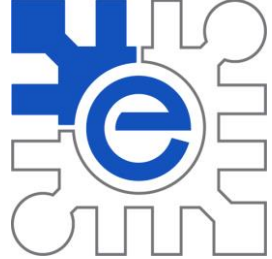


Análisis y comparación de paquetes para el desarrollo de web scraping

David Márquez Mínguez

Departamento de Ciencias de la Computación
Universidad de Alcalá

10 de Junio, 2022

Índice general

Índice general

Contexto y objetivos principales del proyecto

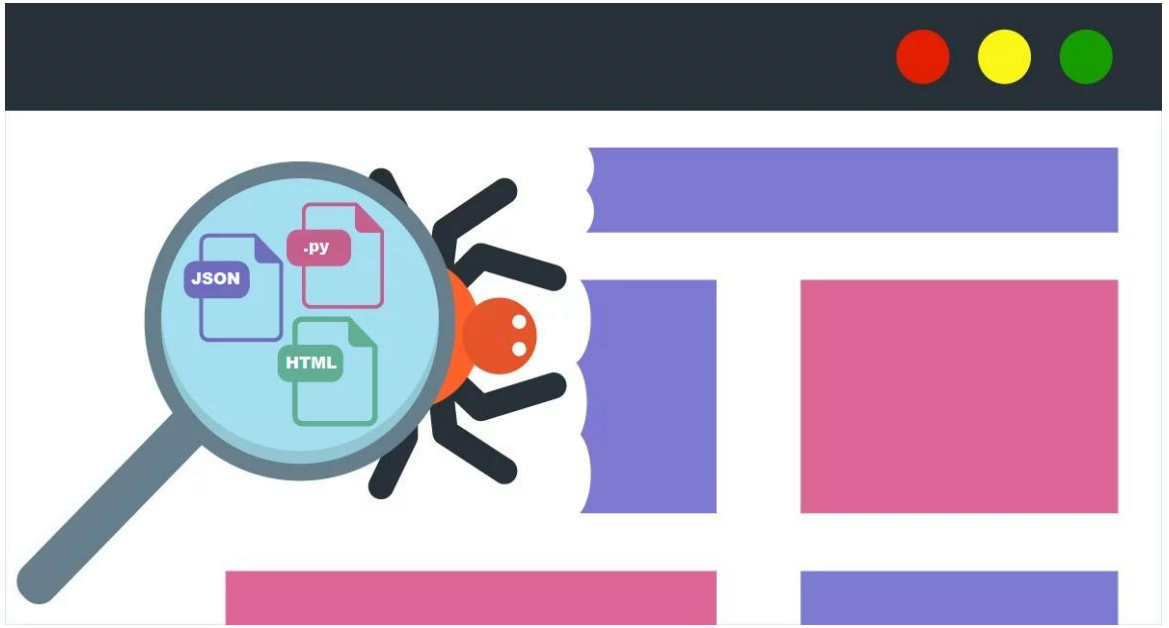
Extracción tradicional vs web scraping

La extracción tradicional es un método:

- Ineficiente
- Poco productivo
- Provoca desestructuración en el conjunto final de datos

El web scraping pretende:

- Automatizar al extracción y almacenamiento
- Incrementar la optimización



Objetivos

Los web scrapers puede dividirse en varios enfoques, ya sean bibliotecas de programación de propósito general, frameworks, o entornos de escritorio.

El objetivo de este proyecto es realizar un análisis **cuantitativo** de las diferentes **técnicas** y **paquetes software** para el desarrollo de un correcto proceso de web scraping:

- ¿Qué solución es la más rentable para el minado web?
- ¿Cuál tiene un mejor rendimiento?
- ¿Es el web scraping una solución válida a los problemas de la extracción tradicional?

Índice general

¿En que consiste el web scraping?

El web scraping o minado web, es una solución tecnológica para extraer información de sitios web, de forma **rápida**, **eficiente** y **automática**, ofreciendo datos en un formato más estructurado y más fácil de usar.

¿Se usa realmente el web scraping?

- Bancos e instituciones.
- Redes sociales.
- Análisis de mercado.
- etc.

¿Cómo funciona el web scraping?

Se **accede** a la web, se **extraen** datos y se **almacenan** en una estructura determinada para su posterior análisis o recuperación.

Un agente de software, también conocido como robot, imita la navegación humana convencional y paso a paso accede a tantos sitios web como sea necesario.

