Kent School of Law Project

Antoine Gargot A20410860, Student, IIT

[1]IIT, Illinois Institute of Technology, 3300 S Federal St, Chicago, IL 60616

Abstract – Nowadays, machine learning and data science become a useful tool for most of the business surrounding our life thanks to our current ability to store huge amount of data. The United States of America government digitalized their trail for each jurisdiction in the country. What can we do with this data, is it possible to extract information of each of them in order to create an accurate model predicting the outcome of a case. This model will be really useful for people and lawyers who want to avoid unnecessary trial and lose money at the same time. In this project, we will reply to this huge question and try to create a reliable database and set of features in order to work on a machine learning model.

I. INTRODUCTION

A - What we seek to address

The main propose of this project is to work among a huge data set from the Legal project of the Kent School of Law in order to find pattern through this data and analyze potential verdict of from cases. The Legal project team of the Kent School of law will try to find a pattern in past judge verdict based on several criteria in order to link a judge to a specific class regarding the profile of the defendant. The Legal project team asked for help from Illinois Institute of technology students in order to bring data science knowledge and practice inside their project in order to understand what can be made for their project and study potentials from this idea.

The goal of this project is to work on unstructured data by doing information extraction and create a structured dataset and working with this data for different experiment :

* Opinions Graph Analysis in order to study the link between each case of different trials.
* Sentiment Analysis in order to extract some useful information over the outcome of a trial.
* Cluster Analysis to look into the correlation between some cases and try to cluster most of them thanks to machine learning technique.
* Outcome prediction based on different extracted features and on previous experiments.

B - Methodology

The IIT team will tend to understand the data first of all. After an overall understanding of the data set and cleaning the data set, we will find the most representative feature in each observation in the API and the database.

We have, through this project, to improve the process of identifying the opinion (decision of the court) and citations (there can be more than one for each opinion).

We will work on data overview thanks to representation as a graph of opinion and citations.

We will try to find the best model in order find patterns in our dataset based on what we studied during our Data Mining course but also with personal knowledge which we acquired during.

II. DATA OVERVIEW

The data come from the huge data base from the government. *CourtListerner* is a web service that provides more than 3 million document from different cases in different jurisdiction all over the United States. In addition to written reports from cases, we can find useful information about judges in the US and their political affiliation and more than 800,000 audio records from different cases.

Free Law Project is a United States federal 501(c) (3) nonprofit that provides free access to primary legal materials, develops legal research tools, and supports academic research on legal corpora. This federal institution was in the initiative of the CourtListener project.

In all those different cases, there are 419 jurisdictions which control each case. We are also able to extract the data based on a specific jurisdiction. The specificity is that form all different jurisdiction, each report has their own format and way to express information such as people involved in the case or the outcome. Here is the example of how the information can be expressed in a report :

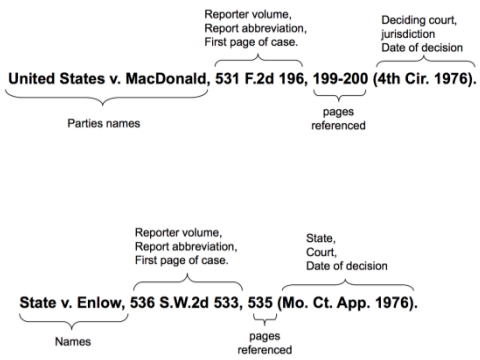

FIG 1 : STRUCTURE OF DIFFERENT CITATIONS

When there is a citation of another case, we can find this kind of format in order to know where to find the citation, this information is composed of 4 different parts :

* The parties names which express the person or the organization involved in the case
* The reporter volume, abbreviation and first page in order to show where to find this specific case.
* The page reference in order to know where to find the citation in the report volume
* The state and the court involved in the case and also the final date of the decision.

In this small example, you can see how the information is represented differently among different cases.

That kind of information was extracted thanks to optical characters recognition in order to extract the text from legal documents. This information was represented in different formats :

* A plain text with all the unstructured data
* An HTML file with some information of each part of the document

Some recent ones were directly extracted based on the source code of generated PDFs.

