License-aware web crawling

Stefan Ilich, University of Twente, The Netherlands

The training of generative artificial intelligence (AI) models demands extensive datasets often sourced from web scraping. However, current practices frequently overlook copyright compliance, posing significant ethical and legal challenges. This project aims to develop a tool for license-aware web crawling leveraging natural language processing (NLP) techniques to detect and extract licensing information from websites automatically. The tool demonstrated high accuracy in license type detection, achieving 100%, and moderate effectiveness in extracting license text, with ROUGE-L scores showing an F1 score of 0.499, precision of 0.588, and recall of 0.503. By identifying the specific license type, the algorithm facilitates the creation of legally compliant datasets essential for responsible AI training. This tool not only ensures adherence to copyright laws but also promotes ethical data usage, thereby supporting the sustainable advancement of AI technologies.
Additional Key Words and Phrases: Natural Language Processing, Web Scraping, Artificial Intelligence

1 INTRODUCTION

In the rapidly evolving field of artificial intelligence (AI), the development of generative AI models is increasingly dependent on vast data. However, many current models do not adhere to copyright laws, leading ethical and legal challenges [3]. To address this there is a growing need for training data with texts and images that is explicitly available under open licenses. The goal of this research paper is to develop a tool that will use Natural Language Processing techniques to detect and extract licensing information from websites. This will enable the creation of legally compliant datasets that respect copyright. The tool will enable the gathering data that can be confidently used in training generative models and therefore it will facilitate the ethical advancement of artificial intelligence. According to my findings, the tool effectively identifies and extracts licensing information from various websites, achieving an average license type accuracy of 100% and moderate ROUGE-L scores (F1 score of 0.499, precision of 0.588, and recall of 0.503). These results highlight the tool's potential to create large, ethically sourced datasets for AI training while ensuring compliance with copyright laws.

2 RELATED WORKS

In [1] the focus is the state-of-the-art web scrapers and web crawlers and the legal and ethical challenges that arise when this technology is made use of. It outlines the technical processes involved in scraping and emphasizes the importance of adhering to legal standards to avoid potential violations. While this work provides an essential foundation in the technical and ethical frameworks of web scraping, it does not delve into specific technical solutions for detecting and handling licensing information on websites. The proposed project extends the general discussion by focusing specifically on developing a tool that can detect and extract licensing data, thus offering a targeted solution to the legal challenges outlined.

In [2] the ethical landscape of web scraping is outlined. It discusses the nuanced ethical considerations that must be navigated when extracting data from websites, including privacy concerns and the potential for data misuse. Unlike the broader ethical considerations discussed in this work, the proposed research is aimed at implementing a practical solution that addresses one of the specific challenges mentioned—license detection. By creating a tool that helps ensure compliance with licensing agreements, the project not only acknowledges the ethical issues raised but also contributes a practical mechanism for adhering to these ethical standards.

In [4] a tool for detecting license incompatibility in open-source software is presented. This tool automates the interpretation of various license types to identify potential conflicts, which is crucial for maintaining compliance in software development. The implemented tool aligns with this work by focusing on the automation of license detection, albeit in the context of web data rather than software.

In recent studies, there has been a significant focus on enhancing web crawling techniques to make them more efficient and context aware. For instance, in the study [5] sentiment-aware web crawling methods are introduced. Although they were primarily aimed at constructing sentiment dictionaries, the methods can be adapted for license detection by focusing on specific text patterns related to licensing.

In the context of website license information extraction, several recent studies have made significant contributions. For instance, the study [6] on the C4Corpus presents a multilingual web-size corpus with a focus on license identification. This work emphasizes the importance of identifying and processing licensing information at scale, providing a valuable dataset and framework for future research in this area.

Additionally, the research [7] on web scraping for geographic data acquisition discusses the legal and ethical challenges related to intellectual property rights and data privacy, emphasizing the need for compliance with licensing agreements.

In [8] explore the ethicality of web crawlers by studying content control mechanisms and legal standards on the web. Their research highlights the importance of annotations and

the effectiveness of license-aware crawling in adhering to recent legal standards. This study provides valuable insights into the legal and ethical considerations that must be integrated into web crawling practices, supporting the proposed project's objectives.

