Code Summary

This Python script creates a **Doctor-Patient Q&A App** using Streamlit, designed to answer user queries based on a processed doctor-patient conversation transcript. The key components and functionality of the code are:

1. **Libraries Used**:
    - **Transformers**: Uses Hugging Face's pipeline for the question-answering model.
    - **Sentence-Transformers**: Utilizes the SentenceTransformer model for encoding dialogues into embeddings.
    - **Streamlit**: Provides the user interface for asking questions and displaying answers.
    - **Torch**: Supports tensor operations for computing cosine similarity.
2. **Data Loading**:
    - The transcript is loaded from a JSON file (processed_transcript.json), and cleaned dialogues are extracted from it.
3. **Models**:
    - **Sentence Embedding Model**: all-MiniLM-L6-v2, used to encode dialogues into vector embeddings.
    - **Question-Answering Model**: distilbert-base-uncased-distilled-squad, used to generate answers based on the identified relevant section.
4. **Similarity Calculation**:
    - **Cosine Similarity**: The script finds the most relevant section of the transcript by comparing the query's embedding with the embeddings of the dialogues.
    - **Threshold**: The similarity threshold is set to 0.3, making it easier to find a match.
5. **Debugging**:
    - Similarity scores between the query and sections are printed for debugging purposes.
6. **Streamlit UI**:
    - Users input a query, and the app identifies the most relevant section of the transcript and extracts the answer. The app responds with the identified answer or a fallback message if the answer is unclear.