Name Letter Effect Using Artificial Intelligence (GPT 3.5 Turbo -1106)

Hamza Aziz Rangwala
Master of Applied Computing (MAC)
Wilfrid Laurier University Waterloo, Canada hamzarangwala51@gmail.com

***Abstract*—**

AI is an emerging and disruptive technology that has already changed how information is managed and processed and promises to continue doing so in the future. In this paper, we discuss the changes and impacts of AI around legal information using the data from database research of Canadian tax court. It shows how the data is parsed through the Open Ai platform and how it interprets the data which is in structured format.

AI has been referred to in the Canadian Case law including legal research, investment tax credits, trademarks. In legal circles, AI is viewed both as an opportunity and a threat. The beneficial use of AI in law is due to the fact that AI technologies promise to alleviate the already high workloads in various aspects of the judicial system, all which are all labor intensive and time-consuming. AI reduces the time and effort which is required to process the data and interpret it. Interpreting the legal cases is a challenging task due to the fact that a Conventional person is not able to read and interpret the Judgment of a Case and predicting the Outcome of it. This paper shows the methodology to implement the extraction of data the main features and benefits of using Artificial Intelligence to extract the relevant information from the case in a diminutive amount of time while being cost effective.

***Index Terms*— Artificial Intelligence, Judgment, Open AI, Interpretation.**

I. Introduction

AI is an emerging and disruptive technology that has already changed how information is managed and using the data from database of the Tax court of Canada. The objective of this paper is to extract key components from the data scraped from the tax court of Canada and investigate how the initials of the judges and the Appellant are related to the Judgment made in the case. AI is used to further find out what would be the time frame for the tax dispute. Furthermore, we use the findings to make suggestions for legal information professionals, as to how they could interpret the data based on the prediction that the Initials of an appellant and a judge can cause bias towards the outcome of the dispute.

The name-letter effect refers to the psychological phenomenon where individuals tend to prefer the letters that appear in their own names, influencing their choices and preferences. [5] The name letter effect is the tendency to evaluate alphabetical letters in one's name, especially initials, particularly favorably. Recent evidence suggests that name initials may even predict career choices. The phenomenon is rooted in the concept of implicit egotism, suggesting that people unconsciously gravitate towards things associated with themselves, even as subtle as their initials. This research paper is predicated on discerning the name-letter effect within the jurisdiction of the Tax Court of Canada.

[1] Some lawyers still consider that the impact of AI software and automation is lower than in other professions, particularly in the banking and financial sectors. However, when AI is built to analyze past legal decisions written by courts and tribunals and used to discover patterns that predict the outcome of future cases with a high percentage of accuracy, this will directly reduce the amount of time lawyers spend on reviewing legal documents. It is our hope that this paper will spark additional research discussions and inspire other scholars to delve deeper into the field of artificial intelligence and its application to the management of legal information.

The AI technologies described are already being implemented in the computer systems of many law firms and courts. In the future, [2] AI could take over some of the time-consuming tasks that normally occupy lawyers and judges, thus freeing them up to do other more important tasks and reduce backlogs in the judicial system. [1] An example application of AI in legal work is the AI system 'Blue J Legal', which was designed to predict outcomes in tax law. It predicts the likely rulings that the judges would make for certain kinds of cases in tax law, and enables lawyers to prepare their court arguments accordingly.[3] Alarie, Niblett and Yoon (2016) argue that although AI is a helpful tool it cannot replace human experts who have good legal knowledge and training.[4] Another similar study further discusses the application of AI to the work of lawyers and to the justice system. These applications rely on the fact that the law is based on sets of rules and that rule-based AI systems can be programmed by software engineers to perform such tasks and process data in accord once with them. However, legal practice touches on complex social issues as well and there is a need to consider the complexities of each jurisdiction's legal process.

