Guillermo Rojo
Análisis informatizado de textos.
Universidade de Santiago de Compostela, 2023 (238 pp.), ISBN 9788419679529

Review of Guillermo Rojo
Análisis informatizado de textos.
Universidade de Santiago de Compostela, 2023 (238 pages), ISBN 9788419679529

Lorena M. A. de- Matteis
(ICIC-CONICET/ Universidad Nacional del Sur)
https://orcid.org/0000-0003-3683-1722

Aunque presenta sus desafíos, la potencialidad de la *línea de comandos* suele ser destacada para distintas ciencias exactas y naturales, que pueden aprovecharla para simplificar el tratamiento de amplios conjuntos de datos, repitiendo tareas en múltiples archivos o, incluso, automatizándolas.[1] La demostración de similares beneficios para filólogos y lingüistas es la materia del académico y profesor emérito Guillermo Rojo en *Análisis informatizado de textos*, texto que complementa de manera eficaz su *Introducción a la lingüística de corpus en español* (2021). Editado en formato digital por la Universidad de Santiago de Compostela, este nuevo libro propone una presentación de diversas estrategias para tratar de manera digital materiales textuales, cuestión que en los departamentos y facultades de Humanidades y Ciencias Sociales suele recibir un abordaje no sistemático o extracurricular en el nivel de grado e, incluso, en el de posgrado. Los recursos y procedimientos que se presentan a lo largo de los seis capítulos centrales no exigen una formación informática previa para ser implementados, y la claridad con que se los ejemplifica es un reflejo directo de la experticia del autor en el aprovechamiento de la tecnología informática en los estudios sobre el lenguaje y la lingüística de corpus. La organización del volumen incluye, además, un breve apéndice de los comandos ilustrados en el texto, seguido de las referencias a los recursos electrónicos mencionados, la bibliografía y, por último, un índice temático.

Además de situar al lector en el contexto del evidente impacto que el desarrollo de las computadoras ha tenido sobre los estudios lingüísticos, en particular en los campos de la lexicografía y la lingüística de corpus, la "Introducción" establece una vinculación inicial entre las herramientas que se abordarán y el campo de las ciencias de datos, que se ocupa de la obtención, depuración, exploración y modelado de datos textuales, como así también de la interpretación de los resultados. El objetivo central del autor en este libro se vincula con las primeras tres de estas tareas, esto es, busca mostrar modos para extraer, modificar y sistematizar datos textuales empleando las herramientas propias de un sistema operativo, a cuya ejecución se accede desde la *línea de comandos*, una interfaz que permite ejecutar instrucciones en un sistema operativo para realizar y controlar funciones diversas. En otras palabras, la propuesta radica en tratar los materiales textuales no a través de programas adicionales –entre los que pueden mencionarse, a título ilustrativo, los conocidos *WordSmith Tools, AntConc* o *FreeLing*–, sino utilizando recursos adicionales del sistema operativo (o integrables a este), en particular los de un sistema *Unix/Linux*.

De acuerdo con este objetivo, la "Introducción" considera una serie de aspectos prácticos que deben tenerse en cuenta al encarar este tipo de tareas. En primer lugar, el autor expone y sistematiza los problemas asociados con los distintos formatos de archivos (o "ficheros") textuales (incluyendo, por caso, distintos formatos de texto, hojas de cálculo, páginas *web*) cuya conversión a texto plano resulta necesaria para aprovechar los comandos (u "órdenes") que se abordan en el volumen. En este sentido, también resulta pertinente y clara la revisión de los distintos sistemas de codificación electrónica de textos (ASCII, ASCII ampliado, UTF, SGML/XML, TEI), en tanto el formato de codificación puede tener un impacto sobre las decisiones que deben adoptarse en el tratamiento de los materiales, en especial, cuando estos obedecen a distintas etapas en la historia de una lengua.

El autor expone en este mismo capítulo la distinción entre codificación *no lingüística* (diferenciando, a su vez, entre la *textual/intratextual* y la de *metadatos/extratextual*) y *lingüística* (léxica, morfológica, sintáctica, semántica o pragmática) que, siempre en función de los fines perseguidos, pueden realizarse en un formato específico (las denominadas *etiquetas*) para que las aplicaciones puedan recuperar y procesar este tipo de información para el posterior análisis. La inclusión de ejemplos de los distintos tipos de codificación ilustra sus respectivas ventajas y desventajas y conecta los contenidos de este volumen con los terrenos de la lingüística de corpus y de la lingüística computacional.

Desde el punto de vista práctico, por último, la "Introducción" también discute brevemente las diferencias entre un *procesador* y un *editor* de textos. Esta distinción es operativa en el marco del volumen, en tanto el segundo tipo de programa facilita la edición de código. Así, un breve detalle de las funciones que deberían guiar la selección de un editor apropiado evidencia la doble intención del capítulo de situar al lector en relación con la temática y de prepararlo, en términos prácticos, para implementar los ejemplos ofrecidos y comenzar a editar código.

