

# Agentic Form-Filling Chatbot

## Automated Form Filling Chatbot

NAME: UTSAV KALRA

PHONE NO: 9811269705

EMAIL: kalrautsav88@gmail.com

## Task

- Create an intuitive chat interface for user interaction (even a notebook is fine)
- Develop an AI agent capable of guiding users through complex form-filling processes.
- Ensure data consistency and accuracy across the chat and form interfaces

## Problem Statement:

### Automated Form Filling Chatbot for Enhanced User Experience

- In today's fast-paced digital world, many users struggle with filling out lengthy and complex forms, often finding the process tedious and confusing. This leads to frequent errors, incomplete submissions, and a poor user experience, especially in areas requiring detailed information, such as insurance claims and loan applications. An automated form-filling chatbot aims to address this problem by using a conversational interface to gather information, automatically populating form fields, and providing a seamless, user-friendly alternative to traditional form completion. This solution not only saves time for users but also increases accuracy and efficiency in data collection.

## Objectives:

- To develop a user-friendly chatbot that guides users through complex forms, simplifying the process of data entry.
- To automate the population of form fields based on user input, reducing the need for manual input and increasing form submission accuracy.
- To enhance user experience by providing instant, conversational support during form completion.

- To improve efficiency by streamlining the data collection process and minimizing errors commonly associated with traditional form-filling methods.
- To enable users with varying technical skills to easily and effectively fill out forms without the need for detailed instructions or training

## Proposed Solution

---

### 1. Transformers in the Chatbot:

**Transformers** are a type of deep learning model designed to handle natural language processing (NLP) tasks like text generation, text classification, and question answering. They have revolutionized NLP by their ability to handle long-range dependencies in text and understand context. In your project, transformers play a crucial role in processing user input and generating meaningful responses.

- **Sentence Transformers:**  
These transformers are used to **encode** text into high-dimensional vectors, which represent the meaning of sentences. In this project, the **SentenceTransformer ('all-MiniLM-L6-v2')** is used to convert both user inputs and predefined knowledge (about insurance claims) into embeddings. These embeddings capture the meaning of the text, allowing the chatbot to find the most relevant responses from the knowledge base.
- **DialoGPT (Language Model):**  
The chatbot uses **DialoGPT**, which is a pre-trained conversational model. This model has been fine-tuned to generate human-like responses in a dialogue setting. When a user interacts with the chatbot, DialoGPT generates responses based on the previous conversation, ensuring the chatbot feels natural and engaging.

### Why Transformers?

- Transformers, especially pre-trained models like **DialoGPT** and **SentenceTransformers**, are effective in understanding and generating human-like text. They can generate meaningful responses even with limited context, making them ideal for a chatbot. The ability of transformers to generate contextually accurate responses ensures that users can seamlessly interact with the system, just like speaking to a real agent.

---

### 2. Large Language Models (LLMs):

LLMs, such as **DialoGPT**, are powerful AI models trained on vast amounts of text data to understand and generate language. These models have a **large number of parameters** (like billions of weights) that enable them to capture complex relationships in language.

In your project:

- **DialoGPT** acts as the core LLM responsible for maintaining the flow of conversation and generating appropriate responses to the user's questions. It ensures that the chatbot can provide conversational, context-aware replies, such as answering questions like "What is your policy number?" or "Where did the incident occur?".
  - **Purpose of LLM in this project:**  
The purpose of using an LLM like **DialoGPT** is to create an engaging, intelligent chatbot that can handle user input effectively. It brings **natural language understanding** and **generation capabilities** to the system, which are essential for real-time interaction with users.
- 

### 3. Retrieval-Augmented Generation (RAG):

RAG is an advanced technique that combines the power of both **retrieval-based** and **generation-based** methods to improve the chatbot's performance.

#### How RAG Works:

- **Retrieval:** The system first retrieves relevant pieces of information (context) from a **knowledge base**. In your case, this knowledge base contains insurance-related information like claim types, documentation requirements, and more.
- **Augmentation:** Once the context is retrieved, it is fed into a **generative model** (like **DialoGPT**). The model uses this context to generate a response that is not only accurate but also conversationally relevant.
- **FAISS (Facebook AI Similarity Search)** is used to implement the **retrieval** step. When the user asks a question, the system encodes the question and searches for the most similar entries in the knowledge base. This ensures that the chatbot uses relevant information to generate a response.
- **Augmenting with DialoGPT:** After retrieving the relevant context, it is appended to the user's input and passed to **DialoGPT** for generating the final response. This enables the chatbot to provide answers that are not only contextually relevant but also conversational.

