Citation Context Extraction and Analysis of Legal Texts

Sai Bhargava Grandhe  
 *college of information*   
 *university of north texas* denton,texas,usa  
sai.bhargava.grandhe@my.unt.edu

Pravallika Bollu  
 *college of information*   
 *university of north texas* denton,texas,usa  
pravallikabollu@my.unt.edu

Sri Varshini Pratapaneni  
 *college of information*   
 *university of north texas* denton,texas,usa  
srivarshinipratapaneni@my.unt.edu

Sudheendra Mayda  
 *college of information*   
 *university of north texas* denton,texas,usa  
Sudheendramayda@my.unt.edu

Siritha Kantamaneni  
 *college of information*   
 *university of north texas* denton,texas,usa  
sirithakantamaneni@my.unt.edu

*Abstract*—In this paper we extracted the citation context using text analytics. The cases were scrapped using web automation and then using NLP contexts were extracted to give the relevant context. For the purpose of web automation, we used selenium to get the relevant page links and then we used html parser with beautiful soup to get the paragraphs. Once we have the paragraph we find relevant paragraph using fuzzy match.

*GitHubLink: https://github.com/bhargav1237874/grandhe\_INFO5731\_spring2021/blob/main/Citation\_context\_extraction.ipynb*

Keywords—Natural Language Processing, web automation, citation extraction, Legal texts, selenium, Levenshtein distance calculation

# Introduction

Text analytics, also known as text mining, is the procedure of inspecting large collections of resources for producing new information and convert unstructured texts into structured data for usage in additional analysis. Text analytics recognizes the assertions, relationships and facts that are buried in the mass of documented big data. Facts of text analytics are turned and extracted in the structured data for refinement by using the machine learning, integration with structured data in warehouses or databases and visualization and analysis.

Most developed text analytics software use the sophisticated Natural Language Processing (NLP) algorithms. NLP permits the software to identify similar concepts and express in different ways. Machine learning and NLP are branches of artificial intelligence. Machine learning techniques can be utilized in helping natural language processing tasks. Natural language processing can be used to improve machine learning, mainly by utilizing it to remove the biggest evidence base of structured data for learning the machine learning algorithms.

Natural language processing includes natural language understanding and natural language generation which helps the machines to read the text by simulating the human ability. Whereas machine learning is an automatic technology to learn from experience than using an explicit programming.

We perform web scraping technique in the processing stage. It is used to extract the raw data from the website which is taken as the input for the further process. Web scraping can also be done manually but here we need to extract the data automatically for accurate results at faster rate and also, they are of less cost. It involves fetching the data, nothing but downloading the webpage and extracting from it.

# Literature Review

Case-Law plays a major role in decision making and legal reasoning pertaining to International law. When lawyers find any precedent, they make a citation towards that precedent. These citations are part of legal analysis as they support the argument legally. The links to the citations are added manually to the search engines earlier, but now there are many methods implemented to extract the information like named entity recognition, regular expression-based recognition, etc. The results of these methods are highly achievable compared to the manual information extraction systems and also the manual process is a time-consuming one.

The databases can be of structured format or unstructured format. To extract the legal citation content, it is necessary to know the structure of International law and also the type of references within each document. There are two types of legal documents:

1. Standard Legal documents: These are used in courts by lawyers.

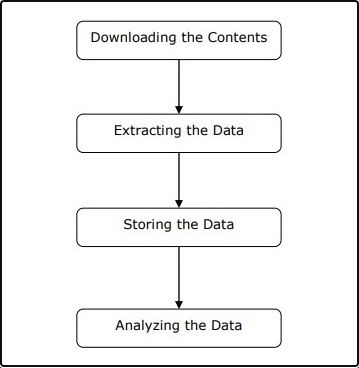
2. Academic Legal documents: These are used in the publication of law reviews.

There are different statutes, rules, and regulations that we need to know before extracting the information. We have federal statutes in which the citation should include the official name of the statute, published source, and the publication date or the year the statute is passed. Whereas, in state statutes, the elements include the name of the code, section number, and the year it is published. Coming to the rules, we need to indicate the abbreviation of the source, followed by the rule number without any comma in between. In the case of restatements, we need to indicate the title of the restatement cited, followed by the number of the section, the name of the publisher and the year published.

