INTRODUCTION

Can a meaningful analysis tool be produced to analyze the quality of, or deduce inference from, various policy documents? If so, what would be the measurement on the dialogue? What are the expectations? What are the implications?

These questions are the focus of this research discussion. The discussion details exploratory studies into developing a web scraper to analyze text and links through variable websites. The target of the example implementation is privacy policy.

Following exploration of prior early research and reports, it became apparent that efforts are being made to implement modes of enforcement and overwatch as it pertains to privacy practices. The awareness of the DNSMI link on webpages is far from absolute. Without being directly confronted with such information discussing the efforts behind this new practice, associated particularly with California, it is hardly likely to be observed at the bottom of a web page. The link supposedly provides options on data collection and privacy for the user. The dialogue is like the introductory steps of a new iPhone or Windows laptop, with various choices for location services, monitoring, and data feedback. It was discovered that these mysterious links in fact exist, as this research project was carried out, though the prevalence of the links is modest. Though accessory exploration is still being done, preliminary scraping results seem to indicate, reasonably, that most links reside in California based websites or New York, the hubs of the Earth.

In this research paper, discussion is carried out on findings and troubles during efforts to produce a formidable web analysis software to determine the location and prevalence of these links, while also conducting in depth, modular analysis on website privacy policy pages. The results in fact confirm that this software is stepping in a promising direction, however the complications encountered in the research, as will be later discussed, make it apparent that more data must be collected to formulate conclusions.

This research approaches the task by attempting to parse out various sorts of keywords and phrases, with a long-term goal of developing a scoring system and scale to attribute valuable meaning to the results. Preliminary findings found convincing deviations based on keyword categories and state or industry variances. That these observations are convincing is credible enough, as the question of validity is only a tease for further development and exploration. The software produced to carry out the study occupied most of the timeframe allotted in the initial stages of research. Therefore, the showcase associated with this presentation is purely demonstrational, as the sample size and keyword space were drastically cut. Some 2,500 websites of 26 industry categories were set to be analyzed. This number was slashed substantially due to time constraint. Further research will seek to accumulate the necessary data to draw the desired conclusions and bolster confidence in the evidence.

Can Informative Web Scrapers Be Built to Analyze Text Such as Privacy Policies?

University of Iowa Research Project:

Networking : 3640

Jonathan Boyd

METHODOLOGY

Considerable time has been spent developing the scraper utilized in the study. The focus and goal were to avoid all signs of specialization or direct implementation. This goal is reflected in the present result, where most all the code manages to avoid reference to policy or privacy links. The initial stages of development began with developing a sense of capabilities utilizing playwright for python.

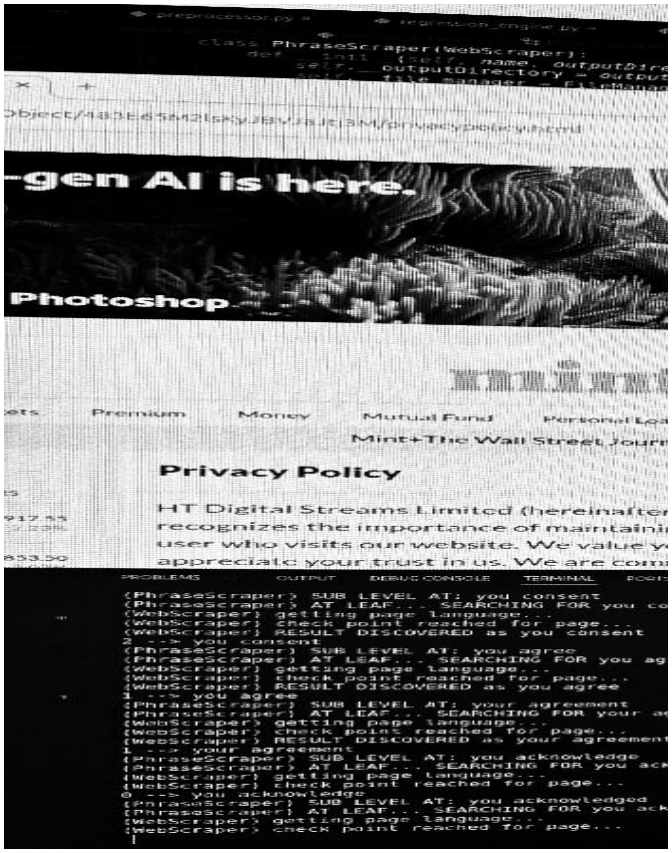
The first development was the playwright driver module, whose goal was to simplify the instantiation and management of playwright, working alongside the produced page manager to facilitate browser navigation for the rest of the scraper module. Before considering anything about privacy policy pages, the focus was set on smooth interaction with webpages, and predictable responses to failures. Following the completion of the driver and page manager, the translation unit was developed. There was a strong desire to observe data in various countries. The results of these attempts will be later extrapolated upon. The module was created to seek the first instance of text on a given web page possible and determine the language using googletrans. The features which draw upon the translator and store results came later in development. Time was spent analyzing German, Japanese, Russian, and French with success.

