AI Insurance Policy Chatbot

BFSI Garage Unit POE Hackathon (TCS)

Name: Shantanu Sharma
Email: yashshar28@gmail.com

1. Introduction

Background:

- Insurance is a crucial aspect of financial planning, offering protection against a variety of risks, such as health problems, accidents, and property damage.
- With a multitude of policies available, customers often find it difficult to understand the details of these offerings.
- An AI-powered chatbot can help customers by answering their questions regarding different types of insurance policies, coverage options, claims, and premiums.

Objective of the Project:

- The primary goal is to build a chatbot that can assist users by answering questions regarding various insurance policies.
- It should be able to provide accurate, timely information about policy types, claim processes, advantages, and other related queries.
- The chatbot will leverage a knowledge base created from insurance-related PDFs and be powered by a large language model (LLM) for natural language processing.

2. Problem Statement

Challenges:

- Insurance customers often face difficulty understanding the various policies, terms, and conditions.

- ➢ Long waiting times for customer support and information overload are common issues in the insurance industry.

- ➢ Providing personalized, 24/7 customer support for complex queries is challenging.

Solution:

1) AI-powered chatbots can reduce waiting times and provide instant answers by understanding and processing customer queries.

2) A chatbot with a robust knowledge base can efficiently help customers navigate through different insurance offerings, making the process seamless.

3. Approach

Knowledge Base:

- ✧ The chatbot uses information from insurance-related PDFs (including life, health, auto, and home insurance policies).

- ✧ These PDFs are loaded and split into smaller chunks using RecursiveCharacterTextSplitter to ensure effective retrieval and comprehension.

Natural Language Processing (NLP):

- ✧ The chatbot uses a large language model (LLM), specifically the google/flan-t5-base model, for question answering.

- ✧ The model generates context-aware answers, ensuring accuracy and relevance in its responses.

Retrieval Mechanism:

- ✧ The chatbot retrieves relevant documents from the vector database (FAISS) based on user queries.

- ✧ It uses a retrieval-augmented generation (RAG) approach, where it first finds relevant documents and then uses the LLM to generate the response based on those documents.

Fallback Mechanism:

- For more complex or unresolved queries, the chatbot will guide users to human agents for further assistance.

4. Architecture

Components:

- **Knowledge Base:** Contains information extracted from insurance PDFs and stored in a FAISS vector store.
- **LLM:** A Hugging Face transformer model for text generation (google/flan-t5-base).
- **Retriever:** Uses FAISS to fetch relevant documents based on the query.
- **Pipeline:** Hugging Face pipeline to process the query and generate the response.

Flow of the Chatbot:

- Step 1: The user asks a question.
- Step 2: The chatbot retrieves relevant documents using the FAISS vector store.
- Step 3: The LLM processes the documents and generates a detailed response.
- Step 4: The answer is presented to the user in a clear and concise manner.

5. Implementation

- Code and Technologies:
  - **Libraries Used:-**
    - **LangChain:** For connecting the LLM with the retrieval system.
    - **FAISS:** For efficient document search and retrieval.
    - **Transformers (Hugging Face):** For NLP tasks and text generation.

- **Streamlit:** For building a simple user interface.

✧ Detailed Steps:

1) **Document Loading:** PDFs are loaded and pre-processed into chunks using RecursiveCharacterTextSplitter.

2) **Vector Store Creation:** The documents are embedded using HuggingFaceEmbeddings and stored in FAISS.

3) **LLM Integration:** A HuggingFacePipeline is created to generate answers using the google/flan-t5-base model.

4) **Retriever Setup:** FAISS is used to retrieve the top N relevant documents for a given query.

5) **Question Answering:** Using RetrievalQA to combine the retrieved documents and generate a response.

6. Results

✧ Performance of the Chatbot:

- The chatbot was able to answer a wide range of questions related to different types of insurance.

- The responses were clear, relevant, and accurate, based on the documents it had access to.

✧ Limitations:

- The chatbot may not always have complete answers if the relevant information is not present in the documents.

- For complex queries outside the document scope, it may redirect users to human agents.

✧ Future Improvements:

- Expand the knowledge base to include more insurance policies.

- Enhance the chatbot's conversational abilities and add more advanced features like personalization and multilingual support.

7. Conclusion

❖ Summary of Findings:

- The AI-powered insurance policy information chatbot provides quick and accurate answers to users' questions based on a robust knowledge base.

- This approach reduces customer waiting times and improves satisfaction by offering 24/7 support.

❖ Recommendations:

- Continue expanding the knowledge base.

- Improve fallback mechanisms and enhance the model for better context management.

❖ Impact:

- The chatbot can be integrated into insurance company websites and mobile apps, greatly improving customer support and decision-making processes.