“PreDicta” Project biography: By: Mario A. Treto

This project is a “proof-of-concept” case analysis tool which may connect to any Google Sheet, allowing the user to search the entirety of the spreadsheet’s data with a partial search. The user’s search then pulls up all potential matches (excluding header columns in the 1st row) and is prompted to select the entry they wish to have analyzed. The Row’s information is then sent to Chat GPT and ran through a liability determination prompt. Upon completion of the Chat GPT request, a new column is added to the end of the program to store the Chat GPT response.

This is all accomplished in a matter of 8 - 9 steps on our Google Colab landing page:
GITHUB REPO: https://github.com/LITSA-Mario/ctlFINAL/blob/main/mario_t_PreDicta.ipynb

This tool was developed for the Law Offices of Leonardo J. Caruso, but also can apply to any busy injury law firms struggling to perform case assessments after regular business hours.

Attorney Caruso is a Boston Based attorney with over 35 years of Civil litigation experience. His office specializes in Personal injury Lawsuits (Auto, med mal, dog bites, & workman’s comp).

While beginning to brainstorm this solution, my conversations with Attorney Caruso, and his Associate Attorney Mr. Samuel Hirl, proved to be essential as they explained the Office’s need for a simple and free way to extract client data from our online intake spreadsheet and produce a case summary.

The key problem we identified is that clients were rarely signed up outside of working hours.

This can be attributed to having no employee available to answer phones/ review online intake submissions after working hours.

As no client may be signed up without the approval of Attorney Caruso, I thought it would be prudent to design a program which can produce a summary of liability and the other merits of the case at any time of day.

“PreDicta” only needs a few columns of information to submit a request to chat GPT and get a detailed response regarding liability, causation, and recoverable damages. I coded the program so that there is no set structure for the order of the columns. The program identifies the column headers and is able to recognize each column’s intended use.

- -

To shift the focus back to Attorney Caruso’s law office, a simple case summary within a google sheet can now prompt a more immediate follow up from a supervising attorney without the need for the human intermediary to deliver the case assessment based on preliminary facts.

Although other case management services (Clio, My Case, Reuters High Q, ect.) have the advantage of scale and the available funds to develop a more robust ecosystem for their users, their services are not free, and they are only beginning to integrate Chat GPT functionalities.

Our office has tried multiple services but could not find one that fit our needs of producing instantaneous case summaries on command from a set of data which is constantly updated.
It became clear to me that the goal was to increase the amount of client cases which were presented to attorney Caruso for consideration. The program also needed to be able to run off a partial search (in case the user was unaware of the exact spelling of a name or address) and be able to produce an isolated row containing the potential client's contact & case information.

The attorneys made it clear that this would be the preferred method of case review, as opposed to searching through a spreadsheet with narrow cells and columns.

Thankfully, when a search is ran on "PreDicta", the client whose information was selected is displayed directly above the response from chat GPT. This presents the user with the opportunity to review all the methods of contact provided by the client and allows for a quicker production of a user's information.

- -

Testing:

My initial drafts of the program opened a CSV. file produced by the user and created a dataframe based off of this information which could then be searched.

Unfortunately, this proved to be an inadequate solution when I first presented it to the Attorneys as they were unfamiliar with what a CSV file was and could not make heads or tails of the google collab Coding UI.

I went back to the drawing board after receiving the following feedback from the attorneys:

- "I am not going to read through code, I don't have time to."
- "Our edits aren't saved to the file? We need to be sure that all updates are saved."
- "I wish this would just connect to our Intake Google sheet, instead of requiring me to upload a file."
- "This UI is very confusing, I don't see any instructions besides the green text and that is very confusing."

As a result, I took "PreDicta" back to the drawing board after having the attorneys attempt to use the project. I included the following improvements:

1. I re-analyzed how the program should be presented to its users and:
    a. I reconstructed the program's instructions, reducing it to 8 simple steps and limiting the amount of code presented to the user whenever possible.
2. I began researching how to connect the program to a google account and gain access to the desired Google Sheet.

    a. I continued to research how to manipulate the Google sheet from within the Google Colab, allowing me to add a new column for the GPT analysis section.
3. I integrated Chat GPT functionalities and developed prompts to produce a liability review.

Following the first round of testing with the law office, I implemented the changes listed above and began testing on non-attorney users to assess the ease of use and clarity of instructions for those who were both unfamiliar to coding and legal work.

After receiving positive results about the 8 Step format from my non-attorney testing group, I finalized my revisions and returned to the law office with my final draft.

Immediately upon placing the program in front of the Attorneys, they commended me for how much the structure had improved. Once I let them know that the program could now link to google sheets and generate a liability summary from ChatGPT, they were shocked at the results.

The tool has proven to be very useful for expediting the new case analysis process and has improved the law firm's ability to contact retainable clients after regular business hours.

Attached here is a link to my partner letter:
https://github.com/LITSA-Mario/ctlFINAL/blob/main/PartnerLetterAttorneyHirl.pdf

I have also provided a link to the test spreadsheet i used:
https://github.com/LITSA-Mario/ctlFINAL/blob/main/Injuryclientlist.csv

The columns house (among other things) the [client's name], [Personal Injury Accident Description], [Resulting injuries], [Medical treatment received], [the cost of ER treatment], and [total medical bills].

However, these values can just as easily be changed to apply to many different kinds of legal practices (including immigration & contracts). All that needs to be done is a change of the Prompts sent to Chat GPT based on the accident_type.

- This modular aspect can be found within Step #8:
  (to be further developed following fall 2023 semester)

The function identify_accident type:

```
def identify_accident_type(row_data):
    description = ' '.join(row_data).lower()

# Regular expressions defined for different accident types
car_crash_patterns = r'\b(car|auto|automobile|vehicle)\b'
workplace_incident_patterns = r'\b(workplace|work\s*comp|work\s*accident|job\s*accident)\b'
```

```
#Basic check for patterns related to specific accident types in the description
if re.search(car_crash_patterns, description):
    return "car crash"
elif re.search(workplace_incident_patterns, description):
    return "workplace incident" else: return "other"
…
# GPT prompts RE: liability & specific automobile accident type
if accident_type == "car collision":
     prompt = "Scenario: An incident report details a car collision. \nDescription: " + ',
'.join(row_data) + "\nBased on this information, assess the liability in this car collision scenario,
considering the actions of drivers and road conditions."

elif accident_type == "car-to-pedestrian incident":
     prompt = "Scenario: An incident report details a car-to-pedestrian incident. \nDescription: " +
', '.join(row_data) + "\nBased on this information, assess the liability in this car-to-pedestrian
incident scenario, considering the driver's actions and pedestrian's presence or behavior."

else: prompt = "Scenario: An incident report details an incident. \nDescription: " + ',
'.join(row_data) + "\nBased on this information, assess the liability in this scenario, considering
the specifics of the incident and the circumstances involved."
```

Link to week 10 Pitch Deck:
https://docs.google.com/presentation/d/1BoOQisJxto6EPY-4KyRmI-YLE9gz8YNc/edit?usp=sharing&ouid=115057053144035077866&rtpof=true&sd=true

Thank you for reviewing my project and teaching us to harness the power of A.I./ data science Professor Colarusso!