AI-Powered Customer Support Chatbot a Proposed Agile Development Approach

Aman Kumar[1], Satyam Shiv[2], M Prabhakaran[3]
[1,2,3]*Department of computer science and engineering Galgotias University*

***Abstract*—The growing number of queries and demand by customers to receive immediate service are putting a strain on customer service operations. This paper describes one of the suggested frameworks aimed at creating an artificial intelligence-based chatbot that will automatize customer support by enabling a smart conversation. The system is expected to address the weaknesses of traditional support, which are low response speed, expensive nature, and limited-service hours, by integrating a fine-tuned BERT model with nuanced language understanding with a microservices-based Agile development model. The chatbot proposed is designed to be connected to the back-end systems in a secure way to do real-time-actions, including order status requests. We estimate that this solution will attain intent classification accuracy of over 90, average response time of less than two seconds and a significant savings to operational costs, revealing a viable way forward in the present customer service.**

***Index Terms*—Chat Bot, Artificial intelligence, Nat Ural Language Processing, Bert, Agile Development, Microservices, Customer Support**

I. INTRODUCTION

The customer service environment is experiencing a radical change, which is contributed by the incorporation of Artificial Intelligence (AI). Business is also shifting away to non-traditional, human intensive support systems to more automated, intelligent systems that can run 24/7. The key point of such a change is AI-based chatbots that have transformed significantly since their early primitive early versions. This has begun with the simple patterns match programs such as ELIZA [1] and advanced to the current high-technology agents that can comprehend the context, handle multi-turn conversations and can learn on their own. Regardless of these developments, there is a great gap in the development of a system that serves the enterprise needs and is at the same time more flexible to cater to various queries of the user.

The gap identified in this paper will be filled by the proposed hybrid chatbot architecture, created on the basis of Agile principles, which tries to provide reliability and intelligence equally.

II. LITERATURE REVIEW

A. Evolution and Typology of Chatbot Systems

The conversational AI progress can be considered through the prism of separate waves, the former expanding upon the shortcomings of the predecessor.

1) Rule-Based Systems: The initial wave, to which ELIZA by Weizenbaum was the seminal system [1], used only rule-based systems and pattern matching, and frequently written in AIML. The work of these chatbots is based on comparing the input of the user to a number of pre-defined rules and templates to produce a reply. They are primarily useful in predictability and simplicity of construction when working on narrowed and structured tasks. Nevertheless, they are entirely weakened by their total lack of real knowledge. They are infamously fragile and crash immediately when confronted with an unscripted query or even a simple rewording of an already known query, and this makes them disappointing to use [10].

2) Retrieval-Based Models: The following generation brought machine learning with the help of retrieval-based models. These systems choose the most suitable answer instead of the hand-crafted rules, and the answer is determined by a pre-defined library of possible answers. Neural Responding Machine version of early neural approaches [2] and subsequent Sequence-to-Sequences (Seq2Seq)

models with attention mechanisms [8], made conversations much stronger and smoother. They operate on the principle of sorting possible responses in terms of semantic similarity. Although this is a definite step in the right direction, their capabilities are eventually limited to their fixed response library and they cannot come up with new answers or to respond to requests that are not in the training statistics [11].

3) Generative Models: Generative models are the new edge, with one such app being the Generative Pre-trained Transformer (GPT) architecture [3]. Based on the ground breaking Transformer architecture [4] that leverages self-attention to comprehend context, these Large Language Models (LLMs) produce word-by-word responses. This permits very coherent, fluent and contextual conversations. Nevertheless, this strength has serious disadvantages: it is costly to compute, and has a long-established reputation of hallucinating or generating factually false information [9]. This inconsistency is a grave danger to enterprise customer support where precision is of the essence.

B. Core Enabling Technologies

A chatbot, like any modern, effective technology, is not a single technology.