One more important thing here is to select the dataset for which we are working on. We need to collect the data and then process it. The data preprocessing includes the following steps: Data cleaning, Data Integration, Data Transformation, Data Reduction, and data Discretization, which in detail include removing the duplicates, fixing the structural errors, handling the missing data, validation. According to the annotation, the file consists of sentences with references or without references. In order to teach the classifier to identify whether the sentence is with a reference or without references, the Term frequency-inverse document frequency feature model was implemented or syntactic textual feature was implemented which is also known as POS(Parts of speech) tags.

The dataset we collected is related to slavery laws. American slavery generated thousands of legal disputes. Lawyers legitimized slavery by fitting cases involving enslaved people into standard legal categories. The law of slavery become an important part of American law. Such support from lawyers helped slave commerce function. American courts also directly participated in slave commerce, frequently auctioning enslaved people to satisfy debts. These sales even took place on courthouse steps. Courthouse auctions forcibly separated families. They helped enslavers to concentrate economic power.

The entire process can be explained in brief as collecting the dataset and scraping the required data then cleaning the raw data and preprocessing it for the required output. This can be simply shown as:



# Methodology

The focus of this research is to achieve automatic identification of legal judgements within legal texts, using Natural Language Processing. For such a purpose, it is necessary to attain a precise knowledge of the structure of International Law as well as the type of references within each document. International Law practitioners not only refer to Treaties and Conventions between countries, but also Cases, Decisions and Awards related to the Treaties and Conventions. Consequently, the links within each of the above documents are mainly external links with legal instruments as their target.

An anchor is the string of characters that marks the occurrence of a citation in the running-text of a paper, such as “(Fireman 2007)” or “[57]”.2 The sentence that this anchor resides within is then the anchor sentence. The citation continues from before and after this anchor as long as the text continues to refer to the cited work; this block of text may span more than a single sentence. We introduce the citation-site, or c-site for short, to represent this block of text that discusses the cited work. Since more than once sentence may discuss the cited work, each of these sentences is called a c-site sentence. For clarity will also call the anchor the c-site anchor henceforth. A citing paper contains the c-site that refers to the cited paper. Finally, the reference at the end of the paper provides details about a c-site anchor (and the c-site).

1.Implementation: The first step that we follow here is to collect various links of the cases through an automated process. This generates the link address for various cases which was later scrapped and saved in a pickle file.



The final output looks like this where the first column after the index is the case which cites the case in third column. The passage was extracted after the scrapped data in second column where we separated the data into paragraphs and then clustered the data using ml technique to a fuzzy match to find the cited context. Which is extracted as a final outcome in fifth column. In the web automation we used various methods like click and passage save to save the final outcome.

2. Theoretical Background: Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents and other electronically represented sources. In most of the cases this activity concerns processing human language texts by means of natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video/documents could be seen as information extraction.

A broad goal of IE is to allow computation to be done on the previously unstructured data. A more specific goal is to allow logical reasoning to draw inferences based on the logical content of the input data. Structured data is semantically well-defined data from a chosen target domain, interpreted with respect to category and context.

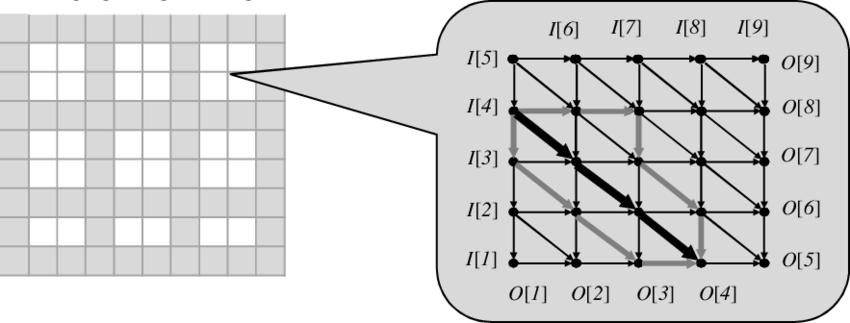
Information Extraction is the part of a greater puzzle which deals with the problem of devising automatic methods for text management, beyond its transmission, storage and display. The discipline of information retrieval (IR) has developed automatic methods, typically of a statistical flavor, for indexing large document collections and classifying documents.

