International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Natural Language Processing in Court Order Scrutiny

Arunima Maitra

Department of Computer Science and Engineering, R. V. College of Engineering, Bangalore, India
Email ID: *arunimamaitra2000[at]gmail.com*

Abstract: ***With an increasing number of court orders coming in daily, it has become difficult for the legal teams to keep up with the case updates. The most time - consuming and inevitable part is to read the legal order again and again until all relevant points are noted. Minimizing the time spent to gather all the facts of the order can speed up the analysis process. Hence, there arises a need for a search system which can resolve this problem by generating a concise and comprehensive report of the court order. In this paper, we propose to create a pipeline that will extract key information from the court orders, analyse the sentiment from the linguistic elements and summarize the important passages pertaining to the final judgement in the order. The project will create a Data Lake, populated with artifacts and related data/ meta data which will accumulate over time and become in valuable to the legal department.***

Keywords: Natural Language Processing, Information Extraction, Regular Expressions, Sentiment analysis, Text summarization.

1. Introduction

An official declaration by a judge that establishes the legal relationships between the parties to an appeal, hearing, trial, or other court proceeding is known as a court order (Morison, J. and Harkens, A., 2019). In this case, we are particularly concerned about the court proceedings that involves the income tax department of India. Income tax assessment order (Kiyasov, S. U., 2022) is an order passed by the assessment officer to verify the documents like computation of income, a book of accounts and bank statements etc. Assessment orders by the Income Tax Department are sometimes appealed by assessee (s) in High courts seeking to impugn such assessment orders that are not in favor of the assessee (s). The process of appeal and different stages there on are well laid out. These requests from petitioners (assesses) may be awarded in favoror against the Income Tax Department. The objective here is to automate and, thereby, expedite the process of reviewing the High Court orders to decide whether to file an appeal in Supreme Court or not. Currently, this process of reviewing the High Court orders is by and large manual and takes a fair bit of time. Also, being manual in nature, the exercise is prone to errors. It is to be noted that an appeal can be filed only up to 90daysfrom the date of the high court order. Of these 90 days, the regional teams of tax department take a significant amount of team to themselves review the high court orders, as well as previous orders from lower appeal authorities and tribunals. So, by the time these orders reach the central legal team, there is very little time left.

In the contemporary process, these High Court Orders are manually read by the personnel and the key aspects of the High Court orders are manually extracted and collated in a spread sheet. This is repeated for every order and information is consolidated in a sheet and sent over for further scrutiny and action. This process is human intensive, monotonous and consumes valuable time of the officers. This paper aims to present a system which can automate most of this tedious task.

In a nutshell, the paper is organized in the following manner. Section I gives a brief introduction to the research. Section II contains all the relevant related work and concepts used to design the system. Section III contains the methodology used for this research to obtain the results. Section IV consists of the result obtained and conclusion. The paper ends with Section V, that gives the limitations as well as the future works associated with this research.

2. Background and Related Work

This section contains various sub – sections which provides a detailed analysis on all the techniques that we are going to use for this research.

A. Information Retrieval

Information extraction and question answering are some of the most used applications of information retrieval. The automatic extraction of structured data from documents that are unstructured or semi - structured for device reading is known as information extraction (IE), whereas information retrieval (IR) is the practice of employing queries to locate content from vast datasets that satisfies an information demand and is unstructured (usually documents) which are most often kept on computers.

Information retrieval is a computing process that can extract relevant information from documents, databases, even the world wide web and so on (Arulselvarani, S., 2022). Conventional information retrieval approaches are unable to satisfy the task of producing high – quality search results while data grows and the demand for high – quality retrieval results rises. Although Natural language processing and Information Retrieval are two very different areas of research, many NLP techniques, Such as part - of – speech tagging, stemming, de - compounding, compound recognition, chunking, word sense disambiguation, and others have been used in Information Retrieval (Brants, T., 2003).

