Universidad de La Habana
Facultad de Matemática y Computación

Desarrollo de una solución adaptativa para web scrapers en respuesta a cambios estructurales en páginas web.

Autor:
Jan Carlos Pérez González

Tutor:

Carlos León

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Fecha

github.com/username/repo

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

Índice de figuras

Ejemplos de código

Introducción

La World Wide Web (WWW) surgió con el objetivo de proporcionar una interfaz unificada y fácil de usar para acceder a información en diferentes formatos, plataformas y protocolos [1]. Desde su creación y popularización en la década de 1990, la WWW ha sido protagonista de transformaciones significativas en la vida de las personas. El acceso democratizado a la información, impulsado en gran parte por los motores de búsqueda, generó un crecimiento exponencial en la cantidad de datos disponibles. Para ponerlo en perspectiva, entre 2010 y 2023, el volumen global de datos creados, copiados y consumidos aumentó de 2 Zettabytes (2000 Terabytes) a 123 Zettabytes [2]. Con este incremento masivo del tráfico y la información en la Web, surgió la necesidad de organizar las páginas existentes dentro de los motores de búsqueda. Para lograr este objetivo, se crearon los Web Crawlers: software capaz de rastrear, indexar y catalogar páginas web de manera automatizada [3]. El primer ejemplo conocido de un Web Crawler apareció en 1993 con el motor de búsqueda JumpStation, mediante una herramienta denominada World Wide Web Wanderer Offsite Link (conocida históricamente como The Wanderer), la cual fue utilizada para medir el tamaño real de Internet [4] [5]. En 2004, la biblioteca BeautifulSoup revolucionó la extracción de datos al facilitar el análisis y la extracción de contenido HTML, optimizando así el tiempo de trabajo de los programadores. Posteriormente, el desarrollo del software Web Integration Platform 6.0, creado por Stefan Andresen, permitió que usuarios sin conocimientos de programación pudieran seleccionar visualmente datos de páginas web y organizarlos en formatos estructurados, como archivos Excel o bases de datos. Esto marcó un avance importante en la accesibilidad y utilidad del web scraping [6].

En la actualidad, el web scraping se ha consolidado como una herramienta fundamental para la adquisición y análisis de datos a gran escala, abarcando una amplia variedad de aplicaciones en distintos sectores. En el ámbito empresarial, se utiliza para el análisis de tendencias de mercado, la evaluación del alcance de productos y el estudio de la competencia. En el campo científico, facilita la recopilación automatizada de grandes volúmenes de datos para investigaciones y análisis posteriores. Asimismo, su aplicación en redes sociales permite realizar estudios de comportamiento y análisis

sociales sobre temas específicos, proporcionando información valiosa para entender dinámicas sociales y opiniones públicas [7]. No obstante, el uso generalizado de estas herramientas también ha revelado una vulnerabilidad significativa en su funcionamiento: su desempeño depende estrictamente de las estructuras y etiquetas definidas en el código HTML de las páginas web, lo que los hace sensibles a cualquier cambio en la estructura de los sitios web. En este contexto, uno de los métodos más utilizados en el web scraping para identificar y extraer elementos específicos de las páginas web es XPath (XML Path Language). XPath es un lenguaje diseñado originalmente para navegar por estructuras XML, pero también es ampliamente empleado en documentos HTML debido a su capacidad para localizar nodos y elementos específicos en un árbol estructurado de datos. Mediante expresiones XPath, los scrapers pueden dirigirse de manera precisa a ciertos elementos, como tablas, imágenes o párrafos, en función de su posición o atributos dentro del documento HTML. Sin embargo, la dependencia de los scrapers en expresiones XPath bien definidas resalta su vulnerabilidad inherente: cualquier modificación en la estructura del HTML subyacente (como un cambio en las etiquetas, clases o jerarquías) puede invalidar las rutas XPath utilizadas, resultando en errores de extracción. Este desafío representa un problema crucial, ya que incluso modificaciones menores en el diseño de una página o en la disposición de sus elementos pueden hacer que los web scrapers dejen de funcionar correctamente. Dichos cambios pueden provocar errores en la extracción de datos, pérdida de información relevante y, en algunos casos, la necesidad de rediseñar las rutas XPath o los scrapers desde cero. Este problema técnico se ve acentuado por el constante cambio dinámico de las páginas web, lo que genera una falta de estabilidad para las herramientas de scraping y plantea retos adicionales en su aplicación a largo plazo [8].

La motivación principal de esta investigación es mejorar la robustez y la eficiencia de los web scrapers, desarrollando soluciones que mitiguen su dependencia de las estructuras HTML fijas. A través de este trabajo, se busca garantizar que los web scrapers sean capaces de adaptarse de manera efectiva a los cambios estructurales de las páginas web sin comprometer su rendimiento, y que puedan seguir cumpliendo su función en la adquisición de datos a gran escala en entornos dinámicos y cambiantes.