By comparing these works with my work, it becomes evident that while existing literature provides essential insights into the challenges of web scraping, there remains a gap in practical, technical solutions tailored towards licensing compliance. The proposed tool aims to fill this gap by offering a focused approach to detect and respect website licensing information, thereby enhancing the legal and ethical scraping practices.

3 PROBLEM STATEMENT

Despite advances in web technology, current web crawling tools often inadequately handle the detection of license data, leading to potential legal issues and misuse of data. This project seeks to develop a tool that that can automatically detect and extract licensing information from websites, enhancing responsible data utilization and compliance.

3.1 RESEARCH QUESTION

The problem statement leads to the following research questions:

1. How can license information on websites be automatically detected and extracted?

2. How effective is the proposed tool in terms of accuracy and efficiency (quantitative measures), and how satisfactory is it from a user perspective (qualitative measures)?

4 METHODS OF RESEARCH

To begin with, the first research question is answered through the implementation of the proposed tool. This involves determining the specific license type the website is under and the license text found on the website. The license type can either be Open or Proprietary. Regarding Open licenses, the tool focuses on Creative Common (CC) licenses and determines the specific license.

The next stage involves testing and refining the tool. This process is carried out by evaluating the tool's performance across a range of websites. During this phase important metrics are gathered and are used to continuously improve the performance of the tool. To begin with, the license type accuracy is calculated to determine how effective the tool is in determining the specific license type the website is under. Next the license text is tested by gathering statistical measures such as precision, recall, and the F1 Score. These measures provide a comprehensive view of the tool's reliability and accuracy.

Quantitative analysis is then conducted to rigorously assess the tool's performance. The analysis focuses on the license type and license text that are extracted. Regarding the license type, I measure the accuracy across a dataset with known license types.

Then for the license text, the Rouge python library is used to calculate the precision, recall and F1 scores to provide a comprehensive view of performance.

Finally, the methodology incorporates a qualitative analysis based on personal observations. This analysis focuses on evaluating the usability and practical utility of the tool. Observations are documented through detailed case studies on specific websites, offering deeper insights into how the tool performs under varying conditions. This qualitative feedback is essential for understanding the real-world applicability of the tool and for making any necessary adjustments to optimize its effectiveness.

5 TECHNOLOGICAL OVERVIEW

The License Extraction Tool is implemented using a combination of Python for the backend and Vue.js for the frontend. Table 1 shows the architecture of the tool and the connection between the different parts.

Python: Serves as the primary programming language for the backend, providing capabilities for web development and data processing. Python was chosen due to its readability, versatility, and extensive support for various libraries and frameworks that simplify complex tasks, making it ideal for implementing robust backend systems.

Flask: A lightweight WSGI web application framework used to create the backend API, enabling the development of scalable web applications. Flask was selected for its simplicity and flexibility, allowing quick development and easy integration with other libraries while maintaining the ability to scale as needed.

Flask-CORS: A Flask extension for handling Cross-Origin Resource Sharing (CORS), ensuring the API is accessible from the frontend. This extension was chosen to simplify the management of CORS policies, which are crucial for enabling secure communication between the backend API and frontend application.

Requests: A simple and elegant HTTP library for making GET requests to fetch webpage content. Requests was chosen for its ease of use and ability to handle HTTP operations efficiently, which is essential for fetching web pages for further processing and analysis.

Beautiful Soup: A library from bs4 used for parsing HTML and extracting data from webpages, facilitating the analysis of web content. Beautiful Soup was selected due to its powerful parsing capabilities and user-friendly syntax, making it straightforward to navigate and manipulate HTML documents.

re: The built-in regular expression library used for pattern matching in strings, essential for identifying specific text patterns. The re library was chosen for its robustness and efficiency in performing complex pattern matching tasks, which are critical for extracting specific pieces of information from text.

urllib.parse: Provides the URL join function to construct absolute URLs from relative ones, helping to manage web navigation. his module was chosen for its ability to simplify URL manipulation, ensuring that the tool can accurately follow and retrieve data from web links.
Vue.js: A progressive JavaScript framework used for building the user interface, simplifying the creation of interactive and dynamic web applications. Vue.js was selected for its ease of integration, flexibility, and strong community support, which facilitates the development of a responsive and efficient user interface.
Rouge: Rouge is a set of metrics and a software package used for evaluating license extraction in natural language processing. Rouge was chosen for its established reliability in evaluating the quality of text extraction and summarization, providing a standard metric for assessing the tool's performance.
RougeL: Variant of Rouge that computes the longest common subsequence (LCS) between two pieces of text. RougeL was selected for its ability to measure the similarity between extracted text and reference text effectively, which is crucial for evaluating the accuracy of the license extraction process.

