Web Scraping Using Summarization and Named Entity Recognition (NER)

Bhavya Bhardwaj, Syed Ishtiyaq Ahmed, Jaiharie J, Sorabh Dadhich R, Ganesan M
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
bhavya1705@yahoo.com, m_ganesan1@cb.amrita.edu

2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) | 978-1-6654-0521-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACCS51430.2021.9441888

***Abstract*—"In the age of information, ignorance is a choice"-Donny Miller, this line perfectly encapsulates the relevance and importance of information in the present digital era. With the rise and spread of the internet, growth and prevalence of social media usage in the youth demographic and tons of data being generated by different businesses and industries, there has been an exponential rise in information being generated on a daily basis on multiple platforms and across all demographics of modern society. This significant rise in information generation has made the development of information processing and analyzing techniques imperative. The spread of the internet across the globe, has made it the largest repository of information and data. Internet companies, stock companies, market analyzers and various other businesses use sophisticated tools and techniques to extract information from the internet. One of the most important and prevalent method of extracting relevant data off the world-wide web is Web Scraping. Web scraping has gained significant popularity due to the ease it offers in extracting information from target webpages and presenting the information in a structured format with no manual intervention. While the traditional approach to web scraping offers significant advantages, it also necessitates foreknowledge of the DOM structure of the target webpages. In the subsequent sections of this publication, an excellent method that allows developers to bypass the aforementioned requirement, and completely automates the process of web crawling and web scraping relevant information from target URLs is presented. In this paper Natural Language Processing (NLP) and Machine Learning (ML) alternatives to the traditional web-scraping approach is presented. To demonstrate the advantages offered by the improved algorithm, an epidemic predictor mapping the spread of a variety of infectious/viral diseases and their impact across the globe is built using the alternative methods provided in the publication.**

***Index Terms*—Machine Learning, Natural Language Processing, Web-Scraping, SpaCy, NLTK, HTML, Summarization, Named Entity Recognition**

I. Introduction

In today's era, information has proven to be the most important commodity. With widespread availability of cellular technology, the spread of social media and near 100% internet penetration all over the world, tons of data are being generated every second. With 4.39 billion people connected to it and generating 2.5 quintillion bytes of data every day, internet has proven to be the largest source of data in the 21st century. It has overtaken Libraries, scholars, intellects and is today even capable of making decisions and choices for us. As the internet floods with streams of data every second, analysis and processing of the generated data is a subject of great significance and relevance. Analysis of data being generated via the internet, helps internet companies and MNCs understand consumer interests, helps security agencies to identify potential threats to society and helps product-based companies in target advertising. One of the most important and highly popular methods used for extracting data from the internet is Web Scraping. Web scraping has gained popularity as an excellent tool for data extraction as it provides structured data from web pages across the internet in an automated way. Web scraping is used by giant internet companies, market trend analyzers, security agencies and various different types of businesses to extract information of significance from across the world wide web and make smarter choices and decisions. The traditional way in which web scraping is performed involves creating a list of target URLs to parse through and requesting access to the HTML content and DOM structure of the target webpages. After gaining access to the HTML and DOM structure of target webpages, a plethora of HTML parsers, and other text locators are utilized to search and identify relevant textual data from a particular webpage. Despite giving excellent results, the traditional web scraping approach necessitated foreknowledge of the DOM structure of the webpage, in order to extract relevant data from different HTML elements and objects in an automated fashion. In this publication, the authors propose an excellent method that can be implemented to bypass the requirement of foreknowledge of the webpage structure, and automates the process of web crawling and web scraping relevant information from across the world wide web using Natural Language processing (NLP) and Machine Learning (ML) for web scraping. The proposed method makes use of sophisticated machine learning techniques like Named Entity Recognition (NER) and summarization on HTML content of webpages. The authors make use of python libraries like SpaCy [1], Natural Language Toolkit (nltk) [2] etc for processing the raw HTML content of target webpages. For the duration of this paper the sample web scraping for the purpose of epidemic prediction using the above method will be discussed as reference. The epidemic predictor based on web scraping using NLP focuses mainly on extracting numerical data especially cardinal numbers which denote the total count of a particular entity.