Another complementary approach is that of natural language processing (NLP) which has solved the problem of modelling human language processing with considerable success when considering the magnitude of the task. In terms of both difficulty and emphasis, IE deals with tasks in between both IR and NLP. In terms of input, IE assumes the existence of a set of documents in which each document follows a template, i.e. describes one or more entities or events in a manner that is similar to those in other documents but differing in the details. An example, consider a group of newswire articles on Latin American terrorism with each article presumed to be based upon one or more terroristic acts. We also define for any given IE task a template, which is a(or a set of) case frame(s) to hold the information contained in a single document. For the terrorism example, a template would have slots corresponding to the perpetrator, victim, and weapon of the terroristic act, and the date on which the event happened. An IE system for this problem is required to “understand” an attack article only enough to find data corresponding to the slots in this template.

# Data collection and pre-processing

The idea is to extract the citation context of a case which is referring from another case. We also want to automate the entire process using NLP so that for new citations we do not have to do it manually. We used context extraction with web-

automation to do the same. The flow is basically like this, first we extract the links where we will can find the page where the case details is available which is referring to a case. This pages does not give the information directly we need to go to the more info page button and then click it to refer us to the page where we can find the case details. We saved all the page links where we can find the case links which is referring a case. The mapping will be a dictionary where the key is a case which is being referred and the value is list of cases details page links which is referring. We extract this page info using html extractor to give us complete case details. Once we have the case in text format we can use our text analytics method to extract it. We use fuzzy match logic to match the context using levenshtein distance. Levenshtein distance is a text similarity measure that compares two words and returns a numeric value representing the distance between them. The distance reflects the total number of single-character edits required to transform one word into another. The more similar the two words are the less distance between them, and vice versa.



Levenshtein distance calculation: One common use for this distance is in the autocompletion or autocorrection features of text processors or chat applications. The highest match paragraph we keep as our final output of the context extraction. Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It is functional for all browsers, works on all major OS and its scripts are written in various languages i.e. Python, Java, C#, etc., we used Python here.

Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It is functional for all browsers, works on all major OS and its scripts are written in various languages i.e. Python, Java, C#, etc., we used Python here.

Why Selenium Python?  
1. Open Source and Portable – Selenium is an open source and portable Web testing Framework.

2. Combination of toll and DSL: Selenium is combination of tools and DSL (Domain Specific Language) in order to carry out various types of tests.

3. Easier to understand and implement – Selenium commands are categorized in terms of different classes which make it easier to understand and implement.

4. Less burden and stress for testers – As mentioned above, the amount of time required to do testing repeated test scenarios on each and every new build is reduced to zero, almost. Hence, the burden of tester gets reduced.

5. Cost reduction for the Business Clients – The Business needs to pay the testers their salary, which is saved using automation testing tool. The automation not only saves time but gets cost benefits too, to the business.

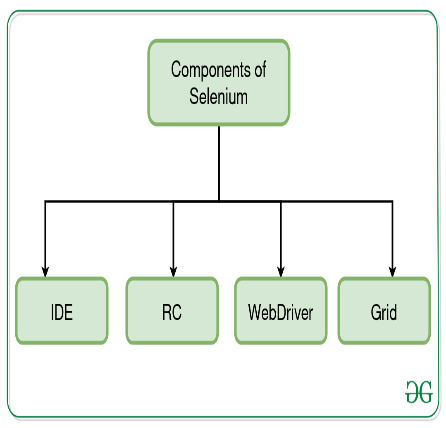
Selenium is a powerful tool for controlling web browser through program. It is functional for all browsers, works on all major OS and its scripts are written in various languages i.e. Python, Java, C#, etc., we will be working with Python. Selenium has four major components – Selenium IDE, Selenium RC, Selenium Web driver, Selenium GRID.