6 PROGRAM FLOW

To begin with, the user opens the License Extraction tool in the browser and enters the URL of the website they want to analyze into an input field provided by the Vue.js frontend. Upon submission, the frontend sends a POST request to the backend, with the URL as JSON data. This is the only data that is needed for the tool to work.
The backend, implemented using Flask, receives the request and first checks if the URL is provided. If not, it returns an error message. When the URL is present, first it gets sanitized so that the main domain is determined. After that the backend uses the Requests library to fetch the webpage content. Then Beautiful Soup is used parse the HTML to identify links to relevant pages, such as "Terms", "License", or "Legal". After navigating to those pages, the backend determines whether the website is under Creative Common License or a Proprietary license. The decision is based on whether the program finds specific keywords and text patterns that are predetermined. This license type is rather easy to determine since most websites mention it in the text which is easy for the tool to detect. After the license type is found, the tool will also find relevant license information from the extracted text based on the determined license type. This is also done using predetermined keywords and phrases, but it is a lot harder to optimize due to the different nature in presenting the information in each website. At the end, both the license type and license text are returned to the frontend.
This data is then displayed to the user, providing clear and relevant details about the licensing terms associated with the website content. This process ensures that users gain a comprehensive understanding of the licenses under which the content is distributed.
Furthermore, the user is also able save the results of a license extraction in the local storage of his browser for later usage. This helps with not having wait for the extraction again when you want to access the information later.
Figure 3 is Sequence diagram that shows this entire process in a more structured and clear way. Additionally, in Figure 4, the user interface is shown in order to get a better understanding of what the flow of information is.

7 PROGRAM STRUCTURE

The license extractor application consists of a structured architecture divided into a front-end and a back end, each with specific components and roles. Figure 2 is a class diagram that illustrates the structure of the entire application.
The front end includes three main components. First, the License Extractor Vue App, which serves as the user interface. It allows users to input website links, extract license information, and save this information. It also has functions to analyze the collected license data. Second, the ExtractedLicense component, which stores details about extracted licensess licenses, such as the license type and license information. Third, the License Analysis component stores the analysis of the extracted licenses such as precision, recall, and F1 score.
The back end is composed of several interconnected components. The Flask App functions as the main entry point for handling requests from the front end. The RouteHandler component manages the processing of license data, including analyzing and finding license information. The License Extractor Back End component is responsible for fetching website content, parsing HTML, and extracting license details from the content.
Additionally, the back end includes the Rouge Evaluator component, which assesses the accuracy of the extracted data by calculating relevant metrics. To support its operations, the back end utilizes several libraries: Requests for fetching website content, Beautiful Soup for parsing HTML, and RougeL for calculating ROUGE scores to evaluate the quality of the extracted information.
Overall, the application's structure consists of a front end that handles user interaction and data presentation, and a back end that processes data and performs the extraction and evaluation of license information. The use of specialized libraries helps streamline tasks related to content fetching, parsing, and accuracy assessment, ensuring the application's components work together efficiently to extract and analyze license information from websites.

8 EXPERIMENTAL RESULTS

8.1 DATA

The data used to evaluate the tool consisted of a variety of websites with known licensing information. Each website was analyzed to determine the license type and extract license text. The tool’s performance was quantified using metrics, with a particular focus on license type accuracy and ROUGE-L scores, including precision, recall, and F1 scores. The evaluation dataset consists of the following 10 websites: Wikipedia, OpenBookPublishers, ThalesGroup, 123rf, NounProject, Louvre,

Envato, TED, Wikimedia, and Alamy. These websites were selected to cover a range of license types and complexities in HTML structures to test the tool's performance. The reason for the low number of websites used in the dataset lies in the time-consuming nature of extracting license information from websites. The decided approach to this problem was to go for quality over quantity and have a small dataset that prioritizes the quality.

8.2 QUANTITATIVE ANALYSIS

The quantitative analysis focused on two primary metrics: license type accuracy and ROUGE-L scores for the generated text. The tool achieved an average license type accuracy of 100%, indicating its reliability in correctly categorizing the license type. This high accuracy is crucial for ensuring legal compliance when using web-scraped data.