978-1-6654-0521-8/21/$31.00 ©2021 IEEE

II. Existing methods and approaches in Web Scraping

Web scraping is the term used to refer to various methods and techniques used to extract relevant information and data from various websites in a structured format [3][18]. The process of web scraping is automated requiring no manual intervention. As of today, web scraping can be implemented using programs, utilizing the excellent libraries offered by powerful languages like Python etc. The process of web-scraping is straightforward and simple. Initially, the scraper makes a request to the target URLs to get access to the HTML content and DOM structure of the target webpages. The content of the target webpage is usually in the form of markup-language document like a HTML document or XHTML document. Once granted access to the content of the target webpage, the next step comprises of parsing through the content of the web-page and processing and analyzing the content and DOM structure [3][16]. While parsing through the HTML content scraped off the webpage, the parser identifies and extracts relevant information like the title/heading of the document, the various opening and closing tags present in the webpage, paragraph tags, class tags and different attributes and their corresponding values etc. The parser stores and retrieves important information, as it passes through the document. The information scraped off the webpage is then stored in either a CSV file, or a JSON file or in some database [4].

Web scraping that entails accessing content from target webpages can be implemented in various ways, it could be done manually by copy-pasting content from the site of interest into a document. This crude and basic method is to be used only when the target webpage blocks the user-agent/robot from parsing through any of its content. Another method is accessing different webpages through their APIs. Accessing information through APIs of webpages is advantageous in that it offers all the content in a very structured format and hence makes parsing easier and more efficient [4]. Developers and companies can also build web scrapers that visit target webpages, extract information from the HTML or XHTML document and build the DOM structure of the webpage. However, the methods described above have a disadvantage that they require the scraper to have foreknowledge of the DOM structure of the target webpages. An excellent technique that will bypass this requirement is presented in the subsequent sections of the chapter. The advancement of web-scraping technologies and commercialization of web-scraping has led to various web-scraping “tools” being offered by companies online to do automatic web-scraping like Scrapy, import.io, Parsehub etc. that offer web-scraping services to their clients[4].

III. Web Scraping using Machine Learning and Natural Language Processing (NLP)

Machine Learning and its related areas have largely dominated the workflow in the 21st century from optimized workflow to automated supply chains. As we conquer newer areas using Machine Learning, it is time to use Machine Learning, or its subset Natural Language Processing for automating web parsing.

Similar to the conventional approach workflow in Fig.1, the process starts with requesting the HTML and DOM structure of the webpage that needs processing. It is of vital interest to understand here that the webpage that is scraped or parsed is completely random, the knowledge and architecture of its DOM structure is not known, and neither is it assumed to be following any trends. Once the tags and any fillers that plague the HTML text are removed the data obtained newly in string format, here after referred to as original text or text, is ready for processing using NLP techniques. In the scope of this paper two approaches will be considered, 1) based on summarization of text and the other 2) based on NAMED ENTITY RECOGNITION (NER). The above methods were used for parsing news articles obtained using GoogleSearch library[5] in Python.

