# ChatRegex for Detective Novels

Anderson Walsh Bryson Gullett Alex Selyutin Anirudh Gajjala

## I. Design and Implementation

### A. Input Processing

Input processing was a difficult problem to address given the limitation of regular expressions in capturing generalized means through which to express specific types of queries. Analysis of keywords broadly applicable for detective novels was applied to form a "dictionary" so to speak, in addition to a meta approach wrought from the particular selection of novels. These keywords, general and novel-specific, were reduced in a method that can be likened to manual "stemming". For example: there is an extent to which one can reasonably account for different usages of the root word "crime". A user could prompt the bot with the word "crime" outright, signaling that they wish to know the manner, specifics, etc of the criminal's infraction. Alternatively, the user could input the keyword "criminal" itself, implying the bot should supply information based on the perpetrator. The "stem" for crime in this case hence becomes "crim".

Six distinct functions based on regular expressions were implemented to account for the six potential prompts. However, there exists intersection between possible prompts. A user could very possibly flag more than one of these functions, simply by passing a string that produces matches from more than one function. Indeed, each function is checked against the input string, and the matching string is returned to a macroscopic input processor as output. Based on truth values for the respective functions, identifiers are set indicative of what prompts are possible.

From this set of identifiers, a decision tree is in place that accounts for multiple functions producing a value that evaluates to true. One can fathom that a user could, in a single query, flag functions intended to uniquely detect questions pertaining to the investigator, criminal, and nature of crime respectively. In this implementation, the most complex cases are detected foremost; greater specificity in a user's query implies a more specific question. Subsequently, more simple cases that are assumed to be mutually exclusive are evaluated. This decision tree passes its output on to the novel processing and answer extraction workflow.

### B. Data Collection and Preprocessing

Project Gutenberg was utilized to obtain the text for each of the novels. The Python `re` package was used to scrape the website and `BeautifulSoup` was used to parse the contents, producing text ready to be analyzed.

Text processing beyond the point of data collection was done without use of any powerful libraries and was done with the help of `RegEx` module and traditional built-in data structures. The module helped the team to first extract unnecessary HTML tags and Project Gutenberg-specific information and then was heavily used to separate the cleaned novel text into chapters and sentences.

Chapter lists were produced first, which were then broken down into sentences in a nested list. The selected structure allowed for simple access to the chapter and sentence numbers, while storing the contents of the novel in an organized and easily accessible manner.

Summaries of the novels from LitCharts and Wikipedia [1] [2] [3] were used to extract the names of the novel-specific characters such as investigators, perpetrators, and suspects. The identities were stored in lists which were then used to facilitate the search for answers.

### C. Answer Extraction

Detecting occurrences of individual characters in the text and providing the details about their mentions was the underlying message for questions #1, #3, and #6 in the project description. Given the known identities of the individuals of interest, the search involved isolating and identifying the correct identity and providing the chapter and sentence details about the occurrence. This process was facilitated by our selection of the data structures used to store that information described above. The first mention of the person of interest was guaranteed to produce a correct result since all novels tend to introduce a character for the first time by stating both their first and last names.

Obtaining the first occurrence of the crime in each novel was accomplished by searching for declarations that there has been a murder of some sort (e.g. "'It means murder,' said he, stooping over the dead man"). `RegEx` was used to search for different phrasings of this with the hard-coded names of the victims, different pronouns, verbs (with different conjugations), and causes of death (two of the novels are poisonings and one is a stabbing). By looping through every sentence in the novels and searching for the same murder declaration pattern, the first occurrence of every crime and the cause of death is able to be extracted.

Identification of the three words around each mention of the perpetrator in the story is done by first searching for all sentences in which the perpetrator of a given novel is mentioned, followed by the careful extraction of the three words before and after the mention into two separate lists. With the idea for this particular search being to enable identification of relevant verbs, other characters, murder weapons, and other insightful information about the plot of the novel, focused

search within sentences increased usefulness of the produced results. Sentences were used as a natural "relevance barrier" found in the text itself.

Answering when and how the investigator and criminal co-occur follows a simple workflow: find the first incidence of the investigator and criminal being mentioned by name in the same sentence. Then, extract action words from that incident. The shortcoming here lies in the potential for investigator or criminal referencing one another resulting in a false positive; both protagonist and antagonist being mentioned simultaneously does not necessarily mean they have met.

### D. Output Processing

Output Processing is used to transform the analysis data into simple sentences that anyone can understand. It can be thought of something like this: if someone asks a question about a detective novel, we dive into the book, grab the answer, and then shape it into a clear response. This has been divided into three parts: Output Interface, Response Templates and Output Processor.