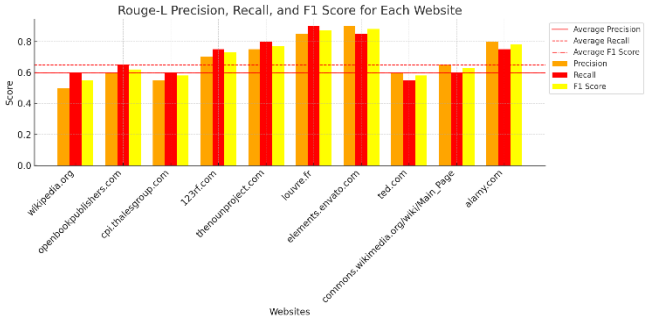
Regarding the extracted license text, the average ROUGE-L scores were moderate, with an F1 score of 0.499, precision of 0.588, and recall of 0.503. These scores suggest that the tool is effective at identifying relevant license phrases, though it may miss some parts of the text, impacting recall. Individual ROUGE-L scores varied across different websites, reflecting the tool's performance under diverse conditions. Websites with clearer and more structured licensing information, such as Envato and TED, showed higher precision, while those with more complex HTML structures or less explicit licensing information, such as ThalesGroup, had lower scores. With ThalesGroup the problem was that the licensing information was in a PDF file which made it impossible to extract for the current version of the tool. This resulted in a very low recall, precision and F1 score for this website. However, the tool was still able to determine that the website was under a proprietary license.
Figure 1 illustrates the differences in F1, precision, and recall metrics for each website, offering a clear comparison of the tool's performance across the dataset.

8.3 QUALITATIVE ANALYSIS

The qualitative analysis focuses on the practical value and usability of the license-aware web crawling tool, identifying areas for enhancement. The analysis, conducted by the researcher, will focus mainly on the generated license text because the license type detection boasts an accuracy of 100%. Initially, it is observed that the tool frequently omits sections containing license information. For the tool to recognize a text section as license information, it relies on the presence of specific keywords or phrases. However, due to the diverse ways in which license information can be presented, the tool occasionally fails to recognize vital information. While expanding the list of keywords and phrases might mitigate this issue, it could also result in a substantial decrease in precision. An excessive number of keywords may cause the tool to extract a large amount of irrelevant information, thereby reducing its overall performance.

The analysis also addresses the prevalence of irrelevant information extracted by the software. The tool is generally adept at omitting unnecessary information, especially when it accurately identifies sections containing license information and the appropriate keywords within the website text. Nonetheless, there are instances where irrelevant information is extracted. This typically occurs when the software captures specific elements of the website that include relevant keywords but are not related to license information. For instance, it is common for the tool to extract the table of contents under these circumstances.
Another significant issue is the formatting of the generated text. The lack of proper formatting often leads to poorly structured sentences, making the extracted information difficult to comprehend. This problem detracts from the tool's usability and the clarity of the information it provides.
The most substantial drawback identified is the tool's inability to extract any information in certain cases. For example, the ThalesGroup website, included in the dataset, presented its license information in a PDF file. Consequently, the tool was unable to extract any relevant data. Additionally, complex HTML structures can prevent the software from navigating to the page containing the license information, further limiting its effectiveness.
In summary, while the license-aware web crawling tool demonstrates practical value and usability by offering license type detection, several areas in the license text generation require improvement. Enhancing the tool's ability to recognize diverse presentations of license information without compromising precision, addressing the extraction of irrelevant information, improving text formatting, and overcoming limitations related to non-HTML formats and complex HTML structures are critical for its advancement. These insights provide a foundation for future enhancements to ensure more reliable and comprehensive license detection and extraction.

Figure 1. Rouge-L Precision, Recall and F1 Score



9 DISCUSSION

9.1 DISCUSSION OF QUANTITATIVE ANALYSIS

The quantitative analysis highlighted the practical utility and effectiveness of the tool based on observations from its usage and the detailed case studies of specific websites. The high license type accuracy demonstrates the tool's reliability in

identifying the correct license type, which is essential for adhering to legal standards and ensuring ethical data usage. This reliability is particularly valuable for users needing legally compliant datasets, especially in AI model training and other applications requiring large datasets.