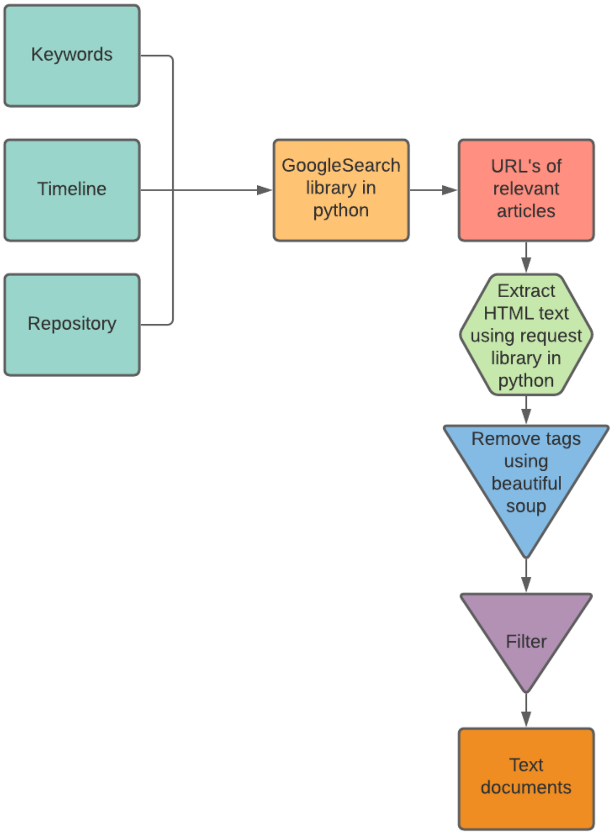

Keywords
Timeline
Repository
GoogleSearch library in python
URL's of relevant articles
Extract HTML text using request library in python
Remove tags using beautiful soup
Filter
Text documents

Fig. 1. Overview of the methodology employed in crawling relevant URLs and extracting HTML content

A. Approach 1: Summarization

Summarization of text can be broadly defined as the sub set of the original text that represents the most important and relevant information, in essence a shortened version of the original text. A summary is a shortened version of the original text that gives the reader full insight and complete details about the original text. In Natural Language Processing (NLP), this is done by assigning weights to the words in the text, creating a corpus and then scoring sentences in the text based on the words it comprises of. The sentences with the maximum scores are then chosen to form the summary of the text. While there exist several better and sophisticated techniques for summarizing text, they will not be considered in the scope of this paper[6][7][8]. To form the corpus or bag-of-words [9][10], Term Frequency Inverse Document Frequency (TF-IDF)[11] was used. The workflow can be seen in Fig.2. In TF-IDF, the text after filtration is passed through a tokenizer that splits the given text into a list of sentences and words, the “bag-of-words model” [9][10] obtained from TF-IDF, is appended with certain keywords (specific to the content to be scraped) which are assigned the highest value. The sentences are then graded based on their word content and stored in a list with their corresponding scores, following which, the top sentences are used to form the summary. As the summary may in itself be larger than the expected result, it is further filtered using keywords, in this case focused on the current corona epidemic [12], relevant numerical values are filtered from the text. Additional data such as country, state and city are also extracted from the text using the pycountry library [13] in python. The results are then stored in the appropriate data structure. Though this method of Summarization is invariant to the DOM structure and requires no foreknowledge of it, it is still not as highly efficient as the conventional method. The data obtained through this may miss some necessary and crucial details lost during summarization or during further filtration. To improve accuracy, the second method of Named Entity Recognition is used.