Furthermore, before developing the scraper, initial steps were taken to scrape website data for the study. This task having been deemed external to the cause explains the informal nature of the files in the website gathering folder, as these pursuits detail raw perspective on the initial phase of learning how exactly playwright worked. The website data was pulled from ahrefs.com. The initial plans accounted for 2,500 websites to be scraped from various industries and countries. Unfortunately time constraints on the project due date slashed this number to 120 or so. The results do indeed indicate that the 2,500 would have been of great use, and therefore further pursuits and development is inevitable, as will be later detailed. The websites country and state data were pulled via an ipwhois website. There were various encounters with websites which were prone to providing random proxy data rather than legitimate information (i.e., saying Chick-fil-a was headquartered in Malaysia). Having discovered one site which was convincingly accurate, effort was put forth to scrape it in entirety. Much difficulty was encountered when dealing with security measures, as also during the scrape. Various tactics were implemented to bypass captcha, by assuming country information based on the url; however, it did limit the pull. It is believed that the resulting data is accurate.

Development later began on a small scale, starting at the web scraper module. It was designed to pull elements arbitrarily by role. The difference from the informal and specialized website data scraper is immediately apparent…

Progress was measured on the capability to pull links from a small number of websites. The focus had not initially referred to seeking to discover any phrases, which plays into the structure present, which separates these functions into two units. The policy scraper was initially only designed to discover a singular link; however, it was noticed that various websites delivered multilink query results, ultimately leading to the development of a control flow which handled various potential links. This was accomplished through trial and error, utilizing playwright via the computer terminal to test idea before translating them to class code. The phrase scraper was conceptualized prior to this in proximity to confirming the functionality of navigation tools and the grab method.

The precise direction of the research was determined while developing the phrase scraper. The initial plans were to develop a scoring system or generally implicative analysis on the nature of speech in various policies. The idea is still echoed through the program structure and the generic mode of development preserves the capability of running a routine which produces the prior desired results. That is, various key words were developed and subcategorized by tone, subject, or potential implication. It was thought that binding these keywords to scalars or developing some statistical analysis would potentially yield promising results. Unfortunately, realization of scarcity of time led to a retraction in the amount of data being scraped (i.e., less key words). While the results still demonstrate the potential and capability, as will be later discussed, the present keywords drastically mask to prior intent as their categorizations have been simplified.