The ROUGE-L scores, while moderate, showed that the tool is effective in extracting structured license information. The higher precision scores indicate accurate identification of relevant text, though variability in recall scores suggests that the tool may sometimes miss certain elements of the license text. This can be attributed to differences in how license information is presented across various websites.

Observations from case studies revealed that the tool generally performs well across different websites. However, its performance can be influenced by the complexity of the site's HTML structure and the clarity of the licensing information. The ability to save extraction results locally enhances usability, allowing users to access previously extracted information quickly without reprocessing, thereby saving time and computational resources because the tool usually takes a few seconds to extract the information.

9.2 DICSUSSION OF QUALITATIVE ANALYSIS

The qualitative analysis reveals both the strengths and limitations of the license-aware web crawling tool. While it effectively detects and extracts license information from websites, several areas need improvement.

A key issue is the tool's reliance on specific keywords, which can result in the omission of important license information. This suggests a need for more advanced text recognition techniques that can handle diverse presentations of license data without compromising precision.

Additionally, the extraction of irrelevant information, such as tables of contents, highlights the need for refining the tool's parsing algorithms. Improving the contextual understanding of the extracted text could enhance the accuracy and relevance of the information.

Formatting issues in the generated text also impact usability, making the extracted data harder to understand. Addressing these formatting problems is essential for ensuring clear and comprehensible outputs.

The tool's inability to handle non-HTML formats, like PDFs, and complex HTML structures is a significant limitation. Expanding the tool's capabilities to support a wider range of formats will enhance its versatility.

In summary, the tool shows promise, but improvements in text recognition, parsing accuracy, text formatting, and format support are necessary to fully realize its potential for reliable license detection and extraction.

9.3 LIMITATIONS

This study has several limitations. Firstly, the accuracy of the tool heavily depends on the structure and clarity of the licensing information provided on the websites. Websites with poorly formatted or ambiguous license data can lead to incorrect detections and extractions. Secondly, complex HTML structures in a website may lead to the tool not finding the license information at all. Thirdly, the current version of the tool does not support the extraction of licensing information from non-HTML formats, such as PDFs or images, which are sometimes used to present license information. Additionally, the tool's performance may vary across different types of websites and licensing terms because it depends on keyword extraction.

9.4 FUTURE WORK

Future work should address the limitations identified in this study. One key area for improvement is enhancing the tool's capability to process and extract licensing information from non-HTML formats, such as PDFs and images. This could involve integrating OCR (Optical Character Recognition) technologies and PDF parsing libraries. Additionally, expanding the tool to support a wider variety of license types and incorporating machine learning algorithms to improve the detection accuracy and adaptability of the tool are important directions for future research.

10 CONCLUSION

In conclusion, the development of a license-aware web crawling tool using NLP techniques addresses a critical need for legally compliant datasets in AI model training. The tool effectively identifies and extracts licensing information from various websites, achieving an average license type accuracy of 100% and moderate ROUGE-L scores, with an F1 score of 0.499, precision of 0.588, and recall of 0.503. This project successfully demonstrates the feasibility of detecting and extracting licensing information from websites, ensuring that AI models can be trained on data that respects copyright laws. The implemented tool not only enhances legal compliance but also promotes ethical data usage, supporting the sustainable advancement of AI technologies. Future improvements and expansions will further solidify the tool's effectiveness and applicability in various domains.

AI USE

During the preparation of this work the author used Chat GPT, a tool created by the Open AI company to help with improving the academic writing quality of the paper. After using this tool, the author reviewed and edited the content with his own words as needed and takes full responsibility for the content of the work.

REFERENCES

1. Khder, Moaiad. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing and its Applications. 13. 145-168. 10.15849/IJASCA.211128.11.

2. Gray, C.C. (2021). Ethical Concerns Surrounding Web Scraping & Internet Data. Bangor University, 2021.
3. J. Quang, "Does Training AI Violate Copyright Law?" Berkeley Tech. LJ, vol. 36, 2021.
4. S. Xu, Y. Gao, L. Fan, Z. Liu, Y. Liu, and H. Ji, "LiDetector: License Incompatibility Detection for Open Source Software," ACM Transactions on Software Engineering and Methodology, 2023.

