**Lab#1-History Profiling with LLMs**

Last name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ First name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction.**

Suspect profiling, also known as criminal profiling, is a technique used in criminal investigations for nearly a century to identify potential suspects based on various psychological, behavioral, and demographic characteristics. Dr. Walter C. Langer, a psychiatrist, was commissioned by the Office of Strategic Services (OSS) to profile Adolf Hitler, marking one of the earliest known attempts at profiling in the 1940s. The rise of digital crime has necessitated more sophisticated techniques for suspect profiling. Although traditional methods of investigating are effective, they are also labor- intensive and time-consuming.

This lab explores the use of Large Language Models (LLMs) for profiling individuals based on their web history data. Profiling involves predicting certain traits, behaviors, or preferences of individuals by analyzing their online activities.

**Objectives:**

* Gain hands-on experience building and using LLMs.
* Apply the LLMs to online profiling

**Software:**

Google Colaboratory: Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. https://colab.research.google.com/

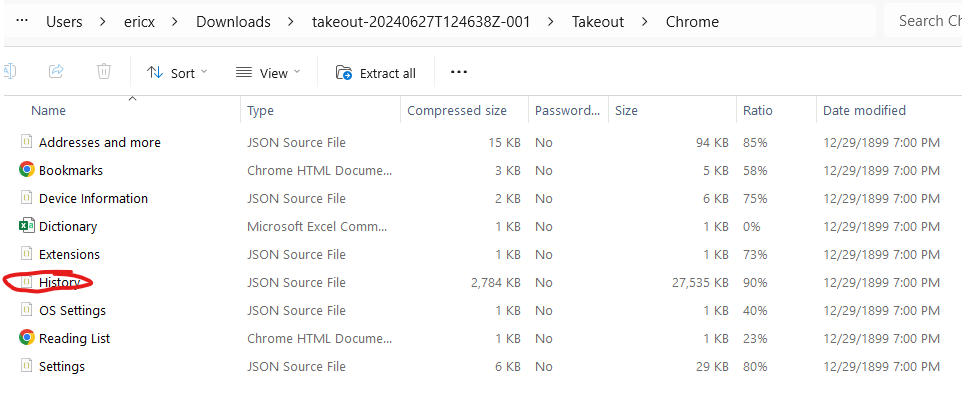
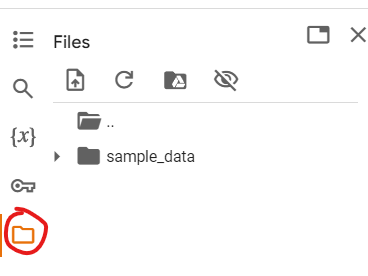
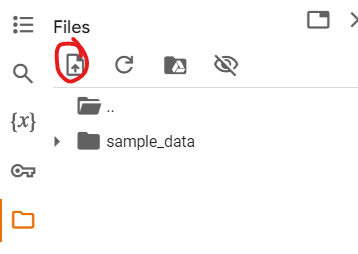
Hardware: Laptop/Desktop

Operating Systems: Win 10

Guide- ***Insert recording here***

Open this Google Collab: <https://colab.research.google.com/github/frankwxu/digital-forensics-lab/blob/main/AI4Forensics/CKIM2024/BrowserHistory/Eric/profile_browser_history_Eric.ipynb>

**Step #0: Downloading and Processing Data**

1. The Google Collab uses the wget command which downloads a sample of data
   1. (Optionally) If you want to use your own data, go to <https://takeout.google.com/settings/takeout>
   2. Unselect all items except for “Chrome: Bookmarks, history, and other settings from Chrome”
   3. scroll down and hit next. Follow the directions given by Google
   4. Depending on how much data is requested, it may take longer for Google to give you your data
   5. Once you receive you file from Google, you should unzip it by right-clicking and selecting the option “Extract all” or “unzip.” Once it is finished extracting, navigate to “Takeout” then “Chrome.” You should see a file titled “History”
      1. 
   6. Open the data processing Collab: <https://colab.research.google.com/github/frankwxu/digital-forensics-lab/blob/main/AI4Forensics/CKIM2024/BrowserHistory/Eric/profile_browser_history_Eric_dataprocess.ipynb>
   7. Upload this data by clicking on the folder icon:
      1. 
      2. You can then click and upload your file:
         1. 
   8. Remove the existing wget command and run the program
   9. Hit the refresh button to see your new file “titles\_with\_timestamp.txt.” Download the entire thing or copy a select amount of entries. Recommended is 100-200 .
   10. Go back to the Profiling Google Collab and upload your new data there.
   11. Remove the wget command that is already there and run step 0.

**Copy a few titles you found interesting:**

|  |
| --- |

**After you finish the rest of the steps and have the time, experiment with the titles you provide. What kinds of titles give the best results and why?**

|  |
| --- |

**Task #1: Installing packages**

In this first task, we'll prepare our lab environment.

1. pip is used to download packages
2. Uncomment these by removing the “#” symbol and run the step.
3. You can put the comments back once you have already ran it. This is so you do not have to wait for the package to be redownloaded.

**Task #2: Config LangChain with Gemini**

**TASK 1: Get a Gemini API key**

1. Access your key using this link: <https://aistudio.google.com/app/apikey>
2. It will ask you to login and create an account. You can use your personal email
3. Hit “Create API key” and follow the instructions given by Gemini.
4. Copy the key to your clipboard.