It is frequently essential to retrieve critical details from reports, publications, documents, and soon. For instance, organization names–price levels from accounting results, judge names–jurisdiction from court judgments, bank details from client complaints, and so on. Such extractions are a component of Text Mining and are critical in transforming unstructured data to structured data that can then be used for analytics/ machine learning. Such entity retrieval employs techniques such as 'rules, "look up' and 'statistical or machine learning'. In this paper, we are particularly focused on the rule - based approach because it makes pattern searches to locate key information (Chapman, C. and Stolee, K. T., 2016). Regular expression is one of the rule - based approach that we are concerned with. Python has an extensive regular expression library name dre. We have utilized this mechanism to extract some of the entities like names of respondent and petitioner, name of the court, date on which the order was filed by court and so on (Bozkurt, S., et al., 2019)

a) Text Summarization

Text summarization is the process to reduce the volume of a given text so that the information remains intact – but it is conveyed with fewer words. Broadly the process for summarization using natural language processing can be classified into abstractive text summarization and extractive text summarization (Widyassari, A. P., et al., 2020). In the case of Extractive text summarization, a coherent summary is provided by selecting phrases directly from the document using as coring method. This method works by identifying important sections of the text, then cropping out and stitching together portions of the content to supply a condense diversion. While Abstractive text summarization methods aim at producing summary by interpreting the text using advanced Natural Language Processing techniques to get a replacement shorter text, parts of which did not appear in the original document, that conveys the critical information from the original text, requiring rephrasing sentences and incorporating information from full text to come up with summaries like a human - written abstract usually does (Gambhir, M. and Gupta, V., 2017). Although sometimes extractive text summarization can be considered obsolete, and the current research emphasizes abstractive and real – time summarization, this paper deals with legal documents, and for legal information, it is in the best interest to not tamper with the specifics. Hence there search here, focuses on generating a summary of the court order with extractive methods. While preserving the meaning of the text, this technique also avoids semantic confusion – as non ovel words or phrases are included in the summary. However, there might be some input files that are not in proper PDF format, rather it is the scanned version of PDF - mostly filled with stamps and signatures. In that case, the OCR text file will produce lot of gibberish text along with the actual ones. Even after cleansing of the text, it might not be 100% accurate. So, if extractive text summarization is used here, it will pick up some of the gibberish words along the way. Hence it would be best to use abstractive summarization in this scenario. So, itwould be safe to assume that the type of summarization should be dependent on the input.

b) Sentiment Analysis

Sentiment analysis, also known as opinion mining, is the quantitative study of people's opinions, feelings, emotional reactions, assessments, and perceptions about various objects, including goods, facilities, organizations, people, issues, events, themes, and their characteristics (Zhang, L., et al., 2018a). Such analysis can be worthwhile for court orders. As legal information increases, it is evident to congregate and analyze such data to predict the judgment in judicial proceedings. Although several pre - trained models and libraries are present in the NLP domain which deals with sentiment analysis, most of them a retrained on general English language datasets like - news, reviews, etc. However, there is a strict shift in vocabulary when working with domain specific information. Legal documents might contain phases that might be irrelevant if legal domain is not considered.

The same hypothesis can be applied to summarization module as well. NLP model strained with legal documents are bound to provide better and pertinent results in case of summarization and sentiment analysis of the linguistic elements.

3. Methodology

This section provides the detailed flow of the project along with the technologies and algorithms used. Figure 1 shows the flow chart associated with this research paper.

At first, the users have to select the court orders they want to inspect from the database and choose the level of summarization. Once that information is gathered from the front end, the proposed system will convert the PDF files into text files. As no Natural Language Processing task can use PDFs, this conversion must be accurate. This is performed using Optical Character Recognition (OCR). The field of pattern recognition that has received the most attention during the past few decades is optical character recognition. The process of automatically distinguishing various characters from typed, handwritten, or printed text and transforming the material to a machine – readable format is known as optical character recognition (Awel, M. A. and Abidi, A. I., 2019). The machine – readable text can be used for several data processing purposes such as editing or searching. Optical character recognition (OCR) is the quintessential technology for automatic text recognition – a core area in NLP. Quite a few open – source OCR engines are out there, among them Tesseract (Boiangiu, C. A., et al., 2016) is one of the most developer – friendly tools. It can be used directly or, for developers, through the use of an API to extract written text from photos. A wrapper for the Tesseract - OCR Engine, Pytesseract, or Python – tesseractisan OCR tool for Python. It is frequently used in Python OCR image - to - text use cases and has the ability to read and detect text in images. Figure2 shows the Tesseract OCR process flow.