In order to access the data, we used a recent project were each opinion can be accessed by calling a rest API. Each observation is represented as a JSON file composed of 21 features, it appears that the extracted data is unclean most of the time, most of the feature values are missing or the format of them is incorrect (the type of the feature is not uniform among all the observations).

Here is an overview of features that we found in the different JSON object :

* resource_uri which gives the link to the case in the rest API
* absolute_url which gives you the source of the case
* cluster which gives you the cluster of the case among all cases in the jurisdiction based on an experiment of the previous project.
* author which is the author of the case
* joined_by, this feature is unknown, we didn't find values for it.
* author_str which is the string format of the author
* per_curial, this feature is unknown, we didn't find values for it.
* date_created which is the date of the creation of the case
* date_modified which is the last date of modification of the case
* type which is the type of trial
* sha1, this feature is unknown, we didn't find an explanation for it.
* page_count which is the number of pages in the original document
* download_url which is the link to download the original document
* local_path which is the path of the document inside the local resource
* plain_text which is the plain text extracted from the document (without attributes on different part)
* HTML which is the HTML representation of the original document with some useful attributes. We can find several HTML fields based on the following feature name :
  * html_lawbox
  * html_columbia
  * html_with_citations
* extracted_by_ocr which is a boolean expressing the way that the data has been extracted from the original document
* opinions_cited which is the IDs of the different opinions cited in this observation.

Most of those features were useless for our project and we wanted to work from scratch by doing information extraction from reliable features such as HTML and plain text. We had to work on both features regarding the fact that sometimes one feature was missing in some cases.

There are 4,085,026 opinions on JSON documents representation (from the supreme court dataset). We finally found that all record where available as download archive in the court listener website. Before working with the whole data set (which is over 34 GB compressed), we work on a sample of it. The data is split by jurisdiction (419 courts in totals). We decided to test our data processing algorithm and model with a small jurisdiction data base (like the supreme court) which is only few GB of data composed of 63981 observations.

For this project, regarding the fact that we had limited resources, we worked on a sample of the data, taking into account the SCOTUS Supreme Court of the United States and a sample another sample of different jurisdictions.

III. EXPERIMENTS

For this project, we worked on three different experiments in order to find as many features and pattern as possible from our data. First, our main goal was to find the best way to extract features from the 2 possible data sources that we had (HTML and plain text) using different known algorithm and models. Secondly, we work on a new way to cluster opinions inside the database, based on different algorithms that we learned in class. Last, we work on outcome extraction and observation labeling in order to have a label data set to fit a supervised model in future projects.

IV. FEATURES EXTRACTION

A - HTML scraping

In order to find features among our different texts, we used beautiful soup (BS4) a Python library which helps us to scrap the HTML feature. This known library is commonly used for online scrapping but can be easily used for HTML scraping as well. The HTML feature was composed of several different CSS class which helped us to extract specific features. From this HTML we were able to scrap some information such as :

* case_cite which represent ids of different cases cited in this specific observation

* parties which represent the person or entities involved in this case (this representation is formed with the following format offenders vs. defendant).

* docket which is a specific id for the case.

* court which is the current court involved in this case.

* date which are the different dates of the case, most of the time there are represented with the description of that date (creation of the case, decision date …)

* indent which is the summary of the people involved in the case and the different big decisions in the case, from this feature we are able to fetch the outcome of the trail.