There are three major approaches for unsupervised learning: -

(a) Competitive Learning

(b) Self Organizing feature Maps

(c) ART Networks.



Components of Selenium:

Selenium has been in the industry for a long time and used by automation testers all around the globe.

The four major components of Selenium –

Selenium IDE

Selenium RC

Selenium Web driver

Selenium GRID

**Selenium IDE** (Integrated Development Environment) is the major tool in the Selenium Suite. It is a complete integrated development environment (IDE) for Selenium tests. It is implemented as a Firefox Add-On and as a Chrome Extension. It allows for recording, editing and debugging of functional tests. It was previously known as Selenium Recorder. Selenium-IDE was originally created by Shinya Kasatani and donated to the Selenium project in 2006.

Selenium IDE was previously little-maintained. Selenium IDE began being actively maintained in 2018 Scripts may be automatically recorded and edited manually providing autocompletion support and the ability to move commands around quickly. Scripts are recorded in Selenese, a special test scripting language for Selenium. Selenese provides commands for performing actions in a browser (click a link, select an option) and for retrieving data from the resulting pages.

**Selenium RC (Remote Control)**

Selenium Remote Control (RC) is a server, written in Java, that accepts commands for the browser via HTTP. RC makes it possible to write automated tests for a web application in any programming language, which allows for better integration of Selenium in existing unit test frameworks. To make writing tests easier, Selenium project currently provides client drivers for PHP, Python, Ruby, .NET, Perl and Java. The Java driver can also be used with JavaScript (via the Rhino engine). An instance of selenium RC server is needed to launch html test case – which means that the port should be different for each parallel run. However, for Java/PHP test case only one Selenium RC instance needs to be running continuously.

**Selenium Web Driver**

Selenium WebDriver is the successor to Selenium RC. Selenium WebDriver accepts commands (sent in Selenese, or via a Client API) and sends them to a browser. This is implemented through a browser-specific browser driver, which sends commands to a browser and retrieves results. Most browser drivers actually launch and access a browser application (such as Firefox, Google Chrome, Internet Explorer, Safari, or Microsoft Edge); there is also an HTML Unit browser driver, which simulates a browser using the headless browser HTML Unit.

Selenium WebDriver does not need a special server to execute tests. Instead, the WebDriver directly starts a browser instance and controls it. However, Selenium Grid can be used with WebDriver to execute tests on remote systems (see below). Where possible, WebDriver uses native operating system level functionality rather than browser-based JavaScript commands to drive the browser. This bypasses problems with subtle differences between native and JavaScript commands, including security restrictions.

**Selenium GRID**

Selenium Grid is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the hub. Tests contact the hub to obtain access to browser instances. The hub has a list of servers that provide access to browser instances (WebDriver nodes), and lets tests use these instances. Selenium Grid allows running tests in parallel on multiple machines and to manage different browser versions and browser configurations centrally (instead of in each individual test).

The ability to run tests on remote browser instances is useful to spread the load of testing across several machines and to run tests in browsers running on different platforms or operating systems. The latter is particularly useful in cases where not all browsers to be used for testing can run on the same platform.

The term text processing refers to the automation of analyzing electronic text. This allows machine learning models to get structured information about the text to use for analysis, manipulation of the text, or to generate new text. Text processing is one of the most common tasks used in machine learning applications such as language translation, sentiment analysis, spam filtering, and many others.

Text processing refers to only the analysis, manipulation, and generation of text, while natural language processing refers to the ability of a computer to understand human language in a valuable way. Basically, natural language processing is the next step after text processing.



Since text processing is one of the machine learning uses that average technology consumers don’t even realize they’re using, but most people use apps daily that are using text processing behind the scenes.

Text data can show a business how their customers search, buy, and interact with their brand, products, and competitors online. Text processing with machine learning allows enterprises to handle these large amounts of text data.

Topic analysis – This technique interprets and categorizes large collections of text into topics or themes.

Intent detection – This classification model detects the intent, purpose, or goal of the text. For example, it may determine whether the intent is to gain information, make a purchase, or unsubscribe from the company.

Language classification – This classifies text based on which language it’s written in.

