Project Report

- **Chatbot**

  ## System Description

  The Batman Chatbot is a natural language processing (NLP) system designed to provide accurate and informative answers to questions related to Batman, including information from movies, TV shows, and other sources about the character. The system employs machine learning techniques, specifically a Sequence-to-Sequence (Seq2Seq) model, for natural language processing tasks.

  ## Data Set Description:

- The dataset contains 500 unique question-answer pairs about Batman.
- The questions cover various topics related to Batman, such as his origin story, abilities, gadgets, allies, villains, and appearances in movies and TV shows.
- The answers were generated by Claude, likely using its knowledge base and natural language processing capabilities.

Data Cleaning:

1. Removing duplicate or near-duplicate question-answer pairs.
2. Checking for and correcting any spelling or grammatical errors.
3. Standardizing the format of the question-answer pairs (e.g., ensuring consistent capitalization, punctuation, etc.).
4. Removing any irrelevant or inappropriate content.
5. Tokenizing the questions and answers for use in the Seq2Seq model.
6. Padding the tokenized sequences to a fixed length.
7. Splitting the dataset into training, validation, and testing sets.

Additionally, the dataset has undergone further preprocessing steps specific to the Seq2Seq model architecture, such as:

1. Creating input and output sequences by combining the questions and answers in a specific format.
2. Generating vocabulary dictionaries for the input and output sequences.
3. Converting the tokenized sequences into numerical representations (e.g., one-hot encoding or embedding vectors).

By generating the dataset using an AI assistant like Claude, the process of creating and cleaning the data was likely more streamlined and efficient compared to manually curating the dataset. However, it is still essential to perform appropriate cleaning and preprocessing steps to ensure the quality and consistency of the data, as well as its compatibility with the chosen machine learning model.

ML Techniques

The core of the Batman Chatbot is a Seq2Seq model that uses an encoder-decoder architecture with Long Short-Term Memory (LSTM) units. The model is implemented using the following components:

1. **Data Preprocessing:**
   - Input and output lines are tokenized using the Tokenizer class from the 'batman.txt' dataset.
   - Tokenized sequences are padded to the maximum sequence length.
   - Vocabulary sizes are determined for both input and output sequences.
2. **Model Architecture:**
   - **Encoder:**
     - Inputs: Tokenized and padded input sequences.
     - Embedding layer: Maps input tokens to dense vectors of fixed size.
     - LSTM layer: Processes the input sequences and returns the final encoder states.
   - **Decoder:**
     - Inputs: Tokenized and padded output sequences (shifted by one token).
     - Embedding layer: Maps output tokens to dense vectors of fixed size.
     - LSTM layer: Processes the embedded output sequences using the encoder states.
     - Dense layer: Converts LSTM outputs to one-hot encoded vectors representing the output tokens.
3. **Inference:**
   - Inference models are created separately for the encoder and decoder.
   - The encoder model takes input sequences and returns encoder states.
   - The decoder model takes decoder inputs and initial states and generates output sequences.

If the Seq2Seq model cannot provide a satisfactory answer, the chatbot falls back to a document retrieval system based on the input question's complexity.

NLP Techniques:

These techniques have been reused from Project 1.

1. **Question Processing:** The chatbot identifies the type of question asked by the user and determines the type of answer it expects. This step helps the chatbot understand the user's intent and frame a relevant response.
2. **Passage Retrieval:** Using TF-IDF (Term Frequency-Inverse Document Frequency) as a feature, the chatbot generates a question vector and vectors for passages in the knowledge base. It then computes the cosine similarity between the question vector and passage vectors to retrieve the top 3 passages that closely resemble the question. To

enhance this step, stop words are removed, and a Porter Stemmer is used to normalize words.

3. **Sentence Retrieval:** After retrieving passages, the chatbot tokenizes the sentences in each passage and computes n-gram similarity between the question and each sentence. This process helps identify the most relevant sentences within each passage.
4. **Answer Processing:** Based on the expected answer type (e.g., fact-based, definition), the chatbot processes the answer sentences to identify specific entities using named-entity recognition techniques and part-of-speech tagging. This step ensures that the chatbot provides accurate and relevant information in its responses.
5. **Text Summarization:** If the type of question is a definition or if the chatbot is unable to identify named-entities from the question, it utilizes n-gram tilting technique to summarize the text. This allows the chatbot to provide concise and informative answers to user queries.

