**City Resilience QA Chatbot**

This project is a Question-Answering (QA) chatbot designed to provide answers related to city resilience. The chatbot utilizes the Haystack framework along with a combination of retriever and reader models to find relevant information and answer user questions.

Prerequisites

Before running the chatbot, ensure you have the following prerequisites installed:

Python 3.6+

Streamlit

PyMuPDF (fitz)

Haystack

Flask

Flask-Ngrok

Sentence Transformers

Roberta-based model (for the FARMReader)

You can install the necessary packages using the following command:

Copy code

pip install streamlit pymupdf farm-haystack flask flask-ngrok sentence-transformers

Usage

Clone the repository or download the provided files to your local machine.

Update the pdf\_directory variable in the script with the path to the directory containing your PDF files related to city resilience.

Run the script using the following command:

arduino

Copy code

streamlit run your\_script\_name.py

A browser window will open with the Streamlit app. Enter your question about city resilience in the text input and click the "Get Answer" button.

The chatbot will retrieve relevant information from the PDF files and present the answers on the web page.

How It Works

The chatbot uses the Haystack framework to extract text from PDF files, store them as documents, and build a retriever and reader pipeline. Here's how the script works:

PDF files in the specified directory are processed to extract text using PyMuPDF (fitz).

The extracted text is stored as Haystack Documents in an InMemoryDocumentStore.

A retriever (BM25Retriever) is used to identify relevant documents based on user questions.

A reader (FARMReader) model is used to find answers within the retrieved documents.

The Streamlit app provides a user interface for entering questions and receiving answers.

Additional Notes

Make sure to replace "deepset/roberta-base-squad2" with the appropriate model name or path for the FARMReader, based on your desired model.

Adjust the parameters such as top\_k as needed for both the retriever and reader to control the number of documents retrieved and answers extracted.

Customize the web interface using Streamlit to suit your preferences.

Please note that this README is a general guide. Adjust it according to your specific use case, project structure, and any additional details you'd like to include.