As an application involving use of AI is very complex and involves parsing through a lot of data the Open AI platforms API (Application Programming Interface) has developed a lot through the course of this project. To monitor the changes in the API and keeping a check on the token amount of the data being parsed is a complex task as the judgement of a case involving a tax dispute can be a very large text data. Natural language processing is another AI technology that allows computers to better understand human languages and enables lawyers to predict the outcomes of legal claims or judicial decisions. Using the Api is significant as it reduces complex tasks involving NLP (Natural Language Processing) and reduction of the task of

parsing the whole case to predict the features of the case. The benefits of using Open AI Api are abundant. One major boon of using Open AI API is increasing the scalability of the application. If a specific parameter or feature is needed to interpret the underlying case the interface is useful to predict those outcomes in a constrained amount of time and cost. We will be exploring how the application is able to scrape information from the database of the tax court of Canada and iterate through each case in a year and provide a set of important features required to determine the bias of a case towards an appellant assuming the appropriate information has been detailed and briefly explained in a case.

II. Proposed System

The purpose of this study is to gather information remotely and provide a gateway for the user to retrieve the collection of data in a file which is in csv format. The proposed methodology can be applied to any data which is in a structured format (JSON) to extract the data, gather relevant information and view it remotely. My Research proposes that using some easily avail- able software's we can extract information from the tax court database for gathering key features from the case. To explain the system in a better way we have broken down the process into sub-tasks:

A. Data Injection (Uploading)

A NodeJS Framework is used in implementation of this project. Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a web browser. It follows a single-threaded, event-driven architecture. It uses an event loop to handle multiple concurrent connections without the need for multithreading. This makes it highly efficient for handling many connections simultaneously. It is commonly used for building web servers, APIs, real-time applications (e.g., chat applications, online gaming), and microservices. Its ability to handle many concurrent connections makes it suitable for applications with high traffic and scalability requirements.

A multer package is configured specifying memory storage for storing the uploaded files in memory. It filters the uploaded files to allow only those files whose MIME type is JSON ensuring only Json files can be uploaded. On root endpoint of the application a page is rendered to show the user the UI for uploading the Json file. Serving static content is essential for providing stylesheets, client-side scripts, and images to the client. A middleware is also integrated to store the uploaded for subsequent processing.

Fig. 1. Data Injection Page (Home Page).

After successfully adding a new file, we press on the submit button which takes us through the validation of the form that checks the input type of the file. If validated it takes us to the next route '/upload'. This route takes a single file as upload and if true it redirects the user to the 'loading' page.

B. Data Extraction (Loading)

The previous step explained how we could extract the data and store it in the memory for processing. In this step we create the instance of extracting the 'unofficial_text' which contains the text data of the whole judgment from the Json file. Since we have Json data in the middleware, we can easily access the data for parsing and extracting the required values using the keys from the file. We extract the URLs provided with each case which backtracks to the Tax Court of Canada website to the case the url is given. We are also able to extract the Appellant Names and the Judge Names from the text data of the judgment using a match pattern by creating a regex (Regular Expression) to identify the pattern in the text, available in the framework. To implement this UI for loading the keyframe component is used inside the Cascading Style Sheet properties on the Web Page.

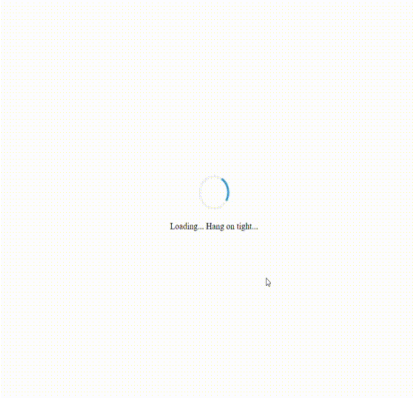

Fig. 2. Data Extraction (Loading).

C. Api Calls (Feature Extraction)

We have extracted the data required for further processing, now we need to make the Api call to pass the text from the Json file to the GPT model API to extract the required information and take the Output from the API for further processing. The OpenAI model constitutes a formidable instrument, demonstrating considerable efficacy in the facilitation of intricate tasks, thereby significantly ameliorating the complexity inherent in their execution.

Till the data is being passed through the API, we have implemented the asynchronous functionality to the application, so it performs the asynchronous operation first before executing the other part. In essence, it waits for the API calls to be completed before executing any other piece of code. The data cannot be parsed at once as it is based on a request response model. Therefore, after each request the data is thrown to a file which is in a JSON format to keep track of the responses and have a backup for the responses as the API could throw an error or stop working if any component is affected. The most substantial challenge faced when working on the Open AI Api is the rate limit for the context length that can be passed through for each request. The counting of tokens becomes an integral component when passing data through the context of the request. Initially setting the tone of the API is very fundamental, the response will depend on the instruction set on initialization. Subsequently counting the context length of the data which we gathered from the Json file is similarly crucial. Based on the length of the input we classify the cases into two components: -

1) Cases having token length less than 16385.