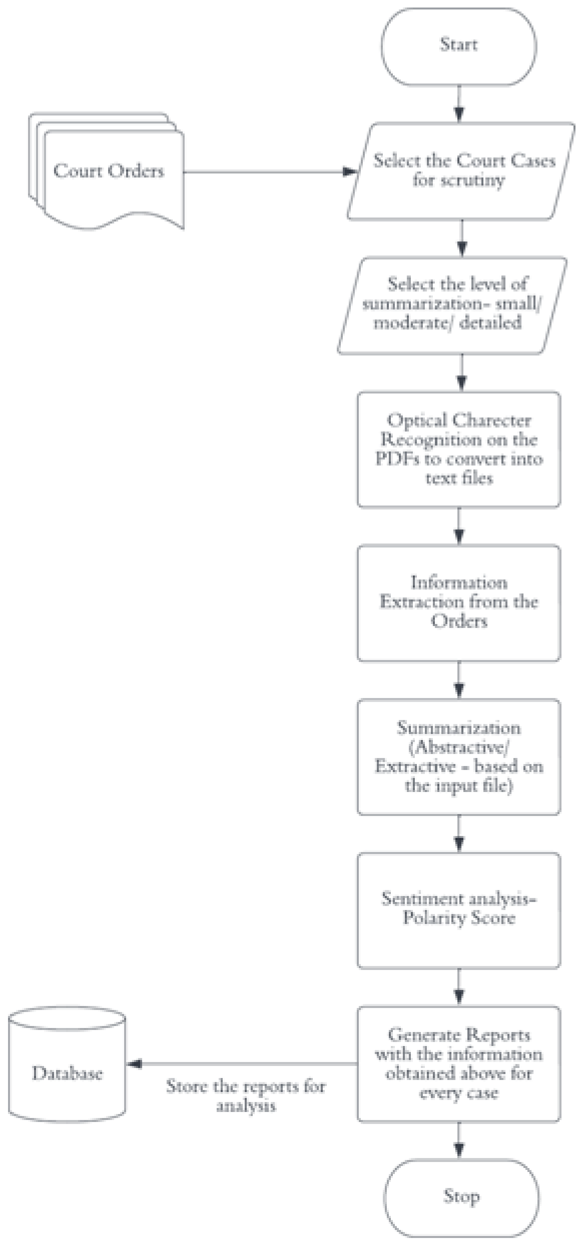
Start
Court Orders
Select the Court Cases for scrutiny
Select the level of summarization- small/ moderate/ detailed
Optical Charecter Recognition on the PDFs to convert into text files
Information Extraction from the Orders
Summarization (Abstractive/ Extractive - based on the input file)
Sentiment analysis- Polarity Score
Generate Reports with the information obtained above for every case
Store the reports for analysis
Database
Stop

Figure 1: Flow chart of Natural Language Processing in Court Order Scrutiny

Text data cleaning is one of Natural Language Processing's (NLP) most frequent tasks. So, once we have the text file, it is crucial to perform the clean - up. Some of the general steps in text cleaning involves the removal of stop words and punctuation, tokenization, spelling check, lemmatization, stemming, and removal of line breaks. These tasks can be easily performed with NLP libraries like NLTK or Spacy.

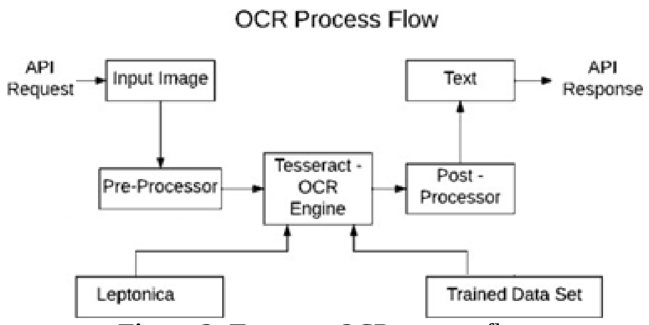

Figure 2: Tesseract OCR process flow

All the desired procedures can be carried out on the text after it has been cleaned. The first task here is information extraction. Partial parsing and automated knowledge acquisition are two areas that current Information Extraction (IE) research appears to be going (Riloff, E., 1999). The broadly categorized approaches for IE include – The pattern matching – based approach, the Gazetteer - based approach, Machine Learning – based approach (Singh, S., 2018). For this research, a pattern matching approach is utilized. In this method, Regular Expressions (RE) are used to define the extraction patterns. The extracted text that matches the patterns and belongs to an instance of that entity is then readily matched with the provided input text. The goal here is to create automation pipeline to extract key information from the order. This will include information such as - Name of the case (petitioner and respondent), Name of the court, Year/Date of Order, Sections, and sub - sections referred to in the order, Citation (s) if any and Important passages pertaining to the final judgement.