Durante la fase de extracción, los paquetes emplean múltiples herramientas software con el objetivo de conseguir una extracción exitosa:

- Expresiones XPath
- Selectores CSS
- Analizadores + heurística

El uso de cada herramienta dependerá del alcance de los algoritmos de minado.

Índice general

Análisis del mercado de paquetes

Búsqueda de paquetes de R y Python

La búsqueda de paquetes mas comunes dentro del web scraping se centra en R y Python. Gran parte de los desarrolladores publican su trabajo en repositorios de código abierto. Algunos de estos repositorios son: GitHub, CRAN o PyPi.

Análisis del mercado de paquetes

Paquetes encontrados durante la búsqueda

Paquetes propios de Python:

- inscriptis
- Beautiful Soup
- jusText
- news-please
- Libextract
- html_text
- Readability
- Trafilatura
- Dragnet
- Goose3
- Newspaper3k
- BoilerPy3

Paquetes propios de R:

- rvest
- Rcrawler
- html2txt
- scraper
- RSelenium
- BoilerpipeR

Ya sea por la **sencillez** de su heurística, por la **similitud** con otras herramientas, o por la **dificultad** en la instalación y uso de las mismas, algunos paquetes han sido descartados del proceso de evaluación.

Índice general

En una herramienta de minado web, se busca **eficacia**, **optimización** y **velocidad** de ejecución.
El objetivo ahora es establecer una serie de variables que satisfagan dichas condiciones:

- Optimización: Uso de recursos del entorno de ejecución (CPU/RAM).
- Velocidad: Tiempo de ejecución.
- Eficacia: Calidad de la extracción.

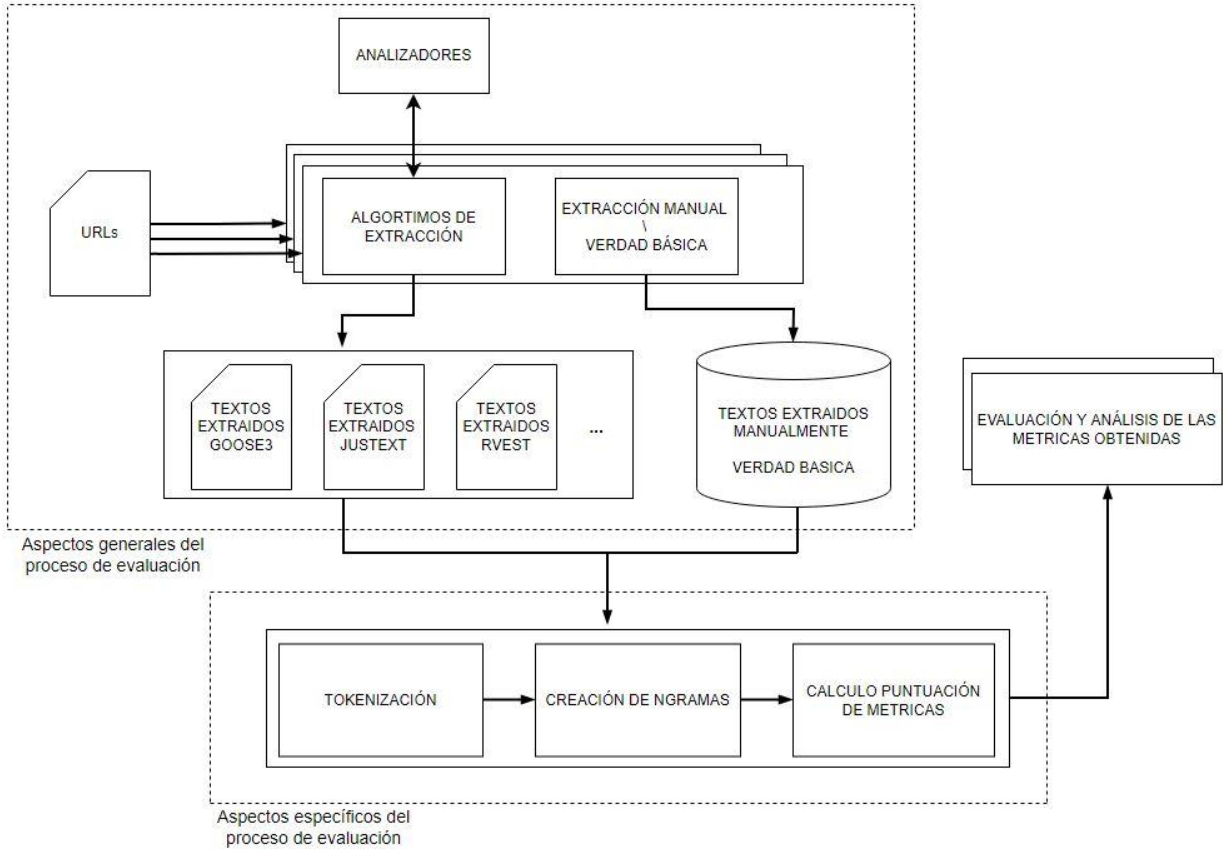
¿Cómo podemos calcular la calidad de extracción de cada herramienta?