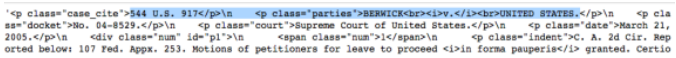

FIG 2 : SAMPLE OF HTML FEATURE

Most of the time, when we scraped the data, we had to transform it regarding the fact that in one tag, several informations were found. At the end of the experiment, we were able to extract useful information such as judges, dates, persons involved in the case, cases cited in this trial (represented as arrays of case ids), verdict and different citation as text.

	opinion_number	filed_dates	judges	opinion_judge	verdict	citations
1	[09-15069]	March 31, 2010	Betty B. Fletcher, Richard R. Clifton and\n ...	[Judge, Bea]	[REVERSED, REMANDED]	[v. FrontierPac. Aircraft Indus., Inc., 813 F...
2	[09-15483]	No reliable dates where extracted	Judges couldn't be extracted.	[Couldn't find the opinion judge]	[]	[v.Selecky, 586 F.3d 1109, 1119 (9th Cir. 200...
3	[08-30050]	April 1, 2010	Thomas M. Reavley* Richard C. Tallman, and\n ...	[Judge, Milan]	[AFFIRMED]	[v. Norwood, 555 F.3d 1061 (9th Cir. 2009), ...

FIG 3 : NEW FEATURES AFTER SCRAPING HTML

B - Plain text extraction

Regarding the fact that we had some missing value for some cases in the different HTML fields, we had to extract information directly based on the structure of the text itself. For doing so, we used a known website which lists all possible structure of texts for different jurisdiction: [Basic Legal Citation](https://www.law.cornell.edu/citation/3-200). Based on those structures, we implemented an algorithm in order to extract information from a document based on its jurisdiction. The text was structured as below :

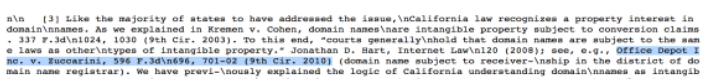

FIG 4 : SAMPLE OF PLAINTEXT FEATURE

Thanks to this new approach, we were able to fetch new features from observation. However, the text data was not well structured or the extraction of the data didn't go well for some observation and didn't follow the initial pattern from its jurisdiction. That's why we worked on a third approach to solve this issue.

C - Name entities recognition

We decided to work on natural language processing in order to extract more information for a specific text. I used for this approach the known Standard openNLP model in order to work on name entities recognition. For extracting data from a text, we can use different approaches such as Name entities recognition or part of speech tagging, the POS tagging is a well-known method that consists of tagging each word in a sentence based on its grammatical role (verb, subject, attributes …). For doing so, there are many different algorithms such as the CYK algorithm, Hidden Markov Model or Viterbi algorithm. In this project, we chose to only work with name entities recognition which enabled us to extract different features from all the sentences in a case (localization, organization, person, dates, misc …). We used the library openNLP written in Java and we converted it in R in order to work with it. We create tokens for each sentence in order to parse each of them. Thanks to maxent entity annotator, we were able to create an array of entities for each observation. We were finally able to retrieve this kind of data from each text :

Citation id = 4654

Person	'Webb' 'John Sidney Webb' 'Reed' 'Campbell' 'Smith' 'Treat' 'Kennedy' 'Beck' 'Hunt' 'Simpson' 'Leavitt' 'Beck' 'Johnson' 'Spencer' 'Gray' 'Locke'
Location	'Maryland' 'District of Columbia' 'St.' 'New York' 'Kennedy' 'New York' 'District of Columbia' 'Pitchard' 'New Jersey' 'New York' 'Mass.' 'Atwater' 'Conn.' 'Pa.' 'Ohio' 'District of Columbia' 'District of Columbia' 'District of Columbia' 'District of Columbia' 'District of Columbia' 'United States' 'Mass.' 'West Roxbury'
Organization	'Burr' 'Barnabas Hospital' 'Barnabas Hospital' 'Lodge'
Data	'1715' 'three years'

FIG 5 : ENTITIES EXTRACTED FOR A SPECIFIC CASE

Thanks to those 3 different approaches, we were able to fetch the entire text of the case, such as information and the proceedings of the case. That data will help us to have description on each party (ethnic description, previous activity, personal information such as age).

In addition to those experiment, we found previous research based on Natural Language Processing and features extraction from legal text called the lexpredict project : www.lexpredict.com.

In this project, researchers made several tools in order to work with legal documents for data extraction :

- Contraxsuite, which is a leading open-source contract analytics and legal document analysis platform. ContraxSuite can identify legal material, extract information, and organize, analyze, de-duplicate, and create visualizations of your data.