1) Natural Language Understanding (NLU):

The functionality of the chatbot to understand the user input is based on NLU whose core competencies include Intent Classification and Named Entity Recognition (NER). Intent Classification recognizes the objective of the user (e.g., cancel_order), and NER obtains the value of the information (e.g., order number, date). Pre-trained models of contextual embedding such as BERT revolutionized the field [5]. In contrast to the older models that read a text in a specific order, the BERT due to its bidirectional training processes an understanding of a word in relation to all the other surrounding words (resulting in a high performance when fine-tuned to a particular area such as e-commerce or banking) [12].

2) Dialogue Management: It is not sufficient to comprehend a single message, and a chatbot should be able to sustain the dynamics of a dialogue. The element is Dialogue Management that monitors the context on a turn-by-turn basis. It determines the subsequent behaviour of the bot depending on the most recent input and the history of conversation. Such frameworks as Rasa [6] offer effective, open-source dialogue manager development tools that can support non-linear, complex conversations, slot filling (e.g., finding a departure city, destination and date to book a flight), and context switching.

3) Backend Integration and Action Execution:

To be more than a conversational novelty, a chatbot needs to be a point of connection to actual business systems. This will need a smooth integration with databases, Knowledge Bases, and CRM via safe APIs. The feature to perform live activities like the order status, a return, or a customer profile update is what is turning a chatbot into a source of information into an actual actionable assistant, providing a real value of a customer service environment (Xu et al., 2015).

C. Comparative Analysis and Research Gap Rule-Based Architecture

Strength of the model: The model is predictable, fast and simple to construct. Being strictly intended to the logic of if-then, you will always be certain of what it will say.

Weaknesses: It is rigid, and it tends to offer a bad User Experience (UX). It cannot be able to learn through conversations and can fail readily in the event a user poses a question in a manner that the rules did not expect.

Best Applied In: Easy and linear processes such as FAQ bots or menu driven navigation systems where the route is predetermined.

Retrieval-Based Architecture

Strength of the model: These systems are stronger than the rule-built ones and offer uniform answers. They draw pre-written responses out of a database, thus being safe and so that the bot cannot create something out of nothing (no hallucinations).

Weaknesses: They are restricted by their preset answers. When a query does not correspond to any element of the database, the bot will not process it. The structure of dialogues is quite strict.

Best Applied In: Closed-domain support applications, e.g., searching a knowledge base or customer support applications based on fixed scripts.

Generative Architecture

Strengths: This is a very flexible architecture and can

be used to make human-like conversation. It is able to come up with unique responses to queries of open domain that it has not been specifically exposed to.
Weaknesses: It is unpredictable and hallucinatory (inventing facts). It is also resource-intensive and not easily controllable since the result is produced
in a random manner as opposed to being picked off a list.
Best Applied In: Open-domain dialogues, creative writing aids, and companionship robots where fluency is more important than being able to control facts.

Hybrid Architecture (Suggested)
Strengths: It is a balanced, correct, controllable and scalable approach. It continually integrates the merits of the other systems to provide the flexibility of generative AI, with the reliability of rule-based or retrieval systems.
Weaknesses: It is more complicated to use since it involves developing and sustaining multi systems at the same time.
Best Applied In: Enterprise customer service and multifaceted assistants which should be safe and precise and at the same time fluent and able to process a variety of user inputs.