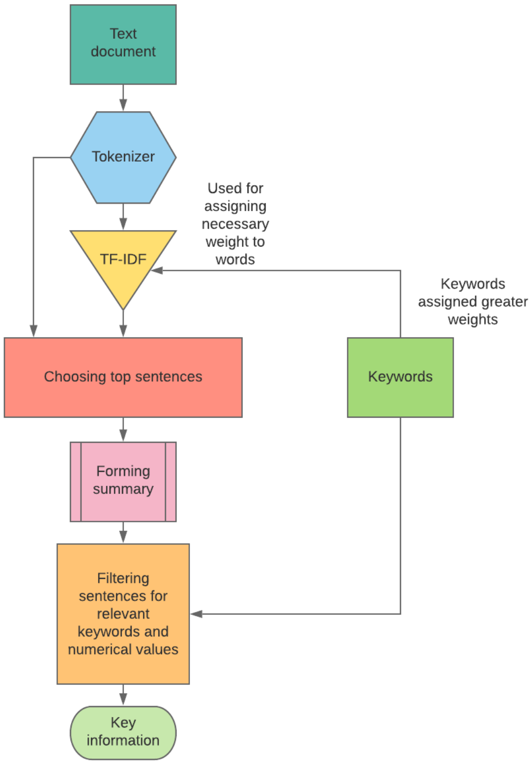
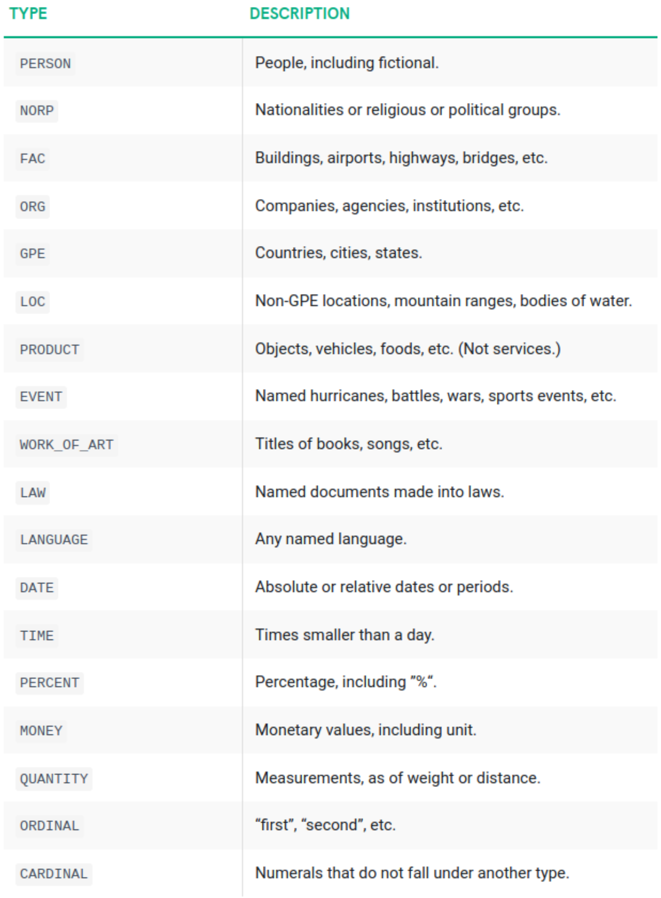

Text document
Tokenizer
Used for assigning necessary weight to words
TF-IDF
Keywords assigned greater weights
Choosing top sentences
Keywords
Forming summary
Filtering sentences for relevant keywords and numerical values
Key information

Fig. 2. Summarizer

B. Approach 2: Named Entity Recognition (NER)

Named Entity Recognition is a subset of Information Extraction and gathering that aims to classify words and segments of a sentence consisting of structured or unstructured text into pre-defined entities such as those mentioned in Fig.3. For the scope of this paper, we will consider three entities specific to the task for which this method was employed - Scraping News Article on Possible Viral Outbreaks [3]. The entities in consideration are GPE, CARDINAL and EVENT. The GPE entity as can be seen from Fig.3, shows Countries, Cities and States I.e., location of possible outbreak. The CARDINAL entity shows numerals that do not fall in QUANTITY, ORDINAL or PERCENT entities that generally depict values that quantify. The above-mentioned entities like QUANTITY, ORDINAL OR PERCENT can also be considered as per need, but will not be discussed in the scope of this paper. The EVENT entity describes world events with large impact such as hurricanes,



TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including "%".
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	"first", "second", etc.
CARDINAL	Numerals that do not fall under another type.

Fig. 3. Entity Types

battles, sports and even pandemics. The workflow for web scraping using Named Entity Recognition (NER) as depicted in Fig.4 can be understood as follows: the text is passed through an NER model, in this case the en_core_web_sm model available readily from SpaCy library [1] in Python. Several such models exist for different datasets, languages, and use, and can be easily downloaded either from SpaCy documentation [1], or can be trained for custom purposes. The individual terms in “en_core_web_sm” [14] model here stand for en- English, core - core for general-purpose model with vocabulary, syntax, entities and word vectors, web- Dataset Web, sm- Small. The output from the model is a list of words classified in different entities. The three entities specific to the task as mentioned above and as depicted in Fig.4, are considered. With the help of keywords, the results are then sorted on the basis of relevance and stored. Compared to summarization, the NER approach yields better results, that are closer and similar to the response obtained through conventional methods.