Proceso de evaluación

Cálculo de la calidad de extracción

El objetivo de cada paquete será extraer el **contenido principal** de los diferentes documentos con el fin de compararlo con una **verdad básica**. Este contenido principal se basa en:

- Título principal del documento
- Texto principal del documento
- Secciones adicionales(comentarios, reseñas…)



Proceso de evaluación

Heurística basada en n-gramas

Debemos comparar los textos extraídos con el original, pero ¿cómo se puede hacer?
Una primera idea sería **comparar** las **palabras** según la **posición** en la que se encuentren.

MADRID	--	Top-ranked	Rafael	Nadal	has	arrived	in	Madrid	to
lead	Spain	in	the	new-look	Davis	Cup	Finals.\n\n\"It's	a	
new	competition	and	we	must	be	focused,\"	Nadal	said	Sunday.

MADRID	\u2014	Top-ranked	Rafael	Nadal	has	arrived	in	Madrid	to
lead	Spain	in	the	new-look	Davis	Cup	Finals.\"It\u2019s	a	
new	competition	and	we	must	be	focused,\"	Nadal	said	Sunday.

Problemas:

- ¿Qué ocurrirá cuando la disposición del texto no sea exacta?
- Puede que ambos fragmentos no tengan el mismo tamaño

Proceso de evaluación

Heurística basada en n-gramas

Hay que pensar en una nueva heurística que permita comparar textos de forma efectiva.
Otra posibilidad sería la búsqueda de palabras coincidentes en ambos textos.

MADRID	--	Top-ranked	Rafael	Nadal	has	arrived	in	Madrid	to
lead	Spain	in	the	new-look	Davis	Cup	Finals.\n\n\"It's	a	
new	competition	and	we	must	be	focused,\"	Nadal	said	Sunday.

MADRID	\u2014	Top-ranked	Rafael	Nadal	has	arrived	in	Madrid	to
lead	Spain	in	the	new-look	Davis	Cup	Finals.\"It\u2019s	a	
new	competition	and	we	must	be	focused,\"	Nadal	said	Sunday.

Problemas:

- Puede que la palabra pivote este repetida mas de una vez, lo que forzaría un resultado irreal.

La solución finalmente propuesta pasa por el uso de n-gramas, donde ambos texto se dividen en ‘bloques’ de tamaño n.

MADRID		Top-ranked	Rafael	Nadal	has	arrived	in	Madrid
to	lead	Spain	in	the	new-look	Davis	Cup	Finals.
It's	a	new	competition	and	we	must	be	focused,\"
Nadal	said	Sunday.						

MADRID	\u2014	Top-ranked	Rafael	Nadal	has	arrived	in	Madrid
to	lead	Spain	in	the	new-look	Davis	Cup	Finals.
\"It\u2019s	a	new	competition	and	we	must	be	focused,\"
Nadal	said	Sunday.						

La **repetición** de n-gramas ya no es preocupante, existe una mínima probabilidad de que esto ocurra a lo largo del texto. La **posición** de los mismos tampoco es relevante.

Una vez que ambos fragmentos de texto han sido convertidos en n-gramas, se deben **comparar** y **calcular** una puntuación de los mismos. Para ello la puntuación se divide en:

- True positive: n-gramas coincidentes en ambos textos.
- False positive: n-gramas que aparecen en el texto extraído pero no en el base.
- False negative: n-gramas que aparecen en el texto base pero no en el extraído.