This project uses two popular NLP libraries for the summarization module. Those libraries are Gensim and Spacy. Their available summarization models are pre - trained on common datasets, like IMDB movie reviews and Amazon fine food reviews. However, there are options to fine - tune the training process on a separate dataset. This makes the testing and training process on the court orders relatively easy. Gensim's version of summarization is based on a variant of the well – known Text Rank algorithm. Based on the TextRank algorithm (Mihalcea, R., 2004), Gensim's summarization module currently only functions for the English language because the text is pre - processed to eliminate stop words and stem words, both of which depend on the language. TextRank algorithm was developed from the infamous PageRank algorithm, which was created as a mechanism for Web link analysis and is arguably one of the most well - known ranking algorithms. SpaCy (Vasiliev, Y., 2020) is a Python and Cython programming language – based open – source software library for sophisticated natural language processing. It is intended for systems that extract information or understand natural language. It offers a clear and approachable API and is designed for use in production. Of all the NLP libraries that have been made available so far, it offers the fastest and most precise syntactic analysis. Unlike Gensim, it has pre – trained pipelines that can now tokenize and train for more than 60 languages. The process of automatic text summarization in Spacy includes an English tokenizer, Part - of - speech tagger, Dependency parser, and Named Entity Recognition (NER). Lastly, it involves finding a sentence score for each sentence. It also has the option the selected the ratio of top -

scored sentences that are needed in the summary. This feature can help to obtain a small, moderate, and detailed summary.

Finding the text's positivity, negativity, or neutrality is a relatively common natural language processing problem called sentiment analysis (Zhang, L., et al., 2018b). Finding the sentiment connected to reviews and comments is highly helpful since it allows us to extract important information from text data. For this research, two of the popular NLP libraries for sentiment analysis have been used. NLTK is the foundation for the Python library called Textblob. It has been around for a while and is quite simple and practical to use (Loria, S., 2018). Text Blob gives the numerical scores for polarity and subjectivity when used to determine the sentiment of a text. The polarity's numerical value indicates how strongly positive or negative a sentence is. Like objectivity, subjectivity indicates how objectiveor subjective a writing is. Each word in the lexicon is given a score independently using Text Blob's technique for calculating sentiment. This process is described in the GitHub implementation of the TextBlob Library. By default, it uses a lexicon of adjectives and their hand – tagged scores to determine the average polarity and subjectivity across each word in each text. To do that, pattern library is used, which leverages sentiword net's (Denecke, K., 2008) individual word scores. Two sentiment analysis implementations are included in the Textblob's sentiments module: Pattern - Analyzer (based on the pattern library) and Naïve - Bayes - Analyzer (anNLTK classifier trained on a movie reviews corpus). However, the text blob. Classifiers module (code available in GitHub) makes it simple to create custom classifiers, that can be used to train custom data.

Figure 3. Sample Output for the Court Order Scrutiny System VADER or, Valence Aware Dictionary and sentiment Reasoner (Hutto, C. and Gilbert, E., 2014), will be the subject of subsequent discussion. Vader does better at identifying negative sentiment. When it comes to social media text sentiment analysis, is quite helpful. It is a rule/lexicon - based, open - source, pre – built sentiment analyzer library that is covered by the MIT license. The dictionary that the VADER sentiment analysis class returns contain the probabilities that a text is positive, negative, or neutral. The sentiment with the highest probability can then be selected by filtering. A dictionary is used in the VADER sentimental analysis to translate lexical input into sentiment ratings, which represent the strength of an emotion. One can calculate the sentiment score of a text by multiplying the intensity of each word in the text.

4. Result and Conclusion

Figure 3, shown at the beginning of the page, is the sample output the system is supposed to generate. This is, however, shown for one case order only. In its actual usage, the system should be able to print out the information for a large number of court cases in the appropriate columns. The columns contain all the fields that were discussed earlier in this paper.

To make it more useful for the end users, creating a web application for this project is in progress. This is done using the flask (Grinberg, M., 2018) framework for python. The website shall be available for all lawyers, paralegals, and assistants. Additionally, there will be a requirement for an administrator to handle the court cases available for scrutiny, in the database of the system. The scrutiny reports generated should be stored safely for further analysis and use.

Thus, NLP can help legal scholars save a bunch of time by pointing out and directing them to specific phrases that exist in long court rulings or where particular terms in a search query appear in relation to other terms. As a result, lawyers can select which instances are not pertinent immediately and move on to the next case, or they can go deeper into the cases whose search phrases more closely fit the intended search criteria.

