. de Psicología (UB)	Col.	13	73	4027	285

Después de recuperar los datos, limpiamos y normalizamos (en nuestro caso, se separaron *keywords*, ya que muchas revistas los servían como un *string* de texto, con las palabras separadas por coma u otros símbolos). Luego, calculamos las frecuencias de repetición por *keyword* y país, y visualizamos en forma de nube de palabras (frecuencia mínima = 3) (fig. 1).

Figura 1. Nube de palabras de *keywords* por país:

5. Conclusiones

En esta comunicación hemos presentado un paquete para R que permite navegar y recuperar metadatos de la aplicación para manejo editorial *Open Journal System* (OJS), titulado ojsr, disponible para descargar e instalar desde *Comprehensive R Archive Network* (CRAN). ojsr es un proyecto en desarrollo y se planean actualizaciones para mejorar su rendimiento, hasta que se encuentre disponible, y se generalice, la posibilidad de hacer este tipo de consultas por la vía de una API Rest, sin las restricciones y limitaciones que se mencionaron para la versión estable más reciente de OJS.

Bibliografía

» Edgar, B. D., y Willinsky, J. (2010). A Survey of the Scholarly Journals Using Open Journal Systems. *Scholarly and Research Communication*, 1(1), 1–22.

» R Core Team (2018). *R: A language and environment for statistical computing*. Recuperado de https://www.r-project.org/

» Wickham, H. y Grolemund, G. (2016). *R for Data Science*. Recuperado de http://r4ds.had.co.nz/

» Willinsky, J. (2005). Open Journal Systems: An example of open source software for journal management and publishing. *Library Hi Tech*, 23(4), 504–519. https://doi.org/10.1108/07378830510636300

» Willinsky, J. (2006). *The Access Principle. The Case for Open Access to Research and Scholarship*. Cambridge: The MIT Press.