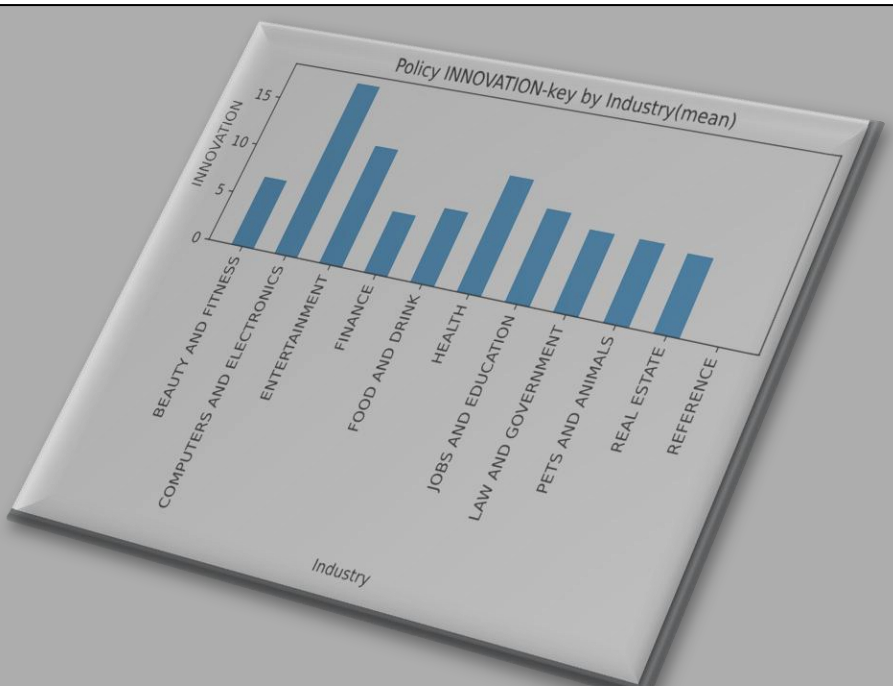


These plans nonetheless guided motive. The particular 'has text' functionality came into being, and various components of the scraper modules were further developed and improved as efforts were made to confirm the capability to acquire text as desired. Components were integrated to ensure that multilanguage support and some form of caching system was in place. Many optimizations however became relatively flawed as the browser operation seemed to result in the scraper having a newfound tendency to repeatedly open new pages for new sites, an act that was not initially prevalent. The precise cause of this is not yet determined. There is concern that the influx of pages could potentially lead to the environment running the playwright engine crashing.

Following completion of the scraper framework, the complexity of actions led to the idea of developing the Web Analysis Suite. This class functions to simplify the interface and avoid crowding implementation pages with confusing code. Even further as this suite was developed, the goal was for it to be generic. It was initially termed as a Policy Analysis Suit, and then refactored to be entirely independent. That is, there were attempts on a small scale to run the file with a small set of key words. Expansion of the codebase into the large system now present began after complete development of the scraper module. All subsequent development began to directly consider how masses of text data should be processed, cleaned, validated, and stored. There was no initial concrete conception of graphic visualizations or end results. The focus of processing the data orderly led to the development of various modules and classes that act together on the data. The concept only detailed a data cleaner and a storage module. It was known that there would be statistics and some sort of analysis, but the details were ignored. The discoveries in implementing these concepts continued to drive development further into more complex integrations. The present WAS Manager and its cleaning pipeline is an effort to mask this complexity for the user and provide a simple way to get data from scrape to CSVs. After conquering various discouraging troubles with accessing data and fixing peculiarities in error handling, testing was directed towards small scale, direct attempts to parse policy pages of varied languages.

The final task was to begin scraping for DNSMI links and policy pages. The initial attempts led to the reduction in ambitions, as the keyword pool was gutted, and the website list was slashed. This and various playwright tweaks drastically improved speed. Website data was collected over 48-72 hours. Unfortunately, there were various crashes and reconfigurations as development continued alongside the scrape. It was observed that various websites incorporated security protocols that required someone to be present to ensure a valid scrape occurred. There were also issues where the opening of new pages for links led to the scraper exploring the wrong pages. A need was found to often manually switch the scraper to a given page. It is likely that this has skewed results in some manner, but the concept and capability is nonetheless preserved. There was a capability to retract and resolve the issue, but efforts were made to avoid interrupting the procedure, which would cause a loss of accessory query data (i.e., DNSMI information), given flaw in the programming control flow. As will be later detailed, the overall results were intriguing, thereby warranting a fix to this bug followed by a repeat of the experiment.

Following the scrape, exploratory graphing was performed. Various modules were created in advance to allow for easy manipulation of data and creation of graphs. For curiosity's sake, machine learning models were tested. A general lack of familiarity led these explorations to nothing. Later bar charting was employed, leading to interesting observations, though speculation is warranted given the bugs prior mentioned. Numerical data ranged from readability/complexity scores to traffic value, domain rating, and traffic count. In time, the full suite of graphs may be shared.



RESULTS

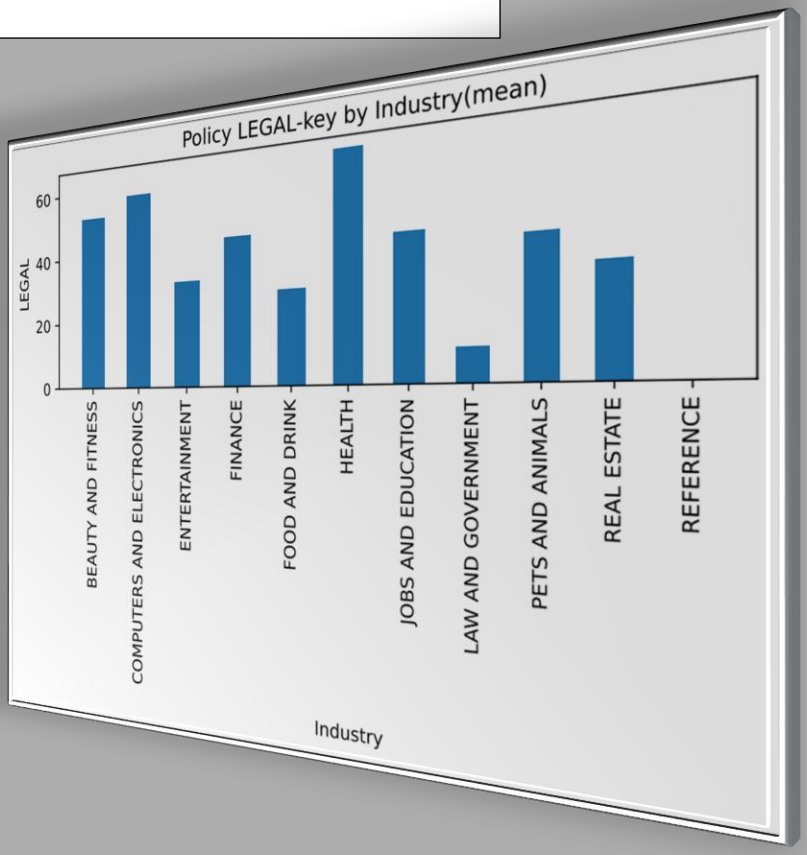
The results of the study were interesting; however, various observations call for further research before developing any profound sense of certainty.