**TASK 2: Load your API key**

1. Create a file in Collab titled “my\_config.env” by right clicking and hitting “new file”
2. Paste in “GOOGLE\_AI\_STUDIO2 = \*your api key here\*”
3. Save the file

**Generation Settings:**

1. Temperature:
   1. Controls the randomness of the model's predictions.
   2. Low temperature (e.g., 0.2): The model generates more deterministic and focused responses, favoring higher-probability words.
   3. High temperature (e.g., 1.0): The model produces more creative and diverse outputs by considering a wider range of possible words.
2. Top-p (Nucleus Sampling):
   1. Defines a probability threshold to filter potential word choices.
   2. The model selects from the smallest set of words whose combined probabilities exceed the threshold 𝑝
   3. Low top-p (e.g., 0.1): The model chooses from fewer words, leading to more conservative and focused responses.
   4. High top-p (e.g., 0.9): The model considers a larger set of words, resulting in more diverse and creative outputs.
3. Top-k Sampling:
   1. Limits the number of word choices to the top 𝑘 most probable words.
   2. Low top-k (e.g., 10): The model picks from a smaller, high-probability word set, making the response more predictable.
   3. High top-k (e.g., 50): The model selects from a larger pool of words, increasing the diversity and creativity of the output.

**After you finish the rest of the steps and have the time, experiment with these settings. What do you expect to be the best combination of settings and why?**

|  |
| --- |

**Task #3: Build a Gemini model with configurations**

* LangChain provides a ChatGoogleGenerativeAI method.
* Allows us to interact with Gemini’s API

**Task #4: Create a prompt template**

* This is a multi-line string containing placeholders in curly braces. formatted\_prompt = prompt.format(  
   role="You are a helpful assistant.",  
   provided\_data="Here's some context: ...",  
   start="Please answer the following question:"  
   )
* {role}, {provided\_data}, and {start} are placeholders that will be filled in later.
  + {role}: definition specifies the role's name, overall objective, task specific context, and any applicable constraints.
    - Role Name: Criminal profiler.
    - Role Task: Create a psychological profile based on browsing history.
    - Role Focus: Motivations, psychological characteristics, behavioral patterns, relevant insights.
    - Role Restrictions: Avoid identification or accusations, no legal advice.
  + {provided\_data}: outlines the required datasets for task completion
    - list of web pages visited with titles and timestamps.
  + {start}: the initiation instruction serves as a trigger, prompting the role to carry out the task

**After you finish the rest of the steps and have the time, experiment with the prompt. There are some changes that can be made such as telling the AI to not refuse to answer. Type the prompt that you used.**

|  |
| --- |

**Task #5; Creating a LangChain Processing Chain.**

* Flow of operation chain = prompt | model | output\_parser
  + The prompt is first formatted and sent to the model.
  + The model processes the prompt and generates a response.
  + The output parser then processes the model's response, ensuring it's in the correct string format.
    - Can interpret CSV or JSON. We will use string
* chain.invoke method executes the chain.
* If you are outside the US and using your own data, change the prompt to guess the continent instead of the state.

**Copy your results here.**

|  |
| --- |

**After you finish the rest of the steps and have the time, experiment with the operation chain. Consider modifying the output parser and prompt to make it output a json or csv for easier evaluation. Paste your modifications.**

**Resources:**

[**https://python.langchain.com/v0.1/docs/modules/model\_io/output\_parsers/**](https://python.langchain.com/v0.1/docs/modules/model_io/output_parsers/)

[**https://python.langchain.com/v0.1/docs/modules/model\_io/output\_parsers/types/json/**](https://python.langchain.com/v0.1/docs/modules/model_io/output_parsers/types/json/)

|  |
| --- |

**Task #6; Evaluation**

You should now have a prediction of who you are based on your Chrome browser history created by Gemini. We will now evaluate the results.

* Age: We will give 1 score if the ranges are the same. If the range is off by one (i.e., if the age is 30-39 and the prediction is 40-49 or 20-29), we will give 0.5 score
* Interests: Score is the number of correct interests over the largest number of interests guessed. If 4 interests are predicted and 3 are correct, it will be given a score of 3/4 or 0.75
* Location: Score of 1 if prediction is correct. Score of 0.6 if the state borders the state. 0.3 if they are in the same section of the United States. Sections are divided into Northeast, Southeast, Midwest, Southwest, and West
  + If you are outside the US, consider an alternate evaluation method. Give a score of 1 or 0 depending on if the continent is correct

**Copy your results here.**

|  | Age | Interests | Location |
| --- | --- | --- | --- |
| Truth |  |  |  |
| Prediction |  |  |  |
| Score |  |  |  |

The output will be in a list format, each item represents a bounding box, the text detected and confidence level, respectively.

Modify the prompt to try to guess other aspects of yourself. Write down the output and how well it did.

|  |
| --- |

**Questions and Answers**

1. What are the key differences between LLMs and traditional machine learning models for profiling?

|  |
| --- |

1. What challenges might arise when using LLMs for profiling based on web history?

|  |
| --- |

1. List other areas where LLMs can be effectively applied outside of profiling.

|  |
| --- |

1. How do LLMs learn to generate text using their training data?

|  |
| --- |

1. What ethical considerations should be taken into account when using LLMs for profiling individuals?

|  |
| --- |