En "El trabajo desde el terminal", el primer capítulo dedicado al aprovechamiento concreto de la línea de comandos, Rojo discute, en primer lugar, las diferencias que existen entre trabajar de esta manera y en entornos gráficos, con los que la mayoría de los usuarios están familiarizados. Aunque queda clara la posibilidad de utilizar algunas herramientas análogas a las que expondrá en sistemas como *Windows* o *MacOS*, el foco recae sobre aquellas desarrolladas en los sistemas *Unix/Linux*. La primera parte del capítulo se concentra, entonces, en la preparación de la computadora para poder emplear estos recursos –en especial cuando se cuenta con un sistema *Windows*– y en cómo acceder luego a la línea de comandos.

Como primera aproximación, el autor explica algunos de los comandos básicos para desempeñarse en el entorno *Unix*, por ejemplo, *mkdir* (crea directorios), *ls* (muestra su contenido), *cd* (cambia el directorio de trabajo), *echo* (imprime texto en pantalla o redirige contenido a un archivo determinado), *cat* (lee o concatena archivos), *cp* (copia archivos/directorios), *mv* (mueve/renombra), *rm* (elimina), entre otros. También se presentan otros más específicos para el tratamiento de los materiales textuales, como *seq* (genera secuencias de números), *wc* (cuenta palabras, caracteres, *bytes*), u otras aplicaciones específicas de *cat*. En todos los casos, como ocurre también en los restantes capítulos, cada orden se ilustra partiendo de una expresión prototípica, modelo sobre el que se van elaborando otras más complejas que, progresivamente, permiten refinar las operaciones.

DOI: https://doi.org/10.56683/rs252196

Ya desde esta presentación inicial se destaca la relevancia del comando *grep* (empleado, básicamente, para buscar patrones de texto), que en combinación con otros como *sed* (buscar, reemplazar, insertar o eliminar secuencias de texto en archivos) o *sort* (ordena líneas de texto), poseen un rol central en la recuperación, modificación y ordenamiento de datos textuales y que serán abordados a partir de múltiples ejemplos prácticos en el resto del volumen. Así, en el siguiente capítulo, "La exploración y explotación del texto electrónico", el autor focaliza la atención sobre el ya mencionado comando *grep*, ilustrando una serie de operaciones, de complejidad creciente. Basadas en *grep* y refinadas con distintas estrategias, se exponen los comandos que pueden emplearse para recuperar una secuencia determinada de caracteres en un archivo de texto plano, para lograr que los resultados sean indiferentes al uso de mayúsculas o minúsculas, para introducir distintas opciones de cuantificación o restricciones de posición del elemento buscado, entre otras operaciones de interés para el procesamiento de textos. Un apartado puntual del capítulo se dedica a demostrar en mayor detalle la utilidad de *cat*, *sort* y *cut* (extrae secciones específicas de un archivo), además de *grep*, para aprovechar archivos de frecuencias léxicas, mientras que las operaciones de recuperación necesarias para generar listas de palabras se exploran de manera detallada en otra sección, que introduce a los lectores en la relevancia del encadenamiento de órdenes (*pipes*) para obtener los resultados deseados.

Las estrategias para modificar archivos de texto se exploran en "Modificación de datos", capítulo que ilustra los comandos para realizar la sustitución o la adición de líneas de texto para adaptar materiales en función, por ejemplo, de posibles dificultades con su transformación a texto plano o para adecuarlos a los objetivos específicos de una investigación determinada. En este caso, las órdenes que se demuestran incluyen el empleo de *tr* (traduce una secuencia de caracteres en otra) y *sed* (sustituye caracteres o secuencias), cuyas ventajas y desventajas se cotejan. En este capítulo se introducen también las *banderas* (*flags*), expresiones que permiten modificar el comportamiento y alcance de las diversas órdenes.

Las *expresiones regulares* se abordan en el capítulo homónimo, que conecta naturalmente con el tercero. En este caso, Rojo expone con complejidad progresiva la manera de construir este tipo de formulaciones abstractas que se emplean junto a *grep* y otras órdenes adicionales para realizar búsquedas precisas y exhaustivas. Se trata de patrones construidos a partir de considerar la posible aparición de caracteres en una expresión, los rangos de caracteres de interés –alfabéticos [a-z], numéricos [0-9] o alfanuméricos–, las posiciones en las que estos caracteres pueden aparecer en la expresión de búsqueda, la cantidad de ocurrencias de cada tipo de elemento (exacta, mínima o máxima), entre otros posibles aspectos. Su importancia, entonces, radica en la capacidad de facilitar búsquedas sistemáticas de patrones muy precisos en grandes volúmenes de texto, lo que facilita su recuperación, cuantificación, ordenamiento o eventual alteración.