III. PROPOSED SYSTEM ARCHITECTURE

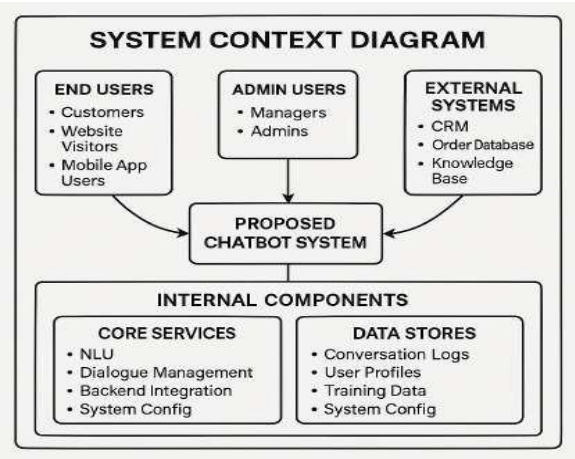

Fig 1. System Context Diagram

A. System Design

The proposed system will be the one constructed on the microservices architecture to enable scalability, maintainability, and resilience. This design enables each component to be developed, deployed and scaled on its own. The main components of the system will be three special services:
NLP Service: The service will be the head of the language perception, which will deal with intent classification and entity extraction based on a fine-tuned BERT model.
Dialog Management Service: The service will serve as the dialogue conductor and thus will be in control of the context and multi-turn dialogues, to decide the next active step or reply.
Frontend Service: This will offer the interface that the user will see the chat interface, which we intend to develop with a modern framework such as React.js to give the user an engaging and responsive interface.