- Lexsemble, which is a powerful tool for guiding strategic decisions by tapping hidden human capital. LexSemble incorporates knowledge management, prediction, machine learning, gamification, and other techniques to help your organization run better.

- Lexreserve, which is a software tool for tracking your legal risks from start to finish (this tool will not be useful in our case).

- Lexnlp, which is a natural language processing library for working with real, unstructured legal text, including contracts, plans, policies, procedures, and other materials.

Unfortunately, due to the fact that this project was not open source, we were not able to use their libraries in order to fetch more information in our text based on legal lexicon natural language processing.

After, doing so, we worked on a clustering approach in order to group different cases.

V. CLUSTERING

The main objective of this experiment was to group different trail based on the citation that they had. Most of the time, on a specific trial, we had several cited cases. For working on opinions clustering, we used the networkX library in Python in order to display and compute our graph. We created an oriented graph which was composed as below :

* Node for each case that we studied in our sample (noted in red)
* Edge for a specific opinion when we found a citation of it in the case.
* New nodes were created from those edges which were opinions cited outside the jurisdiction or the sample (noted in blue)

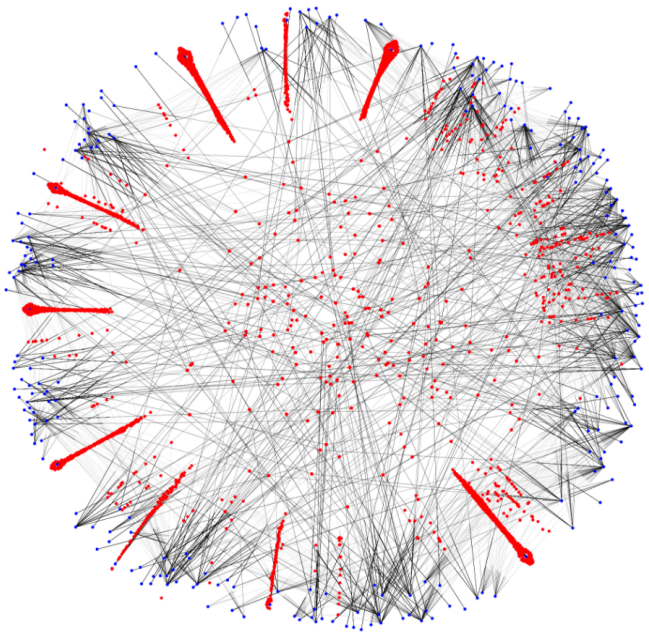

FIG 6 : NETWORK REPRESENTATION FOR OUR CITATIONS

We can see in our graph that some cluster seems to emerge. Some opinions have a huge cluster coefficient based on the fact that they are well cited in several different cases.

In order to find clusters in this graph, we used an algorithm studied in class: the spectral clustering. Regarding the size of the degree matrix we chose to use an optimal way to compute our Laplacian matrix, this matrix was not stochastic because we worked on a directed graph. After the computation of this matrix, we computed its two first eigen vectors.

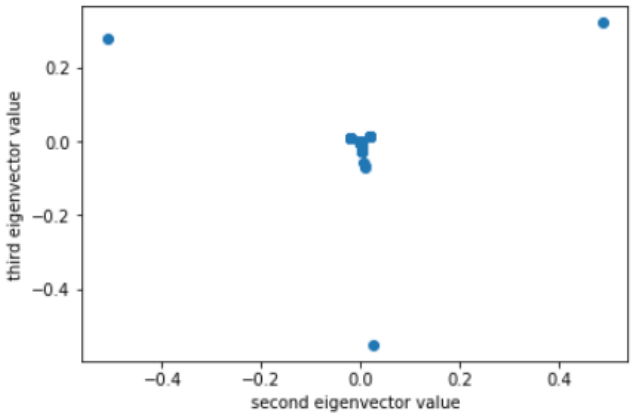

FIG 7 : NETWORK REPRESENTATION OVER 2 EIGENVECTORS

This approach helped us to find a good way to display clusters from our graph. With this representation, we chose to work on the K-means algorithm for all those cases. In order to find the best number of clusters in our graph, we used the Eclust method which consists of the computation of each MSE based on the number of clusters.