In background,

**Hardware requirements**:   
1. I5 Windows System  
2. 4 core CPU machine

**Software requirements**:  
1. Python 3.7  
2. Anaconda

3. PyCharm  
4. Chrome Driver

##### experiment

It is important to begin with an understanding of the structure of Legal texts pertaining to International Law, as well as how they may be tailored into an appropriate format to achieve the extraction of links. The overall analysis mainly focuses on the use of Web Scrapping on the Legal texts for automatic extraction of each of the mentioned references within the data as an annotation. . Mainly, it consists of extracting the text of the document from its original format (often a PDF), checking it, and entrusting it to several algorithms which will recognize its structure (articles, paragraphs, table of contents, footnotes and more). This also enables the next steps of qualifying the data. Qualifying the data means identifying legal concepts held within such a given set. Such steps result from artificial intelligence but also from the participation of lawyers. This is followed by enabling interconnection, which leads to this research study.

The interconnection of references is a key-feature for lawyers: in one click, they should be able to access all documents relating to an article of a treaty or a paragraph of a judgment/award. The data is originally obtained in an HTML format. For the purpose of this analysis, the data is loaded as HTML text in a “.txt” format. Due to this, the text also includes HTML tags, which will need to be included in the analysis and JAPE rules. There exists an alternative of dealing with such data in “.html” format. However, in such a case the data would appear in a text format without any HTML tags. The start position and end position numbers of an identified link is counted as per the number of letters in the document. This position will differ in both formats, due to the HTML tags present in the text format which are also included in the count of position number. It is vital to include the tags in the count for detecting the position to support the later integration of the output with the back-office of the search engine, wherein the data is stored in the original HTML text format. It can hence be justified to operate with the HTML data in a text format for further analysis

A screenshot of a computer

Description automatically generated

The dataset consists of multiple articles but in this project we have considered only few of them for extraction to show the use of Natural Language processing on legal texts. The citations under short name is the actual case which has been cited by several cases and provided judgements. The dataset clearly provides the references of the legal cases for which the citation has been used for making the judgements. The judgement made consists of a different context for all the references. As the citation context is different at multiple instances it’s a tough task for attorneys to go through each and every reference and make judgement for other cases. The context extraction provides all the context that has been cited which helps the attorneys to provide relevant judgement based on the case.

The steps involved in extracting the citation are:

1. First, we need to read the dataset into the csv file.
2. Read all the links using chrome driver.
3. Read the content from the links.
4. We need to create a chunk of files to read.
5. Start the file creation.
6. Collating all the files.
7. Gathering all the link texts

all\_data=[]

for i in range(start,end+1):

    try:

        file = open(str(i)+"scapped\_data.pickle",'rb')

        object\_file = pickle.load(file)

    except:

        pass

    data=(pd.DataFrame.from\_dict(object\_file,orient='index'))

    all\_data.append(data)

pd.concat(all\_data).to\_excel('scrapped\_data.xlsx')

-Here, all the citations that are been referenced by are extracted using these lines of code. Hence, the further extractions of the citations are made easy by extracting only these referenced by content.

all\_data=pd.read\_excel('scrapped\_data.xlsx')

After all the cleaning and preprocessing is done, This line of code is the dataset that is used for this project.

all\_data.rename(columns={'Unnamed: 0':'col\_1',0:'col\_2'},inplace=True)

The columns names are being assigned such that future usage of these columns will be clear and easy.

all\_data=all\_data.dropna()

all\_data

This prints the data of the all references and need to work on this output for final extractions.

all\_dataframe=[]

for key,value in object\_file.items():

    all\_items=[]

    for item\_key,item\_value in value.items():

        all\_items.append(item\_key)

    data=pd.DataFrame(all\_items)

    data['col\_3']=key

    all\_dataframe.append(data)

    data.rename(columns={0:'col\_1'},inplace=True)

final\_data=pd.concat(all\_dataframe)

These lines of code are the ones that extracts the citations from the references we had in the previous output and these are used to extract the contexts that are related to these citations.

def string\_bool(list\_all,item):

    for each\_item in list\_all:

        if item in each\_item:

            return (True,each\_item)

    return (False,'')

This code is used to extract the context of the citations that are stated in that respective reports.

merged\_data.to\_excel('final\_output.xlsx')

This is where we merge all the columns into the excel which will be having 5 columns, col\_1, scrapped data, col\_3, cited and paragraph.

