Python per archivisti e bibliotecari

Uno strumento per rendere semplici le attività complicate

DANILO DEANA
Università degli studi di Milano
danilo.deana@unimi.it

Sul numero di "Code4Lib" del 10 agosto 2020 Katharine Frazier ha pubblicato un articolo in cui annunciava che le biblioteche della North Carolina State University avevano sviluppato uno strumento per automatizzare la generazione dei *report* mensili in GOBI, la piattaforma messa a loro disposizione dal fornitore per effettuare gli ordini di libri a stampa e elettronici. Dato che GOBI non dispone di un'Application Programming Interface, i bibliotecari erano costretti a svolgere ricerche titolo per titolo che richiedevano fino due giorni al mese di lavoro. Grazie a un programma scritto in Python è stato possibile automatizzare questa attività e raccogliere dati per centinaia di titoli in meno di mezz'ora, liberando tempo per altri progetti.[1]

Python è un linguaggio di programmazione di alto livello progettato da Guido van Rossum mentre lavorava al Centrum Wiskunde & Informatica di Amsterdam e reso disponibile per la prima volta nel 1991 (la versione attuale è la 3.8 del 2019). Il nome deriva dalla serie *Monty Python's Flying Circus* trasmessa dalla British Broadcasting Corporation dal 1969 al 1974.

Oggi, lo sviluppo del linguaggio è affidato alla Python Software Foundation, un'organizzazione senza scopo di lucro che si occupa del progresso della tecnologia *open source* relativa a Python. In particolare, la Python Software Foundation produce la distribuzione di Python e la rende disponibile gratuitamente. Ciò include il linguaggio stesso, le sue librerie, la documentazione standard, i programmi di installazione, il codice sorgente, i materiali didattici e diversi strumenti e applicazioni. Essa definisce anche le licenze d'uso, garantendo a tutti il diritto di ottenere, utilizzare, ridistribuire e modificare liberamente la proprietà intellettuale detenuta dalla fondazione stessa.[2]

Python è un linguaggio facile da utilizzare, ma estremamente potente, grazie anche alle numerose librerie di funzioni sviluppate per specifiche esigenze.

In Python non esiste una fase di compilazione separata che generi un file eseguibile partendo dal codice sorgente (semplici file di testo con estensione .py). Questa caratteristica lo rende facilmente portabile. Una volta scritto il codice, infatti, questo può essere eseguito all'interno della maggior parte dei sistemi operativi attualmente in uso: è sufficiente la presenza della versione corretta dell'interprete.

Python può essere utilizzato non solo per automatizzare procedure che devono essere svolte a scadenze regolari, come nel caso delle biblioteche della North Carolina State University. Di seguito è presentata una serie di studi pubblicati negli ultimi cinque anni che hanno fatto uso di linguaggi di programmazione e delle librerie di funzioni ad essi associate per svolgere attività di natura biblioteconomica o archivistica.

Jannette L. Finch e Angela R. Flenner hanno utilizzato delle librerie grafiche per generare una serie di

diagrammi sulle diverse sezioni in cui è divisa la collezione cartacea della biblioteca del College of Charleston nella South Carolina. I diagrammi permettono di rappresentare contemporaneamente, oltre ai dati sulle sezioni, anche quelli sulle spese, sul numero dei corsi e sugli studenti iscritti e hanno fornito un ottimo punto di partenza per discutere su possibili squilibri nella collezione. Nelle intenzioni delle autrici la metodologia utilizzata dovrebbe offrire un modello da seguire per tutti coloro che desiderano esaminare la loro collezione e può rivelarsi utile per distribuire meglio i fondi, suggerire opportunità per nuovi servizi o per pubblicizzare settori della collezione nascosti prima dell'analisi grafica.[3]

Kiri Lou Wagstaff ha dedicato una tesi di dottorato allo scarto bibliografico. Gli studi hanno dimostrato che lo scarto bibliografico (l'eliminazione selettiva di documenti non utilizzati, usurati, obsoleti o non rilevanti) avvantaggia gli utenti e aumenta i tassi di circolazione. Tuttavia, il tempo necessario per esaminare la collezione e decidere cosa scartare rappresenta un ostacolo spesso insormontabile. Utilizzando un insieme di dati provenienti dalla Wesleyan University di Middletown nel Connecticut, Wagstaff ha realizzato una serie di modelli per individuare i candidati allo scarto. A questo scopo ha impiegato Python e una libreria di funzioni specificamente progettata per l'apprendimento automatico: scikit-learn.[4]

