Chat Regex

Andrei Cozma, Manan Patel, Tulsi Tailor, Zac Perry

Introduction & Outline

1. Novel Analysis and Findings
   - ➢ Murder on the Links
   - ➢ The Man in the Brown Suit
   - ➢ The Sign of the Four
2. Design Choices for Implementation
   - ➢ Preprocessing
   - ➢ Chat Bot Algorithm
3. Challenges Encountered
   - ➢ Data Inconsistency
   - ➢ Chat Bot Q&A
   - ➢ Error Handling
4. Demo

Novel Analysis & Findings

Objective: use basic NLP tools to observe any patterns in plot structures across three crime novels.

For each novel, the goal was to answer the following questions using an interactive chat-bot:

1. When does the investigator (or pair) occur for the first time
2. When is the crime first mentioned - the type of crime and the details
3. When is the perpetrator first mentioned
4. What are the three words that occur around the perpetrator on each mention
5. When and how the detective/detectives and the perpetrators co-occur
6. When are other suspects first introduced

Analysis: “Murder on the Links”

Author: Agatha Christie

- ***Investigator***: Hercule Poirot (Chapter 1, sentence 7)
- ***Perpetrator***: Marthe Daubreuil (Chapter 1, sentence 131)
- ***Crime***: M. Renauld’s murder (Chapter 3, sentence 90)
- ***Words around Perpetrator mentions*:** “love”, “crime”, “beautiful”
- ***Detective and Perpetrator co-occur***: (Chapter 27, sentence 157, Chapter 27, sentence 262, and Chapter 28, sentence 145)
- ***Suspects***: Renauld (Chapter 2, sentence 129), M. Bex (Chapter 3, sentence 23), Leonie (Chapter 3, sentence 90), Denise (Chapter 3, sentence 115)

Analysis: “The Man in the Brown Suit”

Author: Agatha Christie

- ***Investigator***: Colonel Race (Prologue, sentence 39)
- ***Perpetrator***: Sir Eustace Pedler (Chapter 1, sentence 5)
- ***Crime***: Nadia’s brutal death (Chapter 3, sentence 127)
- ***Words around Perpetrator mentions*****:** “Knowledge”, “Mill”, “Respecting”
- ***Detective and Perpetrator co-occur***: (Chapter 9, sentence 134, Chapter 24, sentence 188, and Chapter 24, sentence 84)
- ***Suspects***: Guy Pagett (Chapter 2, sentence 54), Suzanne Blair (Chapter 30, sentence 64)

Analysis: “The Sign of the Four”

Author: Sir Arthur Conan Doyle

- ***Investigator***: Sherlock Holmes (Chapter 1, Sentence 1)
- ***Perpetrator***: Jonathan Small (Chapter 3, Sentence 53)
- ***Crime***: Robbery of valuable treasure (Chapter 5, Sentence 175)
- ***Words around Perpetrator mentions***: Includes words such as “wooden”, “leg”, “crime”, “committed”, “associate” (Chapter 7, Sentences 228, 230)
- ***Detective and Perpetrator co-occur***: (Chapter 11, Sentence 9) and (Chapter 12, Sentence 36)
- ***Suspects***: Captain Morstan (Chapter 1, Sentence 196), Major Sholto (Chapter 2, Sentence 43), Tonga (Chapter 11, Sentence 13), and Jack Blair (Chapter 8, Sentence 35)

Design Choices: Preprocessing

Crucial in preparing textual data for analysis and NLP tasks.