2) Cases having token length greater than 16385.

For cases having the length less than the requisite we don't further process the data and send the request to the body of the API and get the response. It is imperative to highlight that the temperature parameter can be configured to enhance the precision of the API. Additionally, for cases having length greater than the requisite we have divided the content to chunks where each chunk is less than the requisite length so that the rate limit error is not thrown when processing this data. The OpenAI platform has introduced a novel feature known as the 'response_format' parameter. This parameter empowers the API to furnish responses in a tailored and specific format, thereby enhancing customization and flexibility in the output presentation. Furthermore, we have seamlessly integrated this feature into the application, thereby enabling the acquisition of responses in the JSON format. This integration optimizes the interoperability and structured data handling within the application ecosystem.

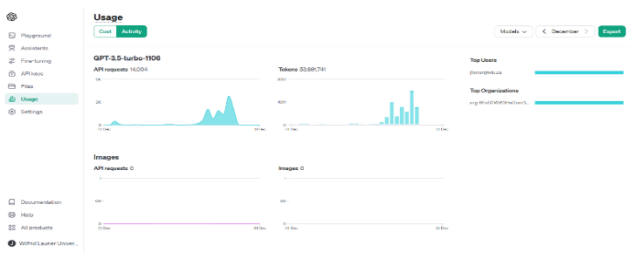

Fig. 3. Open AI API Dashboard Page.

D. Data Extraction from API

The data accumulated from the API is processed sequentially and stored in a JSON file using the fs (File System) module provided by the NodeJS API which is also asynchronous .Once the data have been stored in a file, we extract the information from the file using fs.read() function and partition the data into various features by using jsonObject which identifies the particular object in the array and stores it outcome in an Array which will be used for further processing.

E. Initials Comparison

In this step we are going to explain how the Initial Comparison is made on the Output extracted by us from the API. The Initials of the Judge and the Appellant are initially trimmed to remove any extra space that could have been there in the output. Subsequently, Error handling is done to verify that the input is not empty or undefined so that we have proper initials at the time of comparison. The initials are then extracted from the array created during data extraction and compared on these six conditions:

- 1 – first initial and last initial of taxpayer match with the first initial and last initial of the judge (example: taxpayer is JF and judge is JF)
- 2 – first initial of the taxpayer matches the first initial of the judge (example: JF and JP); last initials do not match
- 3 – last initials match but first initials do not match (example: TG and KG)
- 4 – first initial of taxpayer matches judge's last initial (example: JH and DJ)
- 5 – last initial of taxpayer matches judge's first initial (example: JF and FB)
- 6 – no initials match

The function returns the number based on these conditions. These comparison helps us in understanding the bias in the case if there is one.

F. Excel file (Creation)

In this step we will be creating the excel file for downloading the data which we extracted as well as the comparisons and the data we found out from parsing the text from the JSON file uploaded. We are using the xlsx module from the NodeJS to provide manipulation in the excel file. The book_new() function creates a new excel sheet for us to work on. We also have created a combined array by mapping the data and indexing to ensure all the arrays have the same length which makes it more accessible to combine. We then create a worksheet to input the combined array into the workbook and write it to the excel file. The aoa_to_sheet() (Array of Arrays to Sheet) function helps in inserting the combined array to the excel file.

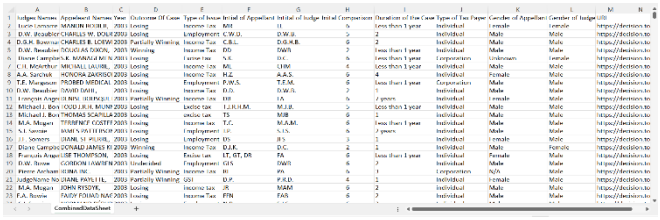

Fig. 4. Excel file

G. Landing Page (Outcome)

The web page designed is to display and facilitate the download of structured legal case data presented in tabular format. The content is organized into sections, each represented by a table, encompassing details such as names of applicants and judges, case outcomes, and various case attributes. The page integrates client-side functionality through JavaScript to notify users upon initiating the download of an Excel spreadsheet containing the displayed data. The utilization of semantic HTML tags and CSS styling enhances the overall readability and aesthetic presentation of the legal case information. Additionally, the integration of client-side scripting contributes to a more interactive user experience.