Ana Maria B. Fresnido e Agnes S. Barsaga hanno utilizzato uno strumento per l'analisi del registro delle transazioni allo scopo di determinare le ragioni dei tassi di fallimento delle ricerche svolte nel catalogo dell'università De La Salle di Manila. I registri delle transazioni annotano le interazioni tra i sistemi di recupero delle informazioni e gli utenti. I risultati dell'analisi hanno mostrato che gli utenti ignorano i limiti del catalogo, che il loro modo di cercare i documenti è lo stesso di quello che utilizzano in Google e che, presumibilmente, si aspettano che il catalogo restituisca risultati simili a quelli del motore di ricerca.[5]

Rachael A. Cohen e Angie Thorpe Pusnik hanno svolto un'analisi di questo tipo sui registri delle transazioni di un Web Scale Discovery Service. Nell'articolo gli autori spiegano anzitutto come hanno raccolto le *query* di ricerca dai registri delle transazioni. Dopo averle categorizzate secondo il sistema di classificazione della Library of Congress, le hanno esaminate utilizzando strumenti di analisi del testo. I risultati della ricerca hanno permesso di comprendere che cosa docenti e bibliotecari dovrebbero comunicare agli studenti per migliorarne l'*information literacy*.[6]

Python può essere impiegato anche per analizzare i registri delle transazioni dell'applicazione utilizzata da moltissime biblioteche per rendere possibile ai loro utenti connettersi alle risorse elettroniche cui la biblioteca stessa fornisce l'accesso in modo semplice ed efficace da qualsiasi postazione e in qualsiasi momento: EZproxy dell'Online Computer Library Center. Nonostante EZproxy disponga di strumenti per l'analisi dei registri, utilizzandoli non è sempre facile ottenere dati significativi. Brighid M. Gonzales, bibliotecaria presso la Our Lady of the Lake University di San Antonio in Texas, ha scritto un articolo in cui descrive come ha utilizzato Python e le librerie create appositamente per l'analisi dei dati per esaminare i registri di EZproxy e ottenere dati più precisi sul modo in cui viene utilizzata l'applicazione.[7]

Gli strumenti per l'analisi del testo sono stati utilizzati anche da Sultan M. Al-Daihani e Alan Abrahams per analizzare i tweet di dieci biblioteche accademiche. I risultati, oltre a mostrare la differenza tra le biblioteche nella distribuzione dei tweet nel corso del tempo, hanno messo in evidenza come la parola usata più frequentemente sia "open", usata in una varietà di contesti, il bigramma (la sequenza di due parole) "special collections" e il trigramma (la sequenza di tre parole) "save the date". I risultati evidenziano anche l'importanza di utilizzare approcci di *data mining* e *text mining* per aiutare nella pianificazione strategica delle attività volte a sensibilizzare gli utenti e a pubblicizzare i servizi.[8]

Svein Arne Brygfjeld, Freddy Wetjen e André Walsøe hanno descritto gli esperimenti condotti dalla Biblioteca nazionale norvegese per studiare la classificazione automatica degli articoli basata sul sistema di classificazione decimale Dewey. Sono state utilizzate varie piattaforme e modelli per l'apprendimento automatico. I risultati indicano che l'apprendimento automatico è un ambiente adatto per la classificazione semi-automatizzata o completamente automatizzata. Inoltre, è stato possibile dimostrare che la creazione delle piattaforme di apprendimento automatico può essere realizzata anche utilizzando documenti artificiali.[9]

Gregory Wiedeman, che opera presso il Mary Elisabeth Grenander Department of Special Collections