5. On, B.-W., Jo, J., Shin, H., Gim, J., Choi, G., & Jung, S.-M. (2021). Efficient Sentiment-aware Web Crawling Methods for Constructing Sentiment Dictionary. IEEE Access.
6. Habernal, I., Zayed, O., & Gurevych, I. (2016). C4Corpus: Multilingual Web-size Corpus with Free License. International Conference on Language Resources and Evaluation.
7. Brenning, A., & Henn, S. (2023). Web scraping: a promising tool for geographic data acquisition.
8. M. Dinzinger and M. Granitzer, "A Longitudinal Study of Content Control Mechanisms," Companion Proceedings of the ACM on Web Conference, 2024.

Appendix

Figure 2. License Extraction Tool Class Diagram

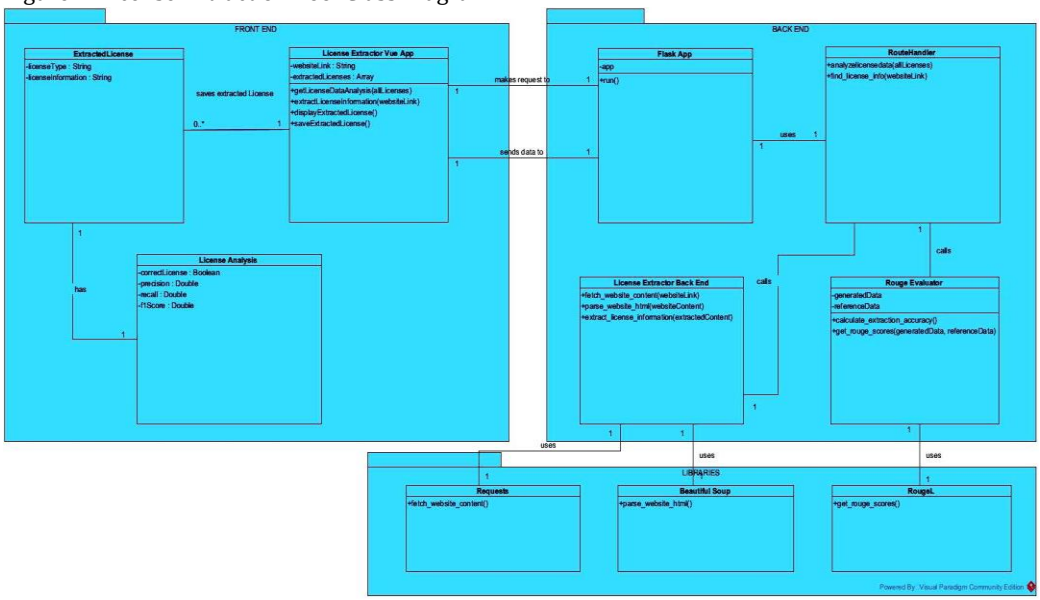


Figure 3. License Extraction Tool Sequence Diagram

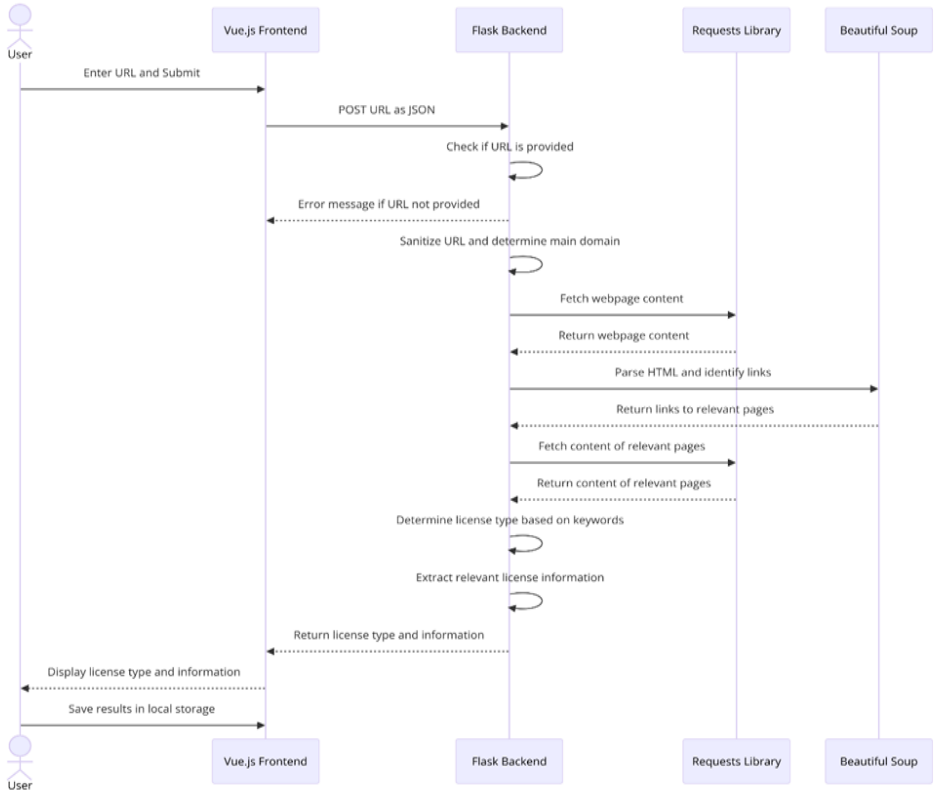


Figure 4. License Extraction Tool UI

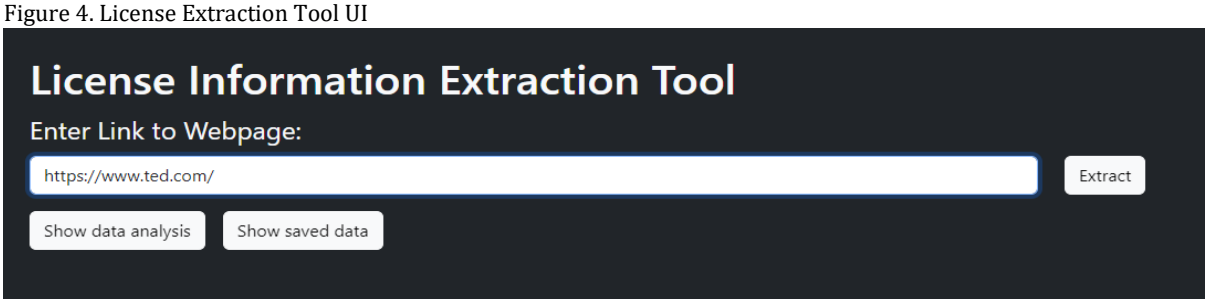


Figure 5. License Type Results for Ted.com

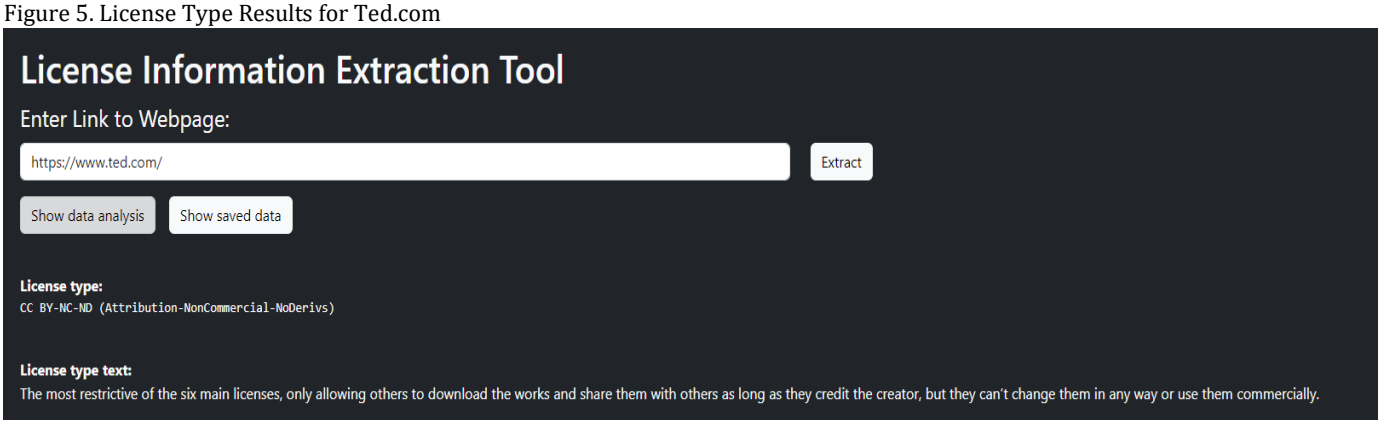


Figure 6. License Text Results for Ted.com