Proceso de evaluación

Cálculo de métricas

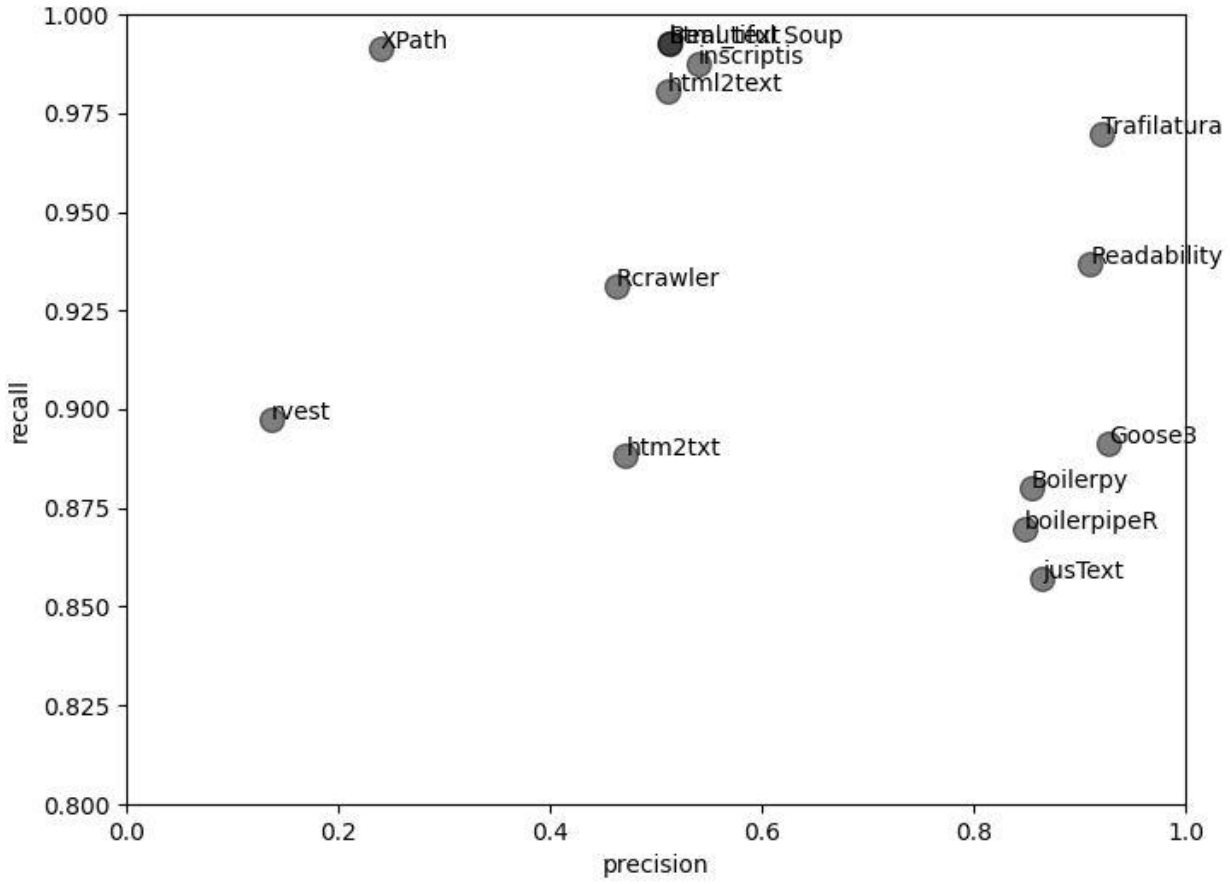
Las puntuaciones calculadas anteriormente sirven ahora para determinar diferentes métricas como precision, recall, Accuracy o F1-score. Debemos recordar también las métricas relativas a la optimización y velocidad de extracción:

- Recall: Eficacia de captar contenido principal.
- Precision: Exclusión de contenido que no es el principal.
- Accuracy: Predicciones correctas de texto extraído.
- F1: Calidad en general de la extracción.
- Uso de CPU: Cantidad de procesador que consume el proceso al completo.
- Uso de RAM: Cantidad de memoria física que el proceso está empleando.
- Velocidad de extracción: Tiempo en realizar el proceso completo de minado.