#### Why Use RAG?

- RAG improves **accuracy** by ensuring the chatbot doesn't rely solely on pre-trained knowledge but also dynamically retrieves real-time context from the knowledge base.
- It helps the chatbot generate more **natural and context-aware responses** by augmenting the user input with relevant context from the knowledge base.

In the insurance claim chatbot, RAG ensures that even if the chatbot doesn't have an exact answer in its training data, it can still retrieve related information and provide a meaningful response.

---

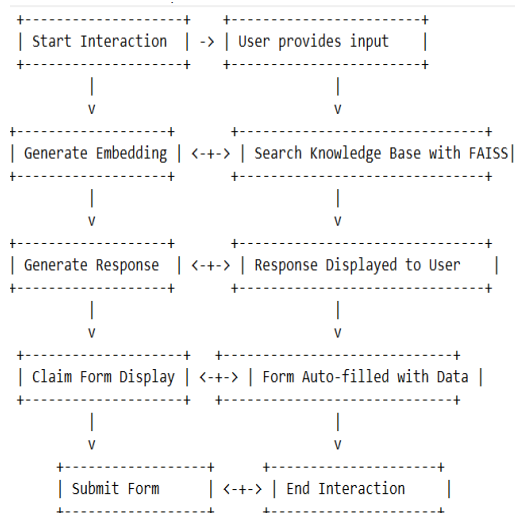
### 4. How Transformers, LLMs, and RAG Improve the User Experience:

- **Contextual Understanding:** Transformers like **SentenceTransformers** help the chatbot understand the user's queries in context. This allows the chatbot to provide personalized responses tailored to the user's situation (e.g., asking for specific claim details or policy number).
- **Human-Like Interaction:** The use of **DialoGPT** ensures the chatbot's responses are conversational and engaging. Users feel like they are chatting with a real agent, which enhances the experience.
- **Efficiency with RAG:** By combining retrieval and generation, **RAG** makes the chatbot more efficient and accurate. The system can search for relevant information in the knowledge base and generate a response based on the context, even if the exact answer is not in the model's training data.
- **Filling Forms Automatically:** The chatbot collects responses (e.g., name, policy number, incident description) and fills out the insurance claim form automatically. By using LLMs and RAG, the chatbot can ask follow-up questions when necessary, ensuring all information is gathered before submitting the form.

## Technology Stack

1. **Gradio:** UI and Web Interface for chatbot and form interaction
2. **Transformers (Hugging Face):** Pre-trained models (DialoGPT for chatbot)
3. **DialoGPT:** Pre-trained conversational model
4. **Sentence-Transformers:** Sentence embeddings and context retrieval
5. **FAISS:** Similarity search for context retrieval
6. **Python:** Programming language for backend logic
7. **PyTorch:** Neural network and model framework
8. **NumPy:** Numerical computations and vector manipulation
9. **Google Colab:** Development and model execution environment
10. **Hugging Face Model Hub:** Access to pre-trained models

## Activity Diagram



## Results 1 -CHATBOT ALONE

The chatbot successfully captured the essential information from the user's responses, while ignoring unnecessary words. For example, when the user typed, "my name is utsav," the chatbot extracted only "utsav" for the "Full Name" field. Similarly, when asked about the claim type, the chatbot identified "Accident" from the user's response.

```

File Edit Format View Help
Hello! I'll help you file your insurance claim. Let's start with your full name.
User: my name is utsav
Thanks, utsav. Can you provide your policy number?
User: 123abc
Got it. What is the type of claim you're filing today? Options are: Accident, Theft, Natural Disaster, Medical, or Other.
User: i am filling for accident
Understood. Could you tell me the date of the incident? Please use YYYY-MM-DD format.
User: 2003-05-06
Thanks. Please provide a brief description of the incident.
User: jnhnkn
How much are you claiming for this incident? Please specify the amount.
User: around 20000
Did you file a police report for this incident? (Yes/No)
User: no i did not
Thank you. Where did the incident occur? Please provide the location.
User: delhi
Was anyone injured in the accident? (Yes/No)
User: no
Finally, could you tell me if you have any additional documentation? Options are: Police Report, Medical Report, Photographs, Invoices/Receipts, or None.
User: yaa maybe medical report

Thank you! Your claim form has been completed. Here are the details:
Name: utsav
Policy number: 123abc
Claim type: Accident
Claim date: 2003-05-06
Incident description: jnhnkn
Claim amount: 20000
Police report: No
Incident location: delhi
Injuries: No
Additional docs: Medical report
|
  