Data Sets and Cleaning

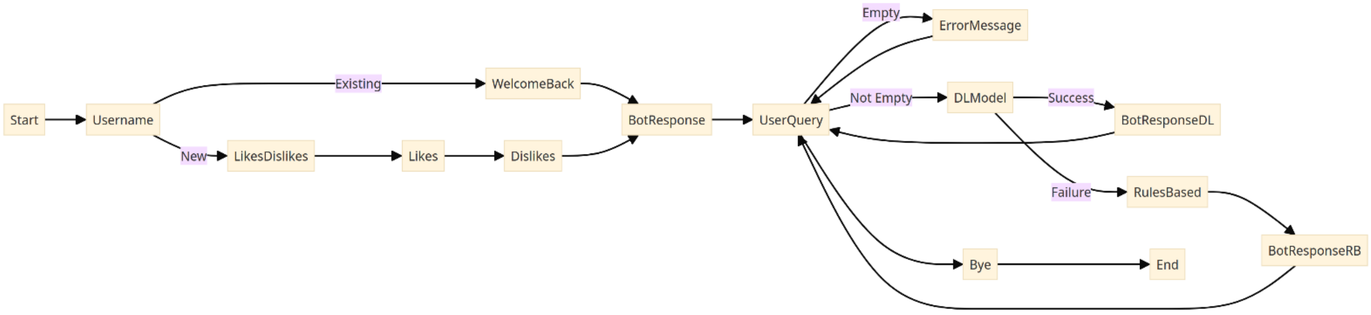
The Batman Chatbot utilizes a dataset file named 'batman.txt', which contains information about Batman. The data is cleaned and preprocessed as follows:

1. Input and output lines are tokenized using the Tokenizer class.
2. Tokenized sequences are padded to the maximum sequence length.
3. Vocabulary sizes are determined for both input and output sequences.

Dialog Tree or Logic

The Batman Chatbot follows a conversational flow based on user input. The dialog tree or logic can be summarized as follows:

1. The chatbot prompts the user for a username and stores it in a JSON file for future interactions.
2. If no command-line arguments are provided, the bot requests rerunning for proper initialization.
3. The chatbot reads the 'batman.txt' dataset file and processes the information for retrieval.
4. The chatbot interacts with the user, processing questions about Batman and providing answers based on the Seq2Seq model.
5. If the Seq2Seq model cannot provide a satisfactory answer, the chatbot falls back to the document retrieval system, depending on the question's complexity.



```
Bot: Welcome back, Ankit!
Bot: Hi Ankit! Ask me questions regarding Batman
Bot: Enter 'bye' to exit
User:  who created batman
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
Bot:  batman was created by bob kane and bill finger

User:  What is the name of Batman's butler and trusted confidant
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
Bot:  batman's butler and confidant is alfred pennyworth

User:  Who killed Bruce Wayne's parents, leading him to become Batman
Bot: Thomas

User:  What is Batman's secret identity
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
Bot:  batman's secret identity is bruce wayne a wealthy industrialist
and philanthropist

User:  Who is the Batman villain that can control plants
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Bot:  the batman villain who can control plants is pamela isley also
known as poison ivy

User:  What is the name of the serial killer obsessed with the number
13
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
Bot:  the serial killer obsessed with the number 13 is known as the
calendar man

User:  bye
Bot: Bye Bye!
```

Appendix: Sample User Model

The sample user models include the names of users who have interacted with the chatbot and stores their likes and dislikes.

Here's an example:

1. **Aaryan**

    - **Likes**: Batman Ninja
    - **Dislikes**: Don't like Batwoman arc
2. **Ankit**
    - **Likes**: Batman Under the Red Hood
    - **Dislikes**: I didn't like the new DC storyline

Chatbot Evaluation:

The chatbot performed well in terms of user-friendliness and providing helpful responses. Users found it easy to interact with and appreciated its ability to offer relevant information. However, there were instances where the chatbot encountered errors and had to fallback to a rules-based method, which impacted its overall performance. Despite these challenges, the chatbot showed promise in engaging users and providing useful responses.

Strengths:

1. **User-Friendly:** Users found the chatbot easy to understand and interact with, which enhanced their experience.
2. **Helpfulness:** The chatbot was able to provide relevant and useful responses to user queries, improving its utility.

Limitations:

1. **Error Handling:** The chatbot encountered errors at times, leading to a fallback to a rules-based method. This affected its reliability and performance.
2. **Accuracy:** Some users noted instances where the chatbot's responses were inaccurate, suggesting a need for improvement in its understanding of natural language.

Data Impact: It's worth noting that with more data, the chatbot could potentially improve its accuracy and provide more reliable responses.

Survey Results:

Based on the responses from Ankit, Sumit, Prathamesh, and Subramanian, we gathered valuable feedback on the chatbot's performance. Here are the summarized results:

Ankit:

- Clarity of Responses: 2 (Clear)

- Helpfulness: 1 (Very Helpful)
- Accuracy: 2 (Accurate)
- Error Handling: 1 (Very Well)
- Overall Satisfaction: 1 (Very Satisfied)

Sumit:

- Clarity of Responses: 2 (Clear)
- Helpfulness: 2 (Helpful)
- Accuracy: 2 (Accurate)
- Error Handling: 2 (Well)
- Overall Satisfaction: 2 (Satisfied)

Prathamesh:

- Clarity of Responses: 2 (Clear)
- Helpfulness: 2 (Helpful)
- Accuracy: 2 (Accurate)
- Error Handling: 2 (Well)
- Overall Satisfaction: 2 (Satisfied)

Subramanian:

- Clarity of Responses: 2 (Clear)
- Helpfulness: 2 (Helpful)
- Accuracy: 2 (Accurate)
- Error Handling: 2 (Well)
- Overall Satisfaction: 2 (Satisfied)