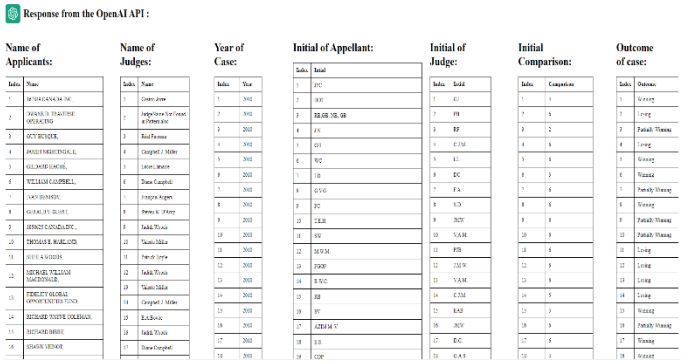

Fig. 5. Outcome Page

III. CONCLUSION

In conclusion, this paper delves into the transformative effects of Artificial Intelligence (AI) on legal information, with a specific focus on the Canadian tax court. By leveraging the OpenAI platform, the research demonstrates a structured approach to parsing and interpreting legal data, showcasing the platform's efficacy in handling complex, labor-intensive tasks. The exploration of AI's role in Canadian case law, encompassing areas such as legal research, investment tax credits, and trademarks, underscores its dual nature as both an opportunity and a potential challenge within legal circles. The paper asserts that the integration of AI technologies holds promise in alleviating the substantial workloads inherent in the judicial system, ultimately reducing processing time and effort. The methodology presented herein illustrates the practical implementation of AI in extracting pertinent information from legal cases efficiently and cost-effectively. As AI continues to evolve, its profound impact on the legal domain becomes increasingly evident, and this paper contributes to the ongoing discourse on the intersection of AI, legal interpretation, and judicial processes.

Acknowledgment

I would like to express my sincere gratitude to Professor Jonathan Farrar, Ph.D., CPA, CA, for his invaluable guidance and mentorship throughout the completion of this project. Professor Farrar's extensive expertise as a Professor of Accounting (Taxation) has been instrumental in shaping the research and enriching its content. I am deeply appreciative of his commitment to academic excellence and the generosity with which he shared his knowledge.

I would also like to extend my heartfelt thanks to Professor Abdul-Rahman Mawlood-Yunis for serving as a mentor and providing valuable insights that significantly contributed to the project's development. His guidance has been instrumental in refining the scope and depth of the research. Special mention is reserved for Er. AliAsgar Fazal Abbas, whose thoughtful comments on the application added significant value to the project. I am sincerely grateful for his constructive feedback, which enhanced the overall quality and rigor of the work.

Their collective support and mentorship have been crucial in navigating the complexities of this project, and I am truly privileged to have had such esteemed individuals as part of this academic journey.

References

[1] Ben-Ari, Daniel, et al. (2016) Artificial intelligence in the practice of law: An analysis and proof of concept experiment. Rich. JL & Tech 23(2).

[2] Intahchomphoo, C.; Vellino, A.; Gundersen, O. E.; Tschirhart, C. Shaaban, E. References to Artificial Intelligence in Canada's Court Cases. *Legal Information Management* **2020**, *20* (1), 39–46. https://doi.org/10.1017/S1472669620000080

[3] Alarie, Benjamin, Niblett, Anthony, & Yoon, Albert H. (2016) Using machine learning to predict outcomes in tax law. Canadian Business Law Journal 58 (3), 231–254.

[4] Alarie, Benjamin, Niblett, Anthony, & Yoon, Albert H. (2016) Law in the future. University of Toronto Law Journal 66(4), 423–428.

[5] Hodson, G., & Olson, M. A. (2008). The Name-Letter Effect in Mate Choice: A Failure to Replicate the Effect on Overt Behavior. Journal of Research in Personality, 42(4), 1094–1097. Retrieved from http://www.stat.columbia.edu/~gelman/stuff_for_blog/hodson.pdf