The Output Interface is responsible for organizing the extracted answers into dictionaries which are unique for each question. The Response Templates are essentially fill in the blank output strings, where the data from the Output Interface will be inserted in the appropriate places. The Output Processor combines the data from the Output Interface and the sentences from randomly selected Response Templates.

## II. Results and Findings

In *The Murder on the Links*, Renauld is the murder victim, killed by stabbing (Chapter 2, Sentence 333). The investigator is Hercule Poirot (Chapter 1, Sentence 6) . Marthe Daubreuil is identified as the criminal (Chapter 7, Sentence 261), while Jack and Bella Duveen are suspects (both in Chapter 4, Sentence 154). Hercule Poirot and Marthe Daubreuil first interact in Chapter 17, Sentence 4242.

In *The Sign of the Four*, the crimes include poisoning. The victim was Bartholomew (Chapter 5, Sentence 171). The investigator is Holmes (Chapter 1, Sentence 1). The murderer is Tonga (Chapter 11, Sentence 14). Suspects include Arthur Morstan and Jonathan Small (Chapter 1, Sentence 204, and Chapter 1, Sentence 22 respectively). Holmes and Tonga co-occur first in Chapter 3, Sentence 379.

In *The Mysterious Affair at Styles*, the crime is the murder of Inglethorp (Chapter 3, Sentence 124). The case is taken up by investigator Hercule Poirot (Chapter 2, Sentence 108). Alfred Inglethorp is the criminal (Chapter 1, Sentence 69). The list of suspects includes Alfred Inglethorp himself (Chapter 1, Sentence 49), Evelyn Howard (Chapter 1, Sentence 82), Cynthia Murdoch (Chapter 1, Sentence 71), Mary Cavendish (Chapter 1, Sentence 6), Lawrence Cavendish (Chapter 1, Sentence 6), and John Cavendish (Chapter 1, Sentence 6). Poirot and Inglethorp are mentioned together first in Chapter 1, Sentence 125.

All of the words around the perpetrators' names cannot be concisely listed in this report.

## III. Challenges

### A. Input Processing

The decision tree implemented for input processing is subjective based on what would be assumed to be common user inputs. The intention is that each evaluated case for input is increasingly general. This ideally captures more pointed inquiries appropriately, while still detecting broader questions. However, the developer necessarily imposes bias in designating artificially which prompts are the most likely.

Further, raw keyword detection is similarly limited. Given the breadth of the English lexicon, it is inevitable that some users will produce queries that confound the input processor. Consistently identifying the nature of a user's query is difficult in the scope of regular expressions alone. The designer's imagination will not universally be a superset of a given user's. Despite these shortcomings, the input processor on average identifies queries correctly. Yet, its iterative development as individual designers test independently reflects the pitfalls of this design.

### B. Data Collection and Preprocessing

Cleaning and preprocessing the data collected from Project Gutenberg posed several challenges. Code had to be written that used `RegEx` to strip out HTML tags and special characters. Two of the three novels have chapter titles in the format "chapter [Roman numeral]" while the other novel uses the format "chapter [decimal number]." The different chapter formats means chapter splitting had to support both. Finally, sentence splitting with `RegEx` required accounting for strings that contain periods that do not represent the end of a sentence. Therefore, code was written to find abbreviations and temporarily replace their periods.

### C. Answer Extraction

A limitation of searching characters' names directly is that it cannot accurately find narrators. John Watson, for example, Holmes's investigation partner, is mentioned by name only through dialogue, even though he provides useful information about himself over the course of telling the story from its start. Another challenge we had to overcome was the proper identification of related individuals with the same last name. Suspects with the same last name were identified based on their first occurrence with their full name. Also, it was difficult to extract action words that describe the "how" in some more detailed responses. Methods that search for word endings such as "-s" and "-ing" ended up being implemented but are clearly flawed.

### D. Output Processing

Because answer extraction was performed on lowercase text, proper nouns had to be recapitalized. When the chat bot listed out items in a list, the comma separation and the word "and" had to be inserted. The type of crime also had to be adjusted to be in the correct tense in the context of the output to ensure grammatical correctness.

## REFERENCES


[1] LitCharts, "The mysterious affair at styles summary," LitCharts, https://www.litcharts.com/lit/the-mysterious-affair-at-styles/summary (accessed Oct. 15, 2023).

[2] LitCharts, "The sign of the four summary," LitCharts, https://www.litcharts.com/lit/the-sign-of-the-four/summary (accessed Oct. 15, 2023).

[3] "The murder on The links," Wikipedia, https://en.wikipedia.org/wiki/The_Murder_on_the_Links (accessed Oct. 15, 2023).