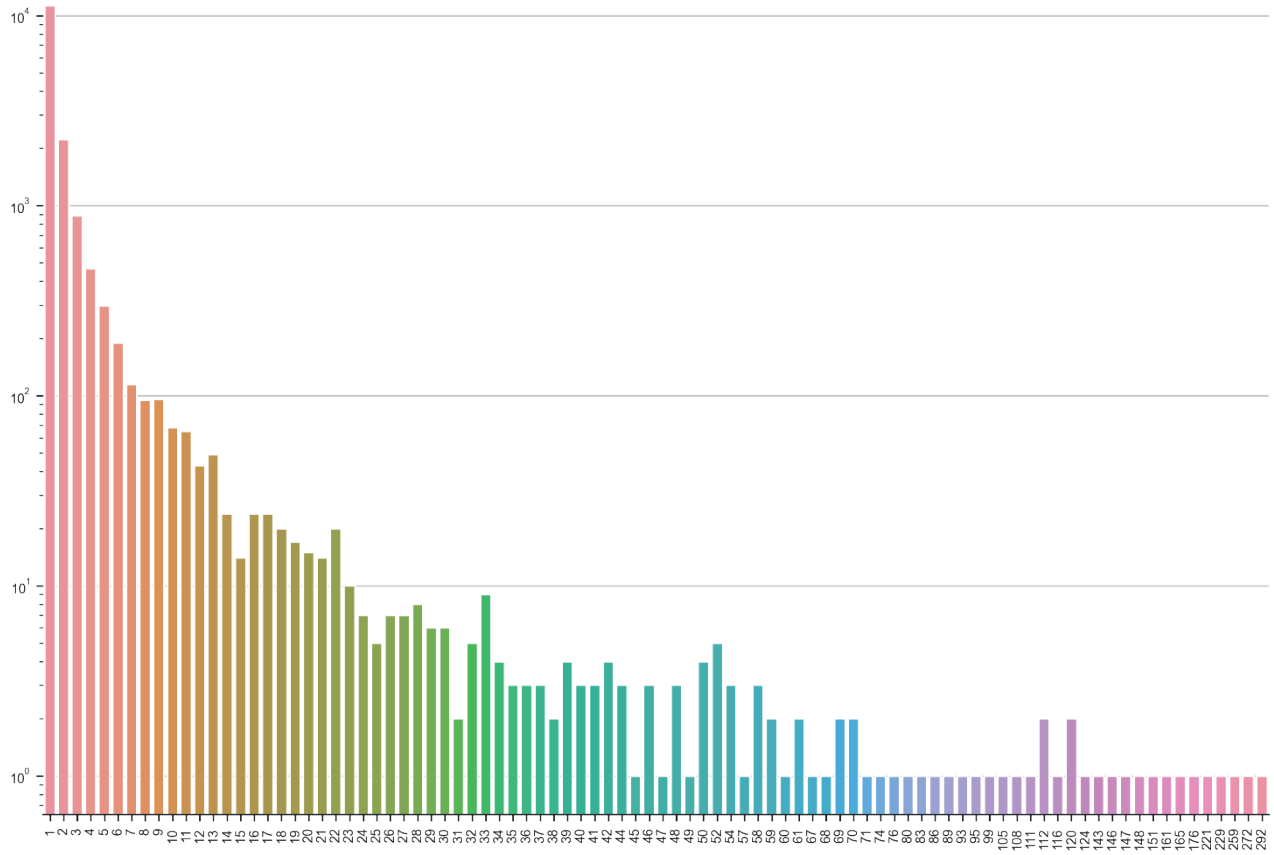

Grafico 1

& Archives dell'Università di Albany, sostiene che una parte importante del lavoro di un archivista consista oggi nella progettazione di flussi di lavoro automatizzati. Sono infatti disponibili molte applicazioni aperte e interoperabili ed è necessario essere in grado di acquisire e trasferire dati tra questi diversi sistemi. Python è lo strumento ideale per svolgere queste attività. La sua sintassi è relativamente semplice e intuitiva, è estremamente ben documentato, con un'ampia base di utenti e ha un numero enorme di librerie ottimizzate per risolvere quasi tutti i problemi che si presentano.[10]

Come si vede, gli ambiti di applicazioni di un linguaggio di programmazione all'interno di una biblioteca o di un archivio sono molto ampi e permettono di ottenere risultati che non potrebbero essere raggiunti in altro modo. Le caratteristiche di Python, tra cui quella di essere *open source* e di proporsi come uno strumento in grado di trasformare chiunque in un programmatore,[11] lo rendono ideale per questo tipo di utilizzi.