5. Limitations and Future Work

The project under stands the challenges posed by the various nuances of scribing of the judgements across India and appreciates that they can beat time, significantly different, even for similar orders. This accounts for the 10 - 20% orders which might need some degree of manual involvement. However, the outcome from the project will only become better as the underlying models/engines get trained by more and more representative data over time. The quality of the input PDF file is quite significant for generating clear text files.

It is to be noted that this project is just one foundational component of a possibly larger program. The full – blown program has the potential for helping the Legal & Research team to identify orders for appeal based on “Winnability” score i. e., based on chances of winning the appeal in Supreme Court. The model for “Winnability” will need to be built and trained using past orders and various attributes of importance. It can also help to almost instantly know the outcome of the orders (for or against Assessment Authorities) at various levels of judiciary. Creating dashboards for executive summary and perform deep analysis at various levels of appeals can be another area of future evolution for the project.

References

[1] Morison, J. and Harkens, A., 2019. Re – engineering justice? Robot judges, computerised courts and (semi) automated legal decision - making. *Legal Studies*, 39 (4), pp.618 - 635.
[2] Kiyasov, S. U., 2022. Improving the Quality of Tax Administration Is a Key to Ensuring the Objectives of The Individual Income Tax Reform. *International Journal of Progressive Sciences and Technologies*, 30 (2), pp.437 - 445.
[3] Singh, S., 2018. Natural language processing for information extraction. *arXiv preprintarXiv: 1807.02383*.
[4] Arulselvarani, S., 2022. An Efficient Information Filtering Using Query Reorganization Based Self – Organizing Map Method. *International Journal of Advanced Research in Engineering and Technology (IJARET)* Volume11, Issue 12

[5] Brants, T., 2003. Natural Language Processing in Information Retrieval. *Computational Linguistics in the Netherlands Journal.*

[6] Chapman, C. and Stolee, K. T., 2016, July. Exploring regular expression usage and context in Python. In *Proceedings of the 25th International Symposium on Software Testing and Analysis* (pp.282 - 293).

[7] Bozkurt, S., Alkim, E., Banerjee, I. andRubin, D. L., 2019. Automated detection of measurements and their descriptors in radiology reports using a hybrid natural language processing algorithm. *Journal of digital imaging*.

[8] Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A. and Affandy, A., 2020. Review of automatic text summarization techniques & methods. *Journal of King Saud University – Computer and Information Sciences*.

[9] Gambhir, M. and Gupta, V., 2017. Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review*, *47* (1), pp.1 - 66.

[10] Zhang, L., Wang, S. and Liu, B., 2018. Deep learning for sentiment analysis: Asurvey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8* (4), p. e1253.

[11] Awel, M. A. and Abidi, A. I., 2019. Review on optical character recognition. *International Research Journal of Engineering and Technology (IRJET)*, *6* (6), pp.3666 - 3669.

[12] Boiangiu, C. A., Ioanitescu, R. and Dragomir, R. C., 2016. Voting – based OCR system. *The Proceedings of Journal ISOM*, *10*, pp.470 - 486.

[13] Riloff, E., 1999. Stepping Stone toward Story Understanding. *Understanding language understanding: Computational models of reading*, p.435.

[14] Srinivasa - Desikan, B., 2018. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.

[15] Mihalcea, R., 2004, July. Graph – based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL interactive poster and demonstration sessions* (pp.170 - 173).

[16] Vasiliev, Y., 2020. *Natural Language Processingwith Python and SpaCy: A Practical Introduction*. No Starch Press.

[17] Loria, S., 2018. textblob Documentation. *Release0.15*, *2* (8).

[18] Denecke, K., 2008, April. Using sentiword net formulating lingual sentiment analysis. In *2008 IEEE 24th international conference on data engineering workshop* (pp.507 - 512). IEEE.

[19] Hutto, C. and Gilbert, E., 2014, May. Vader: A parsimonious rule - based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media* (Vol.8, No.1, pp.216 - 225).

[20] Grinberg, M., 2018. *Flask web development: developing web applications with python*. "O'ReillyMedia, Inc. ".

Author Profile

Arunima Maitra received B. E. degree in Computer Science and Engineering from R V College of Engineering, Bangalore, India in 2022. She worked as a Data Science Intern in a startup for several months. Currently, she is working as a Systems Engineer at Oracle Cerner. She has grown a keen interest in the field of natural language processing and computer vision. She has done various internships and projects in the related field and wish to pursue higher education in the same domain. She can be contacted at email: arunimamaitra2000[at]gmail.com