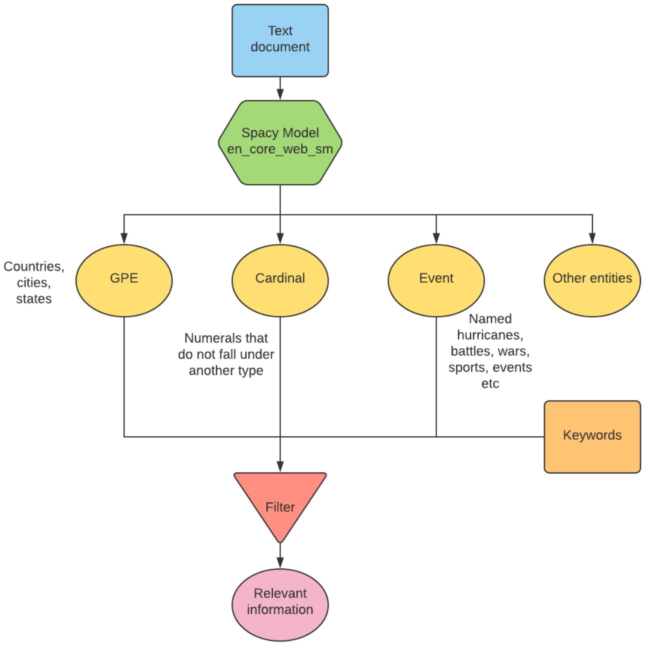
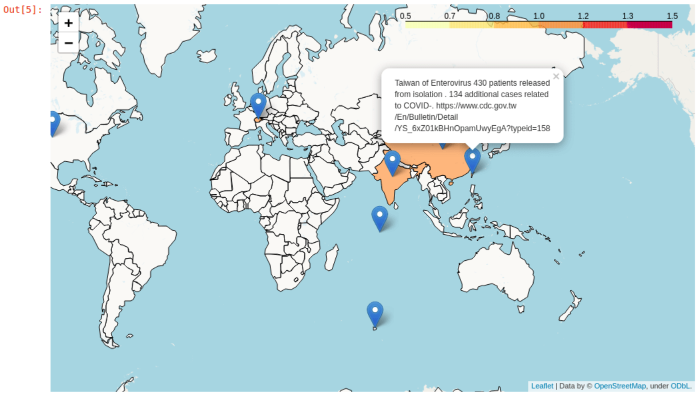

Text document
Spacy Model en_core_web_sm
Countries, cities, states
GPE
Cardinal
Event
Other entities
Numerals that do not fall under another type
Named hurricanes, battles, wars, sports, events etc
Keywords
Filter
Relevant information

Fig. 4. Named Entity Recognition Approach

IV. Results and Discussion

Employing the aforementioned techniques of summarization and NER, the authors built an epidemic predictor as a tool for medical organizations/government agencies to obtain a comprehensive report of the spread of viral diseases across the globe. The algorithm extracts relevant keywords related to diseases/epidemics from news websites and other information sources online, and presents the obtained statistics in a structured format. The results are then plotted on a world map, as in Fig.5 to study the currently active diseases and the number of infected people across the globe. This epidemic predictor helps medical agencies hotspots of various diseases which aids them in taking necessary action.



Taiwan of Enterovirus 430 patients released from isolation . 134 additional cases related to COVID-. https://www.cdc.gov.tw /En/Bulletin/Detail /YS_6xZ01kBHnOpamUwyEgA?typeid=158

Fig. 5. Obtained Output

V. Inference

In the scope of this paper, we have observed the conventional methods of Web Scraping, that are limited by the foreknowledge of the DOM structure, and the proposed alternatives of Machine Learning (ML) and Natural Language Processing based approaches that are invariant to the DOM structure. The response time of conventional and non-conventional methods are almost similar and are dependent on the time required to access the get and post requests. The above methods were tested on News Articles on Possible Viral Outbreaks[1] yielded satisfactory and relevant results.