After reading in the files, the following steps were taken to preprocess the files:

1. **Address the extra whitespace in the text**
    a. (i.e. consecutive spaces, newlines)
2. **Keep essential text**:
    a. such as between the START_OF_THE_PROJECT and END_OF_THE_PROJECT sections
3. **Create and insert chapter delimiters:** (<SOC>)
4. **Unify the text into coherent paragraphs**
5. **Normalize the text**
    a. Convert to Normalization Form D
    b. Strip diacritical marks while preserving the base character
    c. Translate special Unicode symbols to ASCII

6. **Insert special tokens to help chatbot**: (<EOS>, <INVESTIGATOR>, <PERPETRATOR>, <SUSPECT>, <CRIME>)

Design Choices: Chat Bot Algorithm

Chat Loop: Bot (Greet) -> User -> Bot -> User -> Bot …

1. **User message pre-processing**: Removal of stop-words, punctuation, extra whitespace
2. **Identifying user message intent**: Rule-based approach - regex pattern matching
    - Mapping the regex patterns to functions:
        - Allows performing logic such as looking up information and construct a response
        - Return the bot's response in string format
3. **Bot message post-processing:**
    - Creating variations by randomly replacing common words and phrases
        - Using curated list of suitable alternatives that work pretty generally in many contexts
    - Minor fixes, e.g.: capitalizing first word of sentence, punctuation, etc.

Bot Algorithm Data Structure Example

Parsing the pre-processed text into a data structure.
We use the special tokens we inserted during pre-processing to help with this part.

— — — — — — — — — — — —

Investigator: {
    List of matched terms: [
        "Detective",
        "Race",
        "Colonel",
        "Colonel Race",
        "Sherlock Holmes",
        "Investigator"
    ],
    Mentions: [
        {
            Matched Term: "Colonel",
            Sentence: Long life to the 'Colonel,' said the Count, smiling.,
            Sentence Num: 39,
            Chapter Num: 1,
            Chapter Title: "PROLOGUE"
        }, { … }, { … }, …
    ],
}, “Term X”: { … }, “Term Y”: { … }, “Term Z”: { … }, …

Design Choices: Chat Bot Algorithm (cont.)

For the analysis functions, we use regex capture groups to capture certain terms in the user's message

- passed along into the corresponding functions

All these utilize the data structure that was parsed from the pre-processed text.

1. **First Mention of {term}:**
    a. Lookup the term in the data structure; get the list of mentions; grab the first entry
2. **Words Around {term} (on each mention):**
    a. Lookup the term; get the list of mentions; iterate through all entries
    b. At each step: split the sentence text into words; keep track of all the words around the matched term
3. **Co-Occurrences of {term1} and {term2}:**
    a. Lookup both *term1* and *term2* in the data structure; get the list of mentions for each
    b. Iterate through their lists of mentions; find occurrences where chapter # and sentence # match

Challenges Encountered: Data Inconsistency

Chapter Splitting

- **Varying chapter heading formats**
  - Finding an effective way to normalize these
  - Examples: PROLOGUE, CHAPTER 1, Chapter IV Title, 1. Title
- **Unconventional Table of Contents structures**
  - Matching Table of Contents entries to their respective chapter headings in the text
  - Help normalize headings across the different texts, as well as aid in chapter splitting

Sentence Splitting

- Accounting for many edge-cases like abbreviations (e.g. “U. S. A.”, …), honorifics (“Mr.”, “Mrs.”, …), other punctuation
- Random spacing and newlines throughout the text

Adding search terms to the text

- Ensuring the correctness of both their placement and term

Challenges Encountered: ChatBot Q&A

- **Creating flexible rulesets and REGEX:**
  - Ensure it will be able to understand different questions well
  - Creating rulesets to properly identify when it is being asked for the detective, crime, perpetrator, etc.
- **Ensuring it has the ability to break down the questions correctly**
  - Parsing the question, extracting the key elements, finding the answer in the processed data, and returning a well-constructed response with the correct answer
- **Creating a properly organized data structure** based on the pre-processed text
  - This would be heavily relied upon by the various analysis functions of the ChatBot

Challenges Encountered: Error Handling

- **Manual searching through processed text**
  - Check instances of search term tags for the correct placement
- **Took a lot of time to manually check each question and corresponding answer**
  - Remedy: Automated testing - run through all prompt variations defined by us to ensure robustness of Chat Bot
- **Printing everything to console was getting overwhelming to interpret**
  - Remedy: Logging system - Log warnings, errors, and useful information into a log file

Demo