```

## Results 2 -CHATBOT WITH FORM

chatbot x Gradio x Gradio x Gradio x Gradio x No inte x Introdu x Meta x New Tai x +

3bb868a04afa720949.gradio.live

### Insurance Claim Form & Chatbot

Chatbot

Textbox

Type your response here...

Finalize Claim Details

Full Name

Policy Number

Claim Type

Claim Date (YYYY-MM-DD)

Incident Description

Claim Amount

Police Report Filed?

chatbot x Gradi x No in x Gradi x Gradi x Gradi x No in x Introdu x Meta x New x +

07894b92849899ec44.gradio.live

### Insurance Claim Form & Chatbot

Chatbot

I'll help you file your insurance claim. Let's start with your full name.

user

my name is utsav

bot

Thanks, utsav. Can you provide your policy number?

user

Textbox

Type your response here...

Finalize Claim Details

Full Name

utsav

Policy Number

123abc

Claim Type

Accident

Claim Date (YYYY-MM-DD)

2003-05-06

Incident Description

scdcds

Claim Amount

2000

**Insurance Claim Form & Chatbot**

Chatbot

bot

Understood. Could you tell me the date of the incident? Please use YYYY-MM-DD format.

user

2003-05-06

bot

Full Name

utsav

Policy Number

123abc

Claim Type

Accident

Claim Date (YYYY-MM-DD)

2003-05-06

Incident Description

scdcds

Claim Amount

2000

Textbox

Type your response here...

Finalize Claim Details

Textbox

Type your response here...

Finalize Claim Details

Full Name

utsav

Policy Number

123abc

Claim Type

Accident

Claim Date (YYYY-MM-DD)

2003-05-06

Incident Description

scdcds

Claim Amount

2000

Police Report Filed?

Yes

Incident Location

delhi

Injuries?

Yes

Additional Documentation

None

Submit Details

Use via API • Built with Gradio

## Limitations

1. **Limited Understanding of Complex Inputs:** The chatbot struggles with ambiguous or complex user queries.

2. **Lack of Error Handling:** The system does not handle invalid inputs, such as incorrect date formats or non-numeric values.
3. **Static Knowledge Base:** The knowledge base doesn't update automatically, requiring manual updates for new information.
4. **No Deep Learning for Context:** The chatbot follows predefined rules and doesn't adapt to highly varied or dynamic inputs.
5. **Limited Personalization:** The system doesn't provide personalized recommendations based on user history.
6. **Language and Regional Restrictions:** The chatbot may not support multiple languages or understand regional variations.
7. **Data Security Risks:** The system lacks advanced security measures like data encryption, making sensitive information vulnerable.

## FUTURE IMPROVEMENTS

1. **Enhanced Natural Language Understanding:** Implement more advanced NLP techniques for better understanding of complex and varied user inputs.
2. **Dynamic Knowledge Base:** Enable automatic updates to the knowledge base from external sources or databases to ensure current and relevant information.
3. **Context Awareness:** Improve the chatbot's ability to maintain context across multiple interactions and provide more meaningful responses.
4. **Multi-language Support:** Add support for multiple languages to cater to a wider range of users.
5. **Advanced Personalization:** Introduce user profiling and personalized responses based on user data and history.
6. **Error Handling and Validation:** Implement better input validation and error handling to guide users in case of incorrect data or system errors.
7. **Voice and Visual Integration:** Incorporate voice recognition and visual components to create a more interactive and accessible experience

## CONCLUSION

In conclusion, the automatic insurance claim chatbot provides an efficient and user-friendly solution for filing claims. By leveraging advanced NLP techniques, transformers, and knowledge retrieval systems, the chatbot is able to understand user queries and automatically fill out necessary forms with relevant information. Gradio, as the frontend tool, facilitates seamless interaction with users, providing an intuitive interface. While the current system offers a solid foundation, there is significant potential for future enhancements such as improved context awareness, multi-language support, and increased security. With continued development, this system can transform the claim filing process, making it faster, more accessible, and less error-prone for users.