## **ATHINA AI INTERNSHIP**

DOCUMENT PREPARATION:

### **1. Installation and Importing Required Libraries**

First, we install the necessary libraries:

This installs libraries for managing large language models (LLMs), embeddings, and PDF processing.

2. **Setting Up the Environment and Loading Models:**

Here, we configure the embedding model and LLM settings.

**3. Indexing the Document:**

We load a PDF document and create an index.

**4. Defining the Prompt Template:**

We create a prompt template for the conversational AI.

* Clear and structured guidance for the AI.
* Ensures consistency in responses.

**5. Setting Up the Query Engine:**

We initialize the query engine with the prompt template.

**6.Conversational Interaction:**

**Without Retrieval-Augmented Generation(RAG):**

We handle the conversation loop without additional context retrieval.

**With Retrieval-Augmented Generation (RAG):**

We enhance the conversation by retrieving relevant context before generating responses.

**7.Summarizing the conversation:**

We have summarized it in structured format.

### Why Use This Approach?

* **Efficiency:** Automates the pre-screening process, saving time for doctors.
* **Consistency:** Ensures consistent data collection from patients.
* **Scalability:** Can handle multiple patients simultaneously.

### Advantages

* **Time-saving:** Reduces the workload for medical professionals.
* **Accuracy:** Ensures all relevant information is collected in a structured manner.
* **Flexibility:** Customizable prompts and models for different use cases.

### Disadvantages

* **Resource-intensive:** Requires significant computational resources.
* **Complexity:** Setting up and fine-tuning the system can be complex.
* **Dependence on Data Quality:** The system's performance depends heavily on the quality of input data and prompts.

**Constructing the Dataset:**

**1. Collecting Questions:**

- Source questions from a reliable dataset such as SQuAD (Stanford Question Answering Dataset), TriviaQA, or Natural Questions.

- Ensure a variety of topics and difficulty levels to have a comprehensive dataset.

**2. Collecting Reference Answers:**

- Use the provided answers in the chosen dataset as the ground truth.

- For custom datasets, have multiple human annotators provide answers to ensure quality and reduce bias.

**3. Generating Model Answers:**

- Utilize a question-answering model such as BERT, GPT, or any fine-tuned model to generate answers.

- Ensure the model has been properly trained or fine-tuned on a relevant dataset for better performance.

**Choosing Evaluation Metrics:**

**Why These Metrics Were Chosen:**

**1. ROUGE (Recall-Oriented Understudy for Gisting Evaluation):**

- Measures the overlap of n-grams between the generated and reference texts.

- Emphasizes recall, making it useful for evaluating how much of the reference answer is covered by the generated answer.

- Commonly used in summarization and text generation tasks.

**2. BLEU (Bilingual Evaluation Understudy):**

- Measures the precision of n-grams in the generated text compared to the reference.

- Good for tasks where exact match precision is important, like machine translation.

- Provides a straightforward method to evaluate the overlap of text sequences.

**3. METEOR (Metric for Evaluation of Translation with Explicit ORdering):**

- Considers precision, recall, stemming, synonyms, and word order.

- Designed to address some of the weaknesses of BLEU by incorporating more linguistic features.

- Balances precision and recall, making it useful for evaluating text where word order and synonyms are important.

How These Metrics Complement Each Other:

- **ROUGE**: focuses on recall, ensuring that the generated text captures most of the important elements of the reference.

- **BLEU**:emphasizes precision, ensuring that the generated text is concise and accurate.

- **METEOR**:balances both precision and recall, and also considers linguistic factors, providing a more holistic evaluation.

**Improving Accuracy**

**1. Fine-Tuning the Model:**

- Fine-tune the question-answering model on a specific dataset closely related to the target domain.

- Use transfer learning to leverage pre-trained models like BERT or GPT, which can then be fine-tuned on a smaller, domain-specific dataset.

**2. Data Augmentation**:

- Increase the diversity and size of the training dataset by generating synthetic questions and answers.

- Use paraphrasing techniques to create multiple variations of the same question.

**3. Hyperparameter Tuning**:

- Experiment with different learning rates, batch sizes, and other hyperparameters to optimize model performance.

- Use techniques like grid search or random search to find the best hyperparameters.

**4. Ensemble Methods**:

- Combine the outputs of multiple models to improve robustness and accuracy.

- Use techniques like model averaging or voting to aggregate the results from different models.

**5. Error Analysis**:

- Perform a thorough analysis of the errors made by the model to identify common failure points.

- Address specific types of errors by adjusting the model or the training data.

**6. Post-Processing**:

- Apply rules or additional filtering steps to refine the generated answers.

- Use techniques like answer validation or re-ranking to ensure the best possible answers are selected.

By following these steps, you can build a robust question-answering system, evaluate it comprehensively using multiple metrics, and continuously improve its accuracy through targeted techniques.