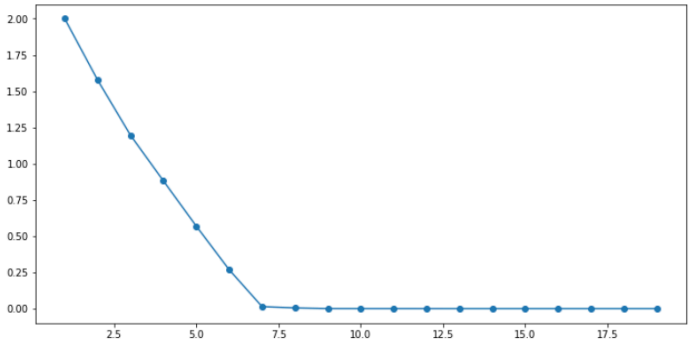

FIG 8 : ECLUST OF THE SPECTRAL CLUSTERING

From this Eclust curve, we can easily extract from the knee of it the optimal number of cluster, which is 7 in our case. The method consists of choosing the number of clusters where the variation of the error begins to stabilize.

cluster_errors	num_clusters
1.999992	1
1.577049	2
1.195764	3
0.881070	4
0.568889	5
0.267930	6
0.013577	7
0.004805	8
0.000421	9

FIG 9 : ECLUST COMPUTATION OF THE SPECTRAL CLUSTERING

After doing these computations, we clustered our observations based on the K-means algorithm using the random point as centroids and assign for each observation a cluster ID. Here is a representation of our final results for this experiment, working on the Supreme Court of the United States.

id	case_cite	cluster_id
143119	[544]	0.0
122028	[536]	0.0
92451	[130, 9, 32]	0.0
134300	[540]	2.0
89793	[97, 24]	0.0

FIG 10 : FINAL RESULTS OF CLUSTERING EXPERIMENT

VI. OUTCOME EXTRACTION / OBSERVATION LABELING

Regarding the fact that our observations were unlabeled, we wanted to find a way to label them depending on the outcome of a specific trial. For this experiment, we worked on outcome extraction. Thanks to openNLP, we made some sentence tokenization in order to list each sentence as a string for indent feature of each observation. After doing so, we worked on sentence extraction based on a set of words in a lexicon related to the outcome of a case such as: "judgment is reversed", "motion", "motions", "denied", "affirmed", "dismissed", "granted" …

After extract the specific sentence related to the outcome we used sentiment analysis in order to label our observation. In order to do so, we first cleaned and tokenize each of our sentences in order to get rid of punctuation, numbers, stop words and spaces. Based on tidy text library and the Sentiword analysis we were able to extract sentiment from each outcome such as follow: "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise" or "trust". Based on those sentiments, we created counters of sentiment for each outcome sentence token such as follow :

For the sentence :

"On the whole, we are of opinion that the decree of the circuit court dismissing the libel of the captors ought to be affirmed and that the cause should be remanded to the circuit court for further proceedings as between the united states and the claimants"

We can extract :

* Two words related to anticipation
* Two words related to trust
* One word which is negative
* And, one word which is positive.

Regarding the different sentiment, we can see that some are oriented as positive and others are oriented as negative. Based on this counter for each sentence, we created a rule in order to compute the final score of our outcome. If the counter of positive, trust and anticipation words is more than the counter of negative, disgust and anger the label of the outcome will be positive otherwise it will be considered as negative.