El último capítulo, "Ampliación: comandos, utilidades y opciones adicionales", profundiza sobre algunas de las cuestiones ya tratadas. Así, retoma en su primer apartado las dificultades que entraña la conversión de distintos formatos de archivo a texto plano ya aludida desde la introducción, explorando con mayor detalle algunos de los resultados problemáticos que suelen acompañar dichos procesos (por ejemplo, la inserción de saltos en los finales de línea) y algunas de las estrategias para subsanar estas dificultades. En otro apartado, este capítulo

añade opciones adicionales para trabajar con las órdenes *grep* y *sed*, mientras que un último apartado introduce el comando *awk*, que también puede ser considerado como un lenguaje en sí mismo por las múltiples posibilidades que ofrece para tareas tales como seleccionar texto específico, filtrarlo, realizar cálculos, entre otras.

Si bien el texto avanza en todos los casos a través de la descripción e ilustración del empleo de las diversas órdenes, tanto independientes como encadenadas, para resolver distintas operaciones de creciente dificultad que se proponen sobre fragmentos de texto literario o ficheros de frecuencias léxicas tomados de corpus en línea del español, abunda también en la propuesta de ejercicios de práctica independiente. Ubicados en la mayoría de los casos al final de cada capítulo, estas propuestas replican en alguna medida las operaciones mostradas y las complejizan gradualmente, desafiando a los lectores a ganar experiencia en la resolución de problemas típicos que se enfrentan al abordar el tratamiento digital de materiales textuales. Si bien las resoluciones posibles no están en el texto, puede considerarse que, si se han puesto en práctica los ejemplos ofrecidos por el autor y se han logrado obtener análogos resultados, su nivel de dificultad no resulta excesivo. Además, la existencia de recursos en línea alternativos para acompañar el proceso de aprendizaje en torno a la línea de comandos (tutoriales en video, plataformas para iniciarse en programación, sistemas de inteligencia artificial generativa, entre otros), junto con las referencias bibliográficas que el volumen ofrece, la ausencia de resoluciones a los ejercicios de práctica no debería constituir un obstáculo insalvable para los lectores. Por el contrario, puede proponerse que constituye un acierto en la medida en que alienta el desarrollo de la autonomía en el empleo de estas herramientas.

En cuanto a los apéndices, la utilidad del que enumera los comandos mencionados en los diversos capítulos resulta relativa, pues aparecen también en el índice temático y no van acompañados ni de referencias a las secciones ni de, por ejemplo, una breve descripción de su función que, a modo de referencia rápida, justifique este anexo. En cambio, la sistematización independiente de los *url* correspondientes a los recursos electrónicos mencionados facilita el acceso a recursos mencionados en el texto.

La experticia del autor con la temática se percibe en la claridad didáctica del volumen, que incluye múltiples orientaciones generales de índole práctica, como, por ejemplo, la precisión del tipo de codificación más habitual en el entorno lingüístico y computacional –UTF-8– o el uso de archivos con valores separados por tabuladores –*tsv*– en lugar de comas –*csv*–. En el mismo sentido, las notas no solo discuten y problematizan algunos de los contenidos, sino que también añaden precisiones que tienen especialmente en cuenta un espectro de lectores más o menos inexperto en técnicas computacionales.

Tal como apunta la introducción, no se le escapa al autor que el empleo de los comandos considerados resultará un desafío mayor a los lingüistas y humanistas que están acostumbrados a trabajar con sistemas gráficos. No obstante, la gradualidad en la exposición de los comandos, el retomarlos en distintos lugares para mostrar cómo encadenarlos, y la propuesta de oportunidades de ejercitación, garantizan que los lectores de la obra puedan utilizarla como un *vademécum* hasta lograr dominar los distintos procedimientos. Sean estudiantes de grado del siglo XXI de carreras humanísticas o sociales, jóvenes egresados que se inician en tareas de investigación o cuya labor profesional requiere del tratamiento digital

de datos textuales, o investigadores ya formados que buscan incorporar nuevas prácticas de investigación, los lectores sin experiencia en técnicas computacionales encontrarán en esta obra una orientación práctica para lograr sus objetivos.

Notas

[1] Para acceder a un panorama simple de algunas de sus posibles aplicaciones, cfr. Perkel (2021), mientras que un tratamiento especializado de la línea de comandos puede encontrarse en la última edición del volumen de Janssens (2021), que actualiza la referencia ofrecida por Rojo.

Referencias

Janssens, J. (2021). *Data Science at the Command Line* (2da. Ed.). O'Reilly.

Perkel, J. (2021). Five reasons why researchers should learn to love the command line. *Nature*, 590, 173-174. https://doi.org/10.1038/d41586-021-00263-0

Rojo, G. (2021). *Introducción a la lingüística de corpus en español.* Routledge. https://doi.org/10.4324/9781003119760