Nel diagramma che segue, ottenuto utilizzando una libreria per l'analisi dei dati (*pandas*) e due librerie grafiche (*matplotlib* e *seaborn*) di Python, è ad esempio rappresentata la distribuzione delle registrazioni bibliografiche relative ai documenti conservati presso la Biblioteca di Filosofia dell'Università degli studi di Milano per il numero di soggetti inseriti all'interno delle registrazioni stesse. La distribuzione è caratterizzata da un'asimmetria positiva così accentuata che è impossibile rappresentarla se non attraverso un diagramma a barre che utilizzi per l'asse verticale una scala logaritmica. Per chi non avesse familiarità con i logaritmi, ricordiamo che 10^0 vale 1, 10^1 10, 10^2 100, 10^3 1.000 e 10^4 10.000 (Grafico 1).

Su 16.220 soggetti utilizzati, 11.279 sono presenti in una sola registrazione, 2.227 in due registrazioni, 886 in tre registrazioni e così via.

Dato che ho utilizzato Phyton per due articoli sulle collezioni e il loro utilizzo pubblicati in "Biblioteche oggi" (*Il dovere di contare*, *Le collezioni messe a nudo*) e

per un libro intitolato *Come valutare la qualità del catalogo di una biblioteca*, con Editrice Bibliografica abbiamo deciso di realizzare un webinar attraverso cui illustrare le caratteristiche principali del linguaggio e favorirne la diffusione, impiegando a questo scopo registrazioni bibliografiche tratte da un catalogo e descrizioni archivistiche. L'esempio è quanto fatto da Library Carpentry con il corso *Python for Librarians*.[12]

NOTE

[1] Katharine Frazier, *Automated Collections Workflows in GOBI: Using Python to Scrape for Purchase Options*, "Code4Lib", 49 (2020), https://journal.code4lib.org/articles/15334.

[2] L'indirizzo del sito della fondazione è python.org.

[3] Jannette L. Finch, Angela R. Flenner, *Using Data Visualization to Examine an Academic Library Collection*, "College & Research Libraries", 2016, p. 765-778.

[4] Kiri Lou Wagstaff, *Automated classification to improve the efficiency Of weeding library collections*, thesis presented to the Faculty of the School of Information, San José State University, Maggio 2017, p. 37.

[5] Ana Maria B. Fresnido, Agnes S. Barsaga, *Transaction Log Analysis of Opac Searches in an Academic Library: Basis For Opac Interface Improvement*, "PHJLIS", 39 (2019), 1, p. 29-40.

[6] Rachael A. Cohen, Angie Thorpe Pusnik, Measuring Query Complexity in Web-Scale Discovery: A Comparison be.tween Two Academic Libraries, "Reference & User Services Quarterly", 57 (2018), 4.

[7] Brighid M. Gonzales, *Analyzing EZproxy SPU Logs Using Python Data Analysis Tools*, "Code4Lib", 42 (2018), https://journal.code4lib.org/articles/13918.

[8] Sultan M. Al-Daihani, Alan Abrahams, *A Text Mining Analysis of Academic Libraries' Tweets*, "The Journal of Academic Librarianship", 42 (2016), p. 135-143.

[9] Svein Arne Brygfjeld, Freddy Wetjen, André Walsøe, *Machine learning for production of Dewey Decimal*, relazione presentata all'IFLA World Library and Information Congress svoltosi a Kuala Lumpur dal 24 al 30 agosto 2018.

[10] Gregory Wiedeman, *Python for Archivists: Breaking down Barriers between Systems*, "Practical Technology for Archives", 7, gennaio 2017.

[11] Guido van Rossum, *Computer Programming for Everybody (Revised Proposal). A Scouting Expedition for the Programmers of Tomorrow*, Corporation for National Research Initiatives, July 1999 (CNRI Proposal # 90120-1a).

[12] Il corso è disponibile all'indirizzo librarycarpentry.org/library-python.

ABSTRACT

Working with a multitude of digital tools is now a core part of a librarian's or an archivist's skillset. We work with collection management systems, digital asset management systems, public access systems, ticketing or request systems, local databases, general web applications, and systems built on smaller systems linked through application programming interfaces (APIs). Yet, this landscape can be difficult to manage. How do we get all of these systems that rely on different software and use data in different ways to talk to one another in ways that help, rather than hinder, our day to day tasks? Python is the perfect power tool for this problem. Its syntax is relatively simple and intuitive, it is extremely well-documented, and it has an enormous number of libraries optimized to solve just about every problem we have.

DOI: 10.3302/0392-8586-202007-045-1