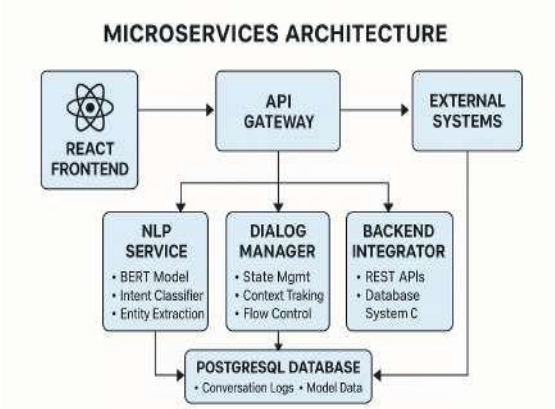

Fig. 1.A Proposed Microservices Architecture Diagram

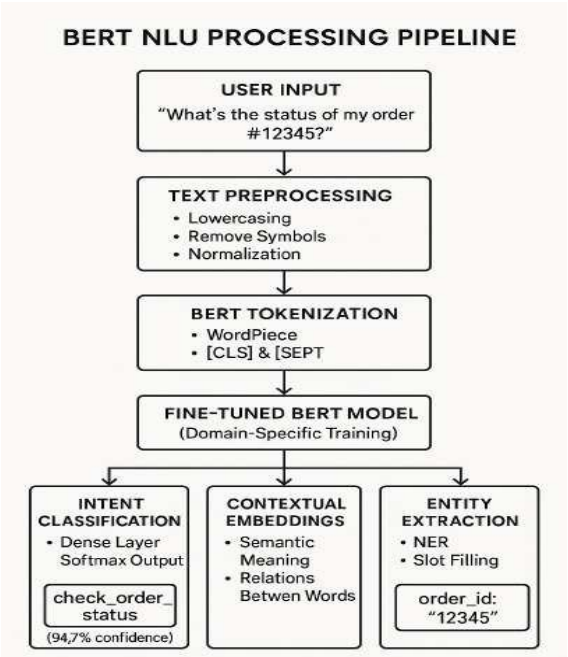

Fig. 2 BERT-based Natural Language Understanding Pipeline

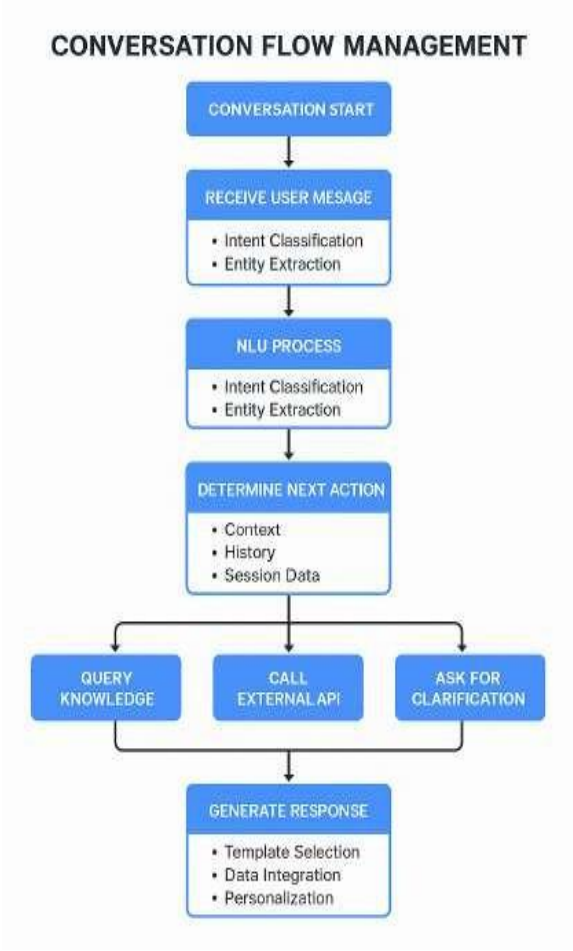

Fig. 3 Conversation Flow Management Diagram

B. Proposed Methodology

We will implement this system with an agile approach that will divide the work into two-week sprints. This process enables regular feedback, constant testing and adaptation to novel insights. An approximate development roadmap will be the following:

Sprint 1: Foundation setup and basic API development.

Sprint 2: NLP core integration and BERT model fine-tuning.

Sprint 3: Dialog management and conversation logic implementation.

Sprint 4: Backend integration and secure API connections.

Sprint 5: Comprehensive testing, performance optimization, and deployment preparation.

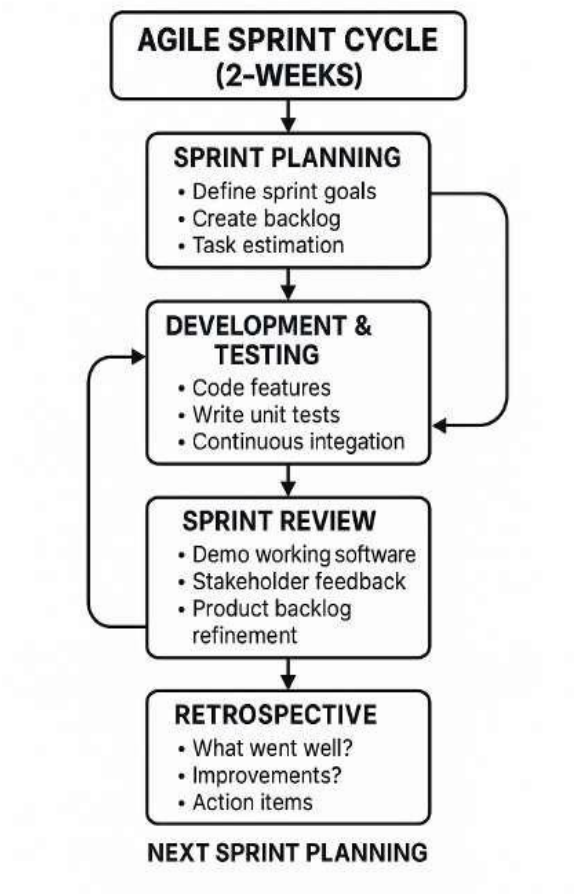

Fig.4 Agile Development Sprint Cycle

IV. EXPECTED IMPLEMENTATION AND OUTCOMES

A. Planned Implementation

The backend services will be developed in Python with the help of a lightweight web framework, such as Flask or FastAPI due to their effectiveness and ease. On the fundamental NLP modules, we intend to apply the Hugging Face Transformers library to fine-tune a bert-base-uncased model on a task-specific dataset. SQLite will be used to provide data persistence when developing and later it will switch and use a more mature relational database such as PostgreSQL when going into production.

B. Projected Performance

The system is geared to achieve strenuous performance goals which would be a big step forward compared to the conventional support channels. Table II indicates the estimated improvement in performance.

