Ben Prescott
First Research/Programming Assignment
MSDS 453 2021SP
4/15/2021

Exploring Remote Work Issues Using Web Crawling

Abstract

The appearance of COVID-19 caused many organizations to scramble in hopes of finding solutions to support employees now required to work from home. For many organizations this was a new challenge, while others have embraced either a flexible work environment, or have had fully remote positions since inception. This approach focuses on the evolution of long-standing personnel issues, such as personal interaction and a sense of belonging, as COVID-19 continues to shift business operations.

To aid in this research, I've desiged a web crawler responsible for collecting many documents from a single web domain. Each document is given an ID number, the URL is recorded, and the main body of text of each article is recorded. This provides a corpus for future use for analysis such as article summarization, sentiment, finding token commonality across the corpus, or identifying phrases that may allude to issues in a remote work culture.

Introduction

Having spent many years in a management role, I am very familiar with employees struggling to feel connected with their peers. As businesses were starting to shutdown due to the spread of COVID-19, many organizations quickly found ways to simply support business operations. However, many employees, who were used to working from an office everyday, started to struggle with the concept of working from home. Depending on the organization, this presents a brand new challenge for management, or may heighten employee sensitivity in organizations who are familiar with a remote workforce.

There are many articles on the internet that provide guidance to management who are faced with a mixed or remote workforce. One website, Business.com, provides articles targeted towards managers, human resources, and general employee-related articles. This web domain was the target of the web crawling exercise described later.

Activities of a Focused Crawler

According to Chakrabarti, van den Berg and Dom in their paper titled ‘Focused crawling: a new approach to topic-specific Web resource discovery’, there are eight activities that a focused crawler should perform:

- *canonical taxonomy creation*
- *relevant example collection*
- *taxonomy refinement*
- *interactive exploration*
- *classification modeling*
- *resource discovery*
- *distillation of topics*
- *user-involved feedback*

Canonical taxonomy creation is a way for the classifier to determine like-URLs, such as a group of website URLs belonging to a topic of ‘remote work burnout’. *Relevant example collection* describes a crawler that only looks at relevant documents based on the user’s topic. *Taxonomy refinement* is a method of ‘honing in’ on the best document classes in effort to provide a more targeted search. *Interactive exploration* is the ability for the crawler to provide the user with some ‘recommendations’ or other similar URLs, allowing the user to select to include or disregard them in the crawling effort. *Classification modeling* leverages new input from the user to refine the initial pre-training from an existing dataset. *Resource discovery* is the actual crawling/scraping process, searching for the most relevant documents based on the input of other activities. *Distillation of topics* is the crawler’s method of finding other ‘chunks’ or ‘hubs’ of URLs that are relevant to the search topic, potentially expanding the crawling landscape and leading to more relevant documents. *User-involved feeback* is more of a continuous inspection of what the crawler is finding and raising to the user. This way the user can provide more real-time input to the relevance of additional URLs/documents found and providing updates to the relevance classifier.

Crawler Method and Results

I started this research by determining if any corpora exists for this specific topic. I was unable to find any existing corpora that focus on remote work issues. Given that this initial stage of exploration is a bit broader, this is not very surprising. I pursued creating a crawler to generate my own corpus of text from articles related to remote work.

The initial crawler I created for exploring the issues faced in remote work only leverages a few of these activities, specifically: example collection and resource discovery. Two slightly different approaches were explored. One being a crawler that takes a manually-added list of URLs from the user (similar to the manual form of *example collection*), and a method where the crawler finds articles on a root URL and creates a new list of the URLs found on that page, matching a specific regex chain for filtering. The current crawler is more basic in nature, in that it leverages a manually-populated Python list of comma-separated URLs to scrape against. This was due to a feature limitation of this Crawler in finding relevant URLs on business.com's search result page.

A third-party plug-in for Google Chrome, named Link Klipper, helped to significantly speed up the otherwise tedious URL collection process. Link Klipper provides the URLs in either a text file or CSV, which is then read and iterated over using Python's *csv* library. The URLs are appended to a Python list which is then used by the bulk scraping effort.

The crawler leverages Python’s *requests* library and *BeautifulSoup* to perform a GET request for each site in the list, then looks for anything in the HTML with ‘article’ in the name, referencing a business.com article to scrape. The crawler then looks for any ‘p’ tags in the article that references a paragraph of text. Each paragraph in the article is appended to a temporary list, which is then added to a Pandas dataframe, which helps make visualizing the corpus easier prior to conversion. Prior to conversion, the article text is converted to lowercase and stopwords are removed using Python’s *NLTK* library. The Pandas dataframe is then converted to a Python dictionary using ‘records’ format, then ultimately written to a JSON lines file for future analysis. The resulting corpus consists of 166 documents and 884,359 tokens.

Improving The Crawler

One of the first approaches to improve the crawler might be to leverage a labeled training dataset to train a supervised classification model to determine relevant pages for the topic. However, this poses the problem having labeled training data, or manually creating some. Another approach would be to explore a clustering method to group URLs based on components within the URL. I would explore the clustering method to avoid potentially unnecessary labeling exercises.

The other addition would be that of a distiller, which will help identify hubs of additional links that may have been skipped or misclassified, but are relevant to the search. There may be links to root domains on a page that are flagged as ‘not relevant’ by the classifier, but by providing distiller feedback to the user are determined to be relevant. They can then be added back into the set to be crawled.

References

Chakrabarti, S., Van den Berg, M., & Dom, B. (1999). Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks*, *31*(11-16), 1623-1640. doi:10.1016/s1389-1286(99)00052-3

Chandrasekhar, G. (2019, November 07). Using ai to automate web crawling. Retrieved April 19, 2021, from https://www.semantics3.com/blog/ai-for-automated-web-crawling/