Through the process of collecting data, various moments were encountered where pages with DNSMI links did not register to the scraper. The reasons could vary. Some pages utilized progressive loading, causing the lower links to fail to register. This was initially fixed with code implementation to briefly scroll down and up the page. This produced significant complications with timeouts, given media heavy pages subsequently were stuck in profoundly long load times. Other pages may have strange html structures that complicate the parser. Other occurrences are without explanation, as implementing likewise steps in playwright via console resulted in success. This variability is perhaps just an aspect of technology as it stands. One website was observed to possess one when it did not have anything referring to the key words during manual investigation.

There were also various cases where websites possessed popups for privacy policy documents. These accessory windows don't appear to register to the parser. Manual exploration has yet to be done to confirm the impact. Various websites utilize Cloud Fare and other security measures that prevent the scraper from accessing the page or its links. These complications paired with an inability to be consistently present with the scraper leave a gray area in confirming the accuracy of the results. There is no present way to confirm what occurred as the scraper was left to work independently. Regardless of such complications, the relative sensibility of the results demand attention and curiosity.

Observations corresponding to key word categories, even in the presented level one hierarchical model, were promising. The meaning behind "Policy LEGAL key" as in the image below refers to a search on terms judged to revolve around law and legality. The sub-tree in the former conceptualization extrapolated in a manner which would have provided grounds for statistical analysis within any given sub-domain. Phrases such as "regulation", "compliance", "law", "in accordance with," were searched. Prior goals split categories to isolate a subbranch of phrase that appeared as "legal guarding;" such as "you consented," "when you signed," "you acknowledge."

Complexity analysis was only relatively informative. There is strong belief that a larger sample size with more diversity would make these statistics more inciteful. In totality, it would be best to observe them as an example of capability for the scraper, given the statistical methods can be dynamically swapped without compromising functionality. Various crashes and the ordering of websites did also limit the capability to reach all industries. On a small scale such as this, it would have been wise to perform mixed sampling. The initial conceptualization assumed all 2,500 websites would be processed regardless of order. Current attempts are being made to run the application on a shuffled set.



CONCLUSION

In conclusion, the research here is far from conclusion. Efforts are being made to increase the sample size and implement the aforementioned scoring concept. The small-scale implementation nonetheless provided grounds to confirm system functionality and feasibility of the general conceptualization. Efforts will be made in increase the sample space of non-US countries. There will also be a focus on diversifying the industry space, with hopes of capturing the entirety of the 2,500 pulls sources. Various features are still to be implemented in the scraper. In observing various bugs and complications with popup windows, it was discovered that it is easily feasible to produce a live interactive mode, which provides a persistent parse of webpages as they are visited, regardless of the changing URL. This was judged as an interesting addition, to later bring into reality. Steps will also be made to account for the websites that have opted to place their policy on nested pages. For example, various sites have their policy information under the legal page. Some URLs are implementing policy pages that have various links to extrapolation. The current conceptualization is to capture new, unobserved links each time a policy page is visited, to handle potential follow ups. It was however observed that the current implementation manages to do this on its own often somehow. "Legal" can be added to the link key phrase list. This was initially avoided, as it was deemed poor form to possess a privacy policy page so deviant from the masses so as to not possess a "Privacy" button. Future implementations will traverse these links also. As these ideas develop, the primary task remains to develop a system of judgement to attribute meaning to the results and numbers. Where the work described more so aligns with a developer or programmer, it is perhaps necessary for a well learned data scientist to ascribe meaning to the data at the finish line.