El objetivo principal de este trabajo es desarrollar un sistema de web scraping robusto y adaptable que minimice su dependencia de las estructuras y etiquetas fijas del código HTML. El sistema propuesto permitirá a los scrapers ajustarse automáticamente a cambios en la estructura de las páginas web, asegurando un funcionamiento continuo y eficiente. Para lograrlo, se analizarán las causas más frecuentes de fallos en los scrapers actuales y se evaluarán las técnicas existentes en su capacidad de adaptación a entornos dinámicos. Posteriormente, se desarrollará un modelo adaptativo que pueda identificar y ajustarse a cambios en el HTML de manera automática. Final-

mente, el modelo será implementado y validado a través de pruebas experimentales en sitios web con estructuras cambiantes, comparando su desempeño con herramientas tradicionales en términos de robustez, precisión y eficiencia. Con esto, se busca aportar una solución más resiliente y eficiente para la adquisición de datos en un ecosistema web en constante evolución.

En este sentido, la presente investigación plantea la siguiente hipótesis: Ün modelo de aprendizaje automático mejora significativamente la capacidad de los web scrapers para adaptarse a cambios estructurales en las páginas web, reduciendo errores y mejorando la eficiencia operativa..Esta afirmación parte de la premisa de que la implementación de técnicas de aprendizaje automático puede dotar a los scrapers de mecanismos adaptativos que les permitan reconocer y ajustarse automáticamente a las modificaciones en la estructura HTML de los sitios web, garantizando un funcionamiento más robusto y eficiente en entornos dinámicos. A través de esta hipótesis, se busca desarrollar y validar un modelo que optimice la resiliencia de los web scrapers, superando las limitaciones de las herramientas tradicionales y contribuyendo al avance de soluciones más eficientes para la adquisición de datos a gran escala.

La metodología de este trabajo se centra en desarrollar un modelo adaptativo para mejorar la eficiencia de los web scrapers frente a cambios estructurales en las páginas web. El proceso consta de cuatro etapas clave:

- Construcción del Dataset: Se recopilarán versiones históricas de páginas web, etiquetando los XPaths correspondientes a elementos clave de cada versión para estudiar las modificaciones en la estructura HTML a lo largo del tiempo.
- Fine-Tuning de un Modelo de Lenguaje: Se utilizará un modelo de lenguaje preentrenado, adaptado para entender HTML, mediante el entrenamiento con datos etiquetados para capturar la estructura y relaciones dentro del código HTML.
- Desarrollo del Modelo Adaptativo: El modelo será ajustado para recibir como entrada el HTML antiguo, el XPath correspondiente a ese HTML y el HTML nuevo. La salida del modelo será el XPath actualizado, que represente el mismo elemento en la nueva estructura del HTML.
- Validación del Modelo: Se evaluará el rendimiento del modelo utilizando un conjunto de datos de prueba, comparándolo con métodos tradicionales de web scraping en términos de precisión, robustez y eficiencia.

Este pipeline busca crear un sistema de web scraping más robusto y flexible, capaz de adaptarse a los cambios continuos de las páginas web y reducir los errores asociados.

Este documento está estructurado en tres capítulos principales que cubren desde los fundamentos teóricos hasta los resultados experimentales y las conclusiones de la investigación. En el Capítulo 1, se aborda el marco teórico y la revisión bibliográfica, proporcionando un análisis detallado de los conceptos clave relacionados con el web scraping, los modelos de lenguaje como BERT y sus aplicaciones en la adaptación a cambios estructurales en las páginas web. El Capítulo 2 describe los detalles de la implementación, presentando el pipeline propuesto, que incluye la construcción de un dataset histórico, el entrenamiento del modelo de lenguaje ajustado a la comprensión de HTML y la validación experimental del sistema desarrollado para actualizar rutas XPath. Finalmente, el documento concluye con el Capítulo 3, donde se exponen las conclusiones más relevantes del trabajo y se presentan recomendaciones para investigaciones futuras, con el objetivo de extender y mejorar los métodos desarrollados.

Capítulo 1

Propuesta

Capítulo 2

Detalles de Implementación y Experimentos

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones

Bibliografía

[1] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen y A. Secret, «The world-wide web,» en, *Commun. ACM*, vol. 37, n.º 8, págs. 76-82, 1994 (vid. pág. 1).

[2] *Data growth worldwide 2010-2028*, en, `https://www.statista.com/statistics/871513/worldwide-data-created/`, Accessed: 2024-12-18 (vid. pág. 1).

[3] S. M. Mirtaheri, M. E. Dinçktürk, S. Hooshmand, G. V. Bochmann, G.-V. Jourdan e I. V. Onut, «A brief history of web crawlers,» 2014. eprint: `1405.0749` (vid. pág. 1).

[4] B. Data, *The history of web scraping and what the future holds*, en, `https://bdaily.co.uk/articles/2022/04/25/the-history-of-web-scraping-and-what-the-future-holds`, Accessed: 2024-12-18, abr. de 2022 (vid. pág. 1).

[5] I. S. H. Almaqbali, F. M. A. Al Khufairi, M. S. Khan, A. Z. Bhat e I. Ahmed, «Web Scrapping: Data extraction from websites,» *J Stud Res*, 2020 (vid. pág. 1).

[6] *Brief history of web scraping*, en, `https://webscraper.io/blog/brief-history-of-web-scraping`, Accessed: 2024-12-18 (vid. pág. 1).

[7] *Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Applications* (vid. pág. 2).

[8] B. V. S. Ujwal, B. Gaind, A. Kundu, A. Holla y M. Rungta, «Classification-based adaptive web scraper,» en *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2017 (vid. pág. 2).