Índice general

Resultados obtenidos

Calidad de la extracción precision-recall



Resultados obtenidos

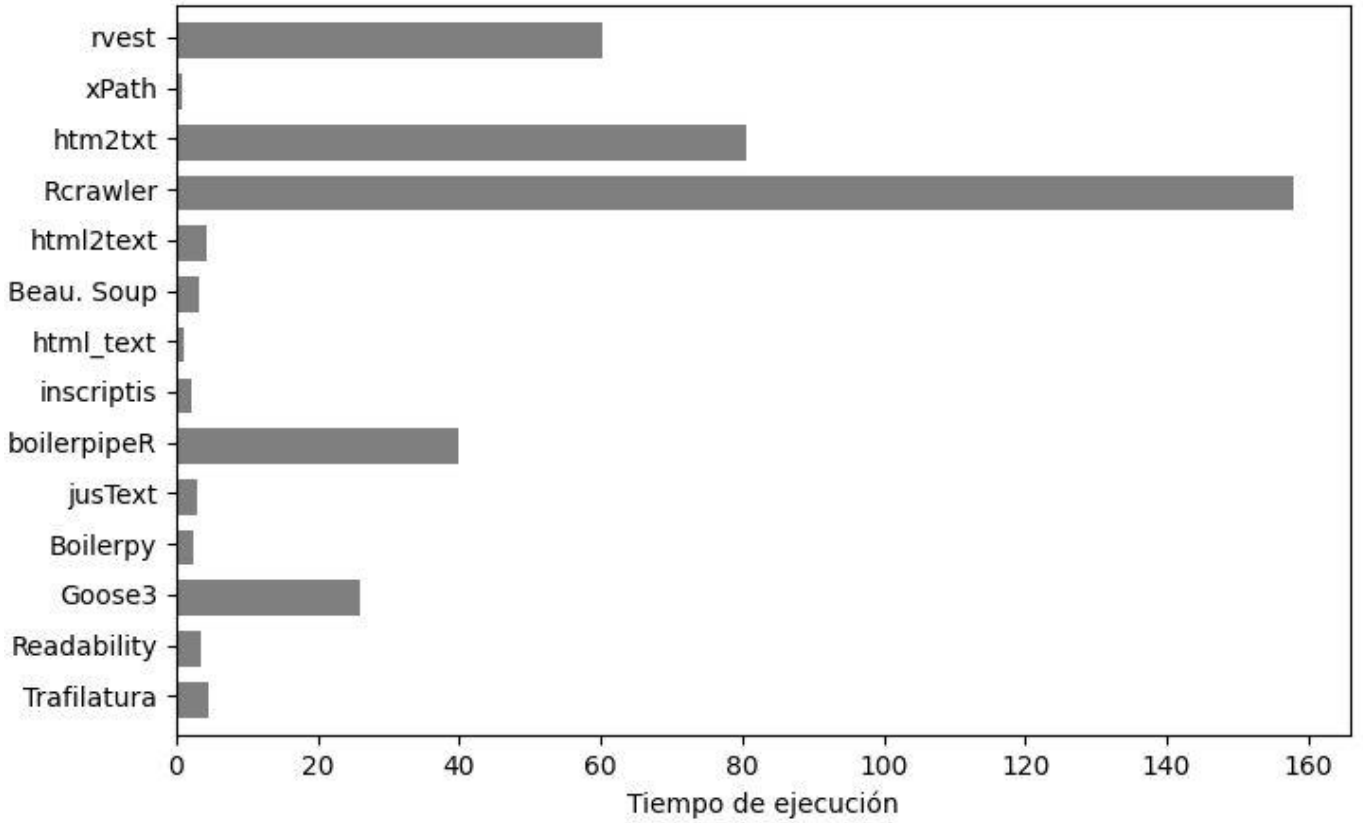
Optimización de la extracción RAM-CPU

	Trafilatura	Readability	Goose3	Boilerpy	jusText	BoilerpipeR	inscriptis
Uso RAM(%)	45.9	45.3	32.2	43.9	45.1	47.9	45.0
Uso CPU(%)	1.4	1.6	6.1	1.9	0.5	2.6	0.2

	html_text	BeautifulSoup	html2text	Rcrawler	htm2txt	XPath	rvest
Uso RAM(%)	44.9	42.0	44.6	46.7	46.7	44.4	44.1
Uso CPU(%)	0.5	1.2	1.8	3.4	2.0	2.0	8.9

Resultados obtenidos

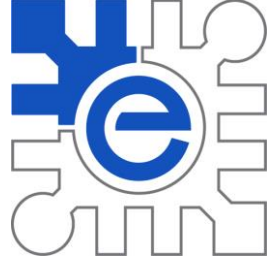
Tiempos de ejecución



Índice general

Conclusiones

Trafilatura es el paquete cuya extracción resultante presenta una similitud más cercana a lo que un usuario vería en el documento HTML convencional. Lo resultados demuestran que este tipo de paquetes están muy cerca de lo esperado.



Muchas gracias por
su atención