We observed some issues with this approach depending on the orientation of the sentence, for instance, we can see that the sentence: "for these reasons this action cannot be maintained and the judgment for the defendant must be affirmed" is oriented to be positive regarding the term « affirmed ». However, the sentence is oriented for the defendant and not the offender (which is the orientation of most of the trial report). Our to the orientation of the sentence, it will be misclassified

Based on this approach, we were able to achieve a quite accurate process to label each of our observation in the dataset. At the end of this experiment, we were able to create the following dataset :

	case_cite	parties	docket	court	date	intent	id	outcome	label
4650	['543']	['IN RE VOVAK.']	['No. 04-7300.']	['Supreme Court of United States.']	['January 10, 2005.']	['Petitions for writs of prohibition denied.']	140925	petitions for writs of prohibition denied	-1
4651	['543']	['CULOTTAv.UNITED STATES.']	['No. 04-6542.']	['Supreme Court of United States.']	['November 1, 2004.']	['C. A. 11th Cir. Certiorari denied. Reported below: 99 Fed. Appx. 881.']	139985	certiorari denied	-1

FIG 11 : SAMPLE OF OUTCOME EXTRACTION AND LABELING

VII. ROLE IN THE PROJECT

I contributed to most of the different part of the project. Working first for the overview of the data in order to understand the whole project and the need for the Kent School Project Team. Then I worked with Tomas and Chirag on the data extraction based on the HTML feature of each observation. After seeing that the results were not the best for all observation, I oriented my work to the name entities recognition part alone. After doing so, I worked with Thomas in the Clustering experiment after extracted citation for each case and create a sample data set in order to get into the network experiment. This part was entirely done by Thomas and myself in collaboration. Finally, after our good result regarding the clustering experiment, I worked on the Outcome extraction and observation labeling based on Sentiword dictionary.

VIII. DIFFERENT TOOLS USED IN OUR PROJECT

For this project, we used different R and python library :

For the experiments using R :

• NLP: This library was used in order to annotate our sentence for each observation and also annotate entities, person, organization …

• openNLPmodels.en: This model from Stanford was used in for our annotation based on the NLP package.

• RWeka: Weka is a collection of machine learning algorithms for data mining tasks written in Java, containing tools for data pre-processing, classification, regression, clustering, association rules, and visualization.
• rJava: This package was used in order to import the Stanford model, initially written in JAVA in order to use it in an R environment.
• magrittr: This library was used in order to create pipes to easily compute some data cleaning and tokenization for our experiments.
• stringr: This library was used in order to make some cleaning for our sentence which we wanted to work with.
• tidytext and tidyverse: Those packages were used in order to do some text mining for each of our observations.
• glue: This package is used in order to perform some operation over strings in R.

For the experiments using Python :

• BeautifulSoup: This library was used in order to perform the scraping on the HTML feature for each observation.
• pandas: This library was used in order to manage and edit our dataset from the CSV file that we created.
• numpy: This library was used in order to compute our work for each matrix, such as computing Laplacian matrix and computing eigenvalues and eigenvectors.
• matplotlib: This library was used in order to generate our plot and different figures for the report and the presentation.
• networkx : This library was used in order to create a network based on the citation for each case, it helped us to create a directed graph for our data set.
• sklearn:
    • KMeans: This part of sklearn was used in order to compute the Kmeans clustering for our graph.
    • silhouette_samples and silhouette_score : We used this part of sklearn library in order to perform the silhouette for our graph in order to select the best number of centroids for our cluster experiment.

IX. CONCLUSION

Regarding the fact that the project outline was totally unknown first and the law domain was not our main field it was really challenging to get the data and the project. We learned a lot about data cleaning and data prepossessing regarding the fact that it was the main goal of this part of the project. The information extraction for each case was a really complex and a good challenge for us as new data scientists. We understood the fact that most of the data scientist work is about preprocess the data and cleaning it. I learned a lot about different approach of how to extract and clean the data for huge datasets. We can easily state that the analysis could be improved with more accurate extracted information using new approaches such as LSTM network for name entities recognition. We can also work on different aspects of the data such as the analysis of the correlation between judges and their political affiliation which, in a sense, can help to find interesting features in our cases.

We can also continue our work on sentiment analysis over our texts in order to connect news cases based on different features and information about a future case.

Finally, one useful step and experiment on the project will be the prediction of a future trial outcome based on relevant features for the judge, the defender and the offender in order to be a useful tool for lawyers.