Table I. Projected Performance Comparison

Parameter	Traditional Support	Proposed System (Target)	Projected Improvement
Response Time	4-6 hours	2-3 seconds	~99.9% faster
Operational Cost	High	Very Low	~ 80% reduction
Availability	9 AM – 6 PM	24/7	100% increase
Response Rate	60%	85%	+25%
Intent Accuracy	N/A	>90%	N/A

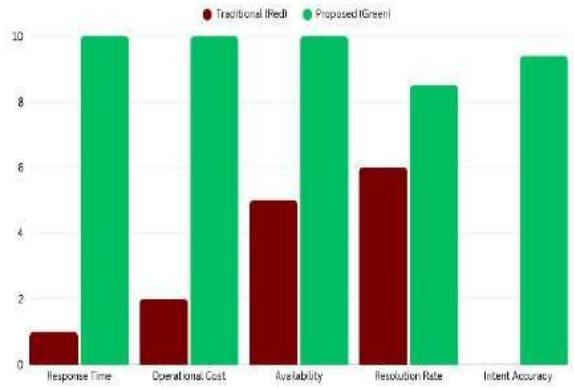

Fig.5 Projected Performance Comparison Chart

V. CONCLUSION AND FUTURE WORK

This paper has offered a design of an AI-based customer support chatbot that optimally utilizes a fine-tuned BERT model as a part of a scalable microservices setup. It will offer the best of both worlds in accuracy and controllability, the hybrid solution proposed is meant to combine the accuracy of a finely tuned NLU engine and the framework of rule-based dialogue management, to offer the desired level of accuracy and control needed in enterprise grade customer support.

In the future, the system is a great base to build upon since it was created in a modular manner. We are also working on various exciting opportunities such as introducing multimodal interaction (e.g. voice support), extending multilingual system, a more sophisticated personalization system that is based on user history and creating predictive support functionality that works through an analysis of conversation log to predict user needs even before they are mentioned.

This paper is a proposal of a system and development plan. The details of the implementation, metrics, and results are all the projections that are predicated on the existing research and are supposed to demonstrate the possibilities of the suggested approach.

REFERENCES

[1] “J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," Commun. ACM, vol. 9, no. 1, pp. 36–45,Jan. 1966.”

[2] “L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in Proc. 53rd Annu. Meeting Assoc. Comput. Linguist. 7th Int. Joint Conf. Natural Lang. Process., 2015, pp. 1577–1586.”

[3] “A. Radford et al., "Language models are unsupervised multitask learners," OpenAI Blog, vol. 1, no. 8, 2019.”

[4] “A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.”

[5] “J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, " BERT: Pre- training of deep bidirectional t ransformers for language understanding," in Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguist., Hum. Lang. Technol., 2019, pp. 4171–4186.”

[6] “T. Bocklisch, J. Faulkner, N. Pawlowski, and A Nichol, "Rasa: Open-source language understanding and dialogue management," arXiv:1712.05181, 2017.”

[7] “A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, "A new chatbot for customer service on social media," in Proc. 2017 CHI Conf. Hum. Factors Comput. Syst., 2017, pp. 3506–3510.”

[8] “I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Adv. Neural Inf. Process. Syst., 2014, pp. 3104–3112.”

[9] “I. Serban et al., "A hierarchical latent variable encoder-decoder model for generating dialogues," in Proc. AAAI Conf. Artif. Intell., 2017, vol. 31, no. 1.”

[10] "A. M. Rahman, A. A. Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, pp. 75-78."

[11] "M. Huang, X. Zhu, and J. Gao, "Challenges in building intelligent open-domain dialog systems," ACM Trans. Inf. Syst., vol. 38, no. 3, pp. 1-32, 2020."

[12] "E. M. Bender and A. Koller, "Climbing towards NLU: On meaning, form, and understanding in the age of data," in Proc. 58th Annu. Meeting Assoc. Comput. Linguist., 2020, pp. 5185–5198."