VI. Conclusion

In conclusion Natural Language Processing and other Machine Learning techniques can also be used for automating web scraping to a greater extent. Summarization, despite being a useful technique, lacks in dealing relevant data for scraping the web for consumer information or sports scores. It fails on data that is discrete and highly unstructured, with the largest structure being the sentence. On the other hand, Named Entity Recognition (NER) is more flexible and allows greater diversity in the data it can sort and scrap. It is the authors opinion that the NER approach will yield better results on discrete and highly unstructured data. Other techniques can also be looked at for varying degree of efficiency and need. While the existing methods serve their intended purpose, they still lack in flexibility, which can be overcome by using the above methods. In the words of Dave Waters “A baby learns to crawl, walk and then run. We are in the crawling stage when it comes to applying machine learning.”

References

[1] https://spacy.io/api/doc
[2] https://www.nltk.org/
[3] Amalia, Amalia, Rizky Maulidya Afifa, and Herriyance Herriyance. ”Resource description framework generation for tropical disease using web scraping.” In 2018 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), pp. 44-48. IEEE, 2018.

[4] GOEL, SANYA, MUDIT BANSAL, ATUL KUMAR SRIVASTAVA, and NEHA ARORA. ”Web Crawling-based Search Engine using Python.” In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 436-438. IEEE, 2019.

[5] https://pypi.org/project/googlesearch-python/

[6] Rani, S. Siji, K. Sreejith, and Arjun Sanker. ”A hybrid approach for automatic document summarization.” In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 663-669. IEEE, 2017.

[7] Shivakumar, K., and Rab Soumya. ”Text summarization using clustering technique and SVM technique.” International Journal of Applied Engineering Research 10, no. 12 (2015): 28873-28881.

[8] Aji, Subhanpurno, and Ramachandra Kaimal. ”Document summarization using positive pointwise mutual information.” International Journal of Computer Science & Information Technology 4, no. 2 (2012): 47.

[9] Zhang, Yin, Rong Jin, and Zhi-Hua Zhou. ”Understanding bag-of-words model: a statistical framework.” International Journal of Machine Learning and Cybernetics 1, no. 1-4 (2010): 43-52.

[10] https://machinelearningmastery.com/gentle-introduction-bag-words-model/

[11] Christian, Hans, Mikhael Pramodana Agus, and Derwin Suhartono. ”Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF).” ComTech: Computer, Mathematics and Engineering Applications 7, no. 4 (2016): 285-294.

[12] Mufid, Mohammad Robihul, Arif Basofi, Saniyatul Mawaddah, Khusnul Khotimah, and Nurul Fuad. ”Risk Diagnosis and Mitigation System of COVID-19 Using Expert System and Web Scraping.” In 2020 International Electronics Symposium (IES), pp. 577-583. IEEE, 2020.

[13] https://pypi.org/project/pycountry/

[14] https://spacy.io/models

[15] Veena, G., Deepa Gupta, S. Lakshmi, and Jeenu T. Jacob. ”Named Entity Recognition in Text Documents Using a Modified Conditional Random Field.” In Recent Findings in Intelligent Computing Techniques, pp. 31-41. Springer, Singapore, 2018.

[16] Sirisuriya, De S. ”A comparative study on web scraping.” (2015).

[17] https://spacy.io/api/annotation#named-entities

[18] Introduction to Web Scraping - GeeksforGeeks

[19] Thandapani, Sachin Prabhu, Subikshaa Senthilkumar, and S. Shanmuga Priya. ”Decision Support System for Plant Disease Identification.” In International Conference on Advanced Informatics for Computing Research, pp. 217-229. Springer, Singapore, 2018.