##### results and discussion

The final output seems like this where a case in the third column is the short name of citation after the index. Following the scrapped data in the full name of citation column, we extracted the data and clustered the data in paragraphs using the ML-technique to fusion to find the above-named context. That is extracted in the context of citation column as the result. We also have a column to represent whether it is cited or not and the final output in the column 5 named as paragraph depends on it. For the true cases we get the output. We used various methods in web automation, such as click and save passage, to save the result.

APPLICATIONS

The applications are endless and some of them are mentioned below:

1. Risk Management: No matter the industry, Insufficient risk analysis is often a leading cause of failure. This is especially true in the financial industry where adoption of Risk Management Software based on text mining technology can dramatically increase the ability to mitigate risk, enabling complete management of thousands of sources and petabytes of text documents, and providing the ability to link together information and be able to access the right information at the right time.

2. Knowledge Management: Not being able to find important information quickly is always a challenge when managing large volumes of text documents—just ask anyone in the healthcare industry. Here, organizations are challenged with a tremendous amount of information—decades of research in genomics and molecular techniques, for example, as well as volumes of clinical patient data—that could potentially be useful for their largest profit center: new product development. Here, knowledge management software based on text mining offer a clear and reliable solution for the “info-glut” problem.

3. Cybercrime Prevention: The anonymous nature of the internet and the many communication features operated through it contribute to the increased risk of internet-based crimes. Today, text mining intelligence and anti-crime applications are making internet crime prevention easier for any enterprise and law enforcement or intelligence agencies.

##### Conclusions

Finally, we obtain the extracted information from the different slavery cases and citation context from each article is extracted. The same can be integrated with other web app systems to automatically extract the context of a citation.

AUTHORS CONTRIBUTION

SAI BHARGAVA GRANDHE: My contribution to this project is to make sure the right data is captured from different sources and taking suggestions from teammates, sharing my insights towards data collection and data cleaning processes and helping them in multiple instances, and also formatting the project details.

SUDHEENDRA MAYDA: My most significant contribution to my final project came in the form of introducing the methodology for my project. This type of contribution means that I spent my time working on inspiring my group mates and encouraging effective communication and creating a comfortable environment by helping my group members to ask the right questions. I am considered a blue-sky thinker and like to promote this way of thinking with my team.

RAMA SIRITHA KANTAMNENI: My contribution for this project is collecting the resources for the project, determining data analysis and some reference documents which were helpful for the work and help others to study more about the project, formatting the document.

PRAVALLIKA BOLLU: My contribution to this project is Literature Review. I have referred to different papers and websites to understand each and every part of this project. Communicated the useful information to the team members and analyzed in detail about the project.

SRIVARSHINI PRATAPANENI: My contribution to this project is working on citation context and research goals. The end goal works on the accuracy of the project. Many law websites and journals are used as the reference for understanding this project in detail. By gathering all the information it has been possible to sum up everything.

##### References

1. <https://www.researchgate.net/publication/228826654_Text_Analytics_for_Dummies>
2. <https://www.researchgate.net/publication/336239322_Extraction_of_Legal_Citations_using_Natural_Language_Processing>
3. <https://www.researchgate.net/publication/336239322_Extraction_of_Legal_Citations_using_Natural_Language_Processing>
4. <https://www.cl.c.titech.ac.jp/_media/publication/645.pdf>
5. <https://link.springer.com/chapter/10.1007/978-3-030-03496-2_14>
6. <https://epubs.siam.org/doi/abs/10.1137/1.9781611973440.41>
7. <https://academic.oup.com/dsh/article-abstract/29/3/452/2938091>
8. <https://case.law/gallery/>
9. <https://case.law/download/citation_graph/>
10. <http://www.citingslavery.org/>
11. <http://www.citingslavery.org/court_cases>
12. <http://www.citingslavery.org/court_cases/540>