A GenAI-Powered Arabic Chatbot for E-Commerce Customer Service

Ahmed Shenawy
Nile University
Cario,Egypt
A.AbdElhamid2417@nu.edu.eg

Abstract—This paper presents the design and implementation of an intelligent question-answering chatbot for Arabic e-commerce customer support. Leveraging Retrieval-Augmented Generation (RAG) alongside AraBERT and OpenAI embeddings, our system seamlessly integrates data scraping, cleaning, Arabic text normalization, vector database storage, and dynamic prompt construction. Serving users over a Streamlit-based interface, it achieves high relevance and low latency while respecting Arabic language nuances

I. Introduction

E-commerce platforms increasingly rely on intelligent customer support systems to enhance user experience and scale operations. However, Arabic-speaking markets remain underserved due to the complexity of the Arabic language and the scarcity of tailored Genarative Artificial Intellegence (GenAI) solutions.

Most existing customer support chatbots are optimized for English and lack robust Arabic NLP capabilities. Specifically, few systems combine retrieval-augmented generation (RAG) with Arabic semantic understanding, leaving a gap in delivering accurate, context-aware responses for Arabic users.

This paper presents a modular, Arabic chatbot designed for e-commerce platforms. The system integrates:

- AraBERT and OpenAI for accurate Arabic text embedding
- Qdrant for fast semantic retrieval
- OpenAI LLMs for dynamic, natural language responses.
- Coupled with a Streamlit-based interface

The chatbot achieves high relevance, low latency, and respects the nuances of Arabic, offering a scalable solution for multilingual customer support.

II. Literature Review

Recent advancements in natural language processing (NLP) and generative AI have transformed the landscape of customer support systems. In this work, we explore a broad range of literature spanning chatbot architectures, retrieval-augmented frameworks, and Arabic language models to inform and shape our proposed system.
Early chatbots relied on deterministic, rule-based scripts with limited contextual awareness. The emergence of large language models (LLMs) marked a shift toward more adaptive and natural dialogue. RAG became a foundational technique by combining neural retrieval with generative reasoning. This hybrid approach has been shown to increase factual consistency and mitigate hallucination in responses.
To address Arabic language complexities, we examined various Arabic NLP models and tools. AraBERT proposed transformer-based model pre-trained on diverse Arabic corpora. It achieves state-of-the-art performance on tasks such as sentiment classification and question answering, making it a strong candidate for embedding-based retrieval in Arabic contexts.

We also reviewed methods for scalable semantic search using vector databases. Qdrant has emerged as a leading open-source solution, offering efficient indexing and approximate K-Top Products search suitable for real-time GenAI pipelines. Additionally, literature on few-shot prompting demonstrates its effectiveness in improving the quality of LLM outputs, especially in domains with limited labeled data or low-resource languages.

Despite these individual advances, there remains a lack of integrated systems that unify RAG, Arabic language modeling, and real-world e-commerce deployment. Our review highlights this gap and motivates the development of an Arabic-friendly, GenAI-powered chatbot for customer support that bridges these technologies.

III. METHODOLOGY

The proposed approach integrates multiple technologies to handle end-to-end customer queries in Arabic. The methodology includes data acquisition, preprocessing, semantic embedding, and generative prompt construction.

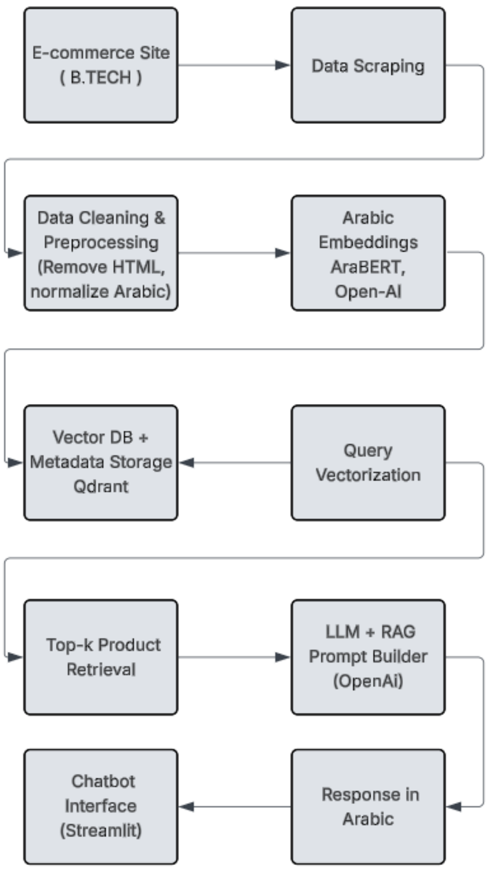
The proposed Arabic e-commerce chatbot is designed as a generic and extensible framework applicable to a wide range of commercial websites, including but not limited to B. TECH and El Araby Group. The system operates as a full pipeline, starting from raw HTML scraping of product pages and culminating in a final user-facing response generated through a Retrieval-Augmented Generation (RAG) architecture. This end-to-end process is illustrated in *Figure 1*, which outlines each stage of the workflow from data ingestion to intelligent Arabic response generation.

A. Data Scraping

The initial phase employs an automated headless browser to render and navigate dynamic, JavaScript-driven pages of any Arabic e-commerce site (e.g., B.TECH, El Araby Group). To capture all listings, the scraper programmatically scrolls through product catalogs and triggers “load more” actions until the full set is visible. From the fully rendered listing view, it extracts each item’s name, price, and detail-page URL. The scraper then visits each product page to locate and parse the specification section identifying key value attribute pairs such as capacity, dimensions, and features and aggregates these details into a structured record. All collected entries are consolidated into a tabular dataset comprising product names, prices, links, and combined specifications, thereby creating a uniform input for subsequent preprocessing and embedding.

B. Data Cleaning:

To prepare the raw Arabic text for downstream semantic processing, we apply a multi-stage cleaning and normalization pipeline. First, all punctuation—both standard ASCII symbols and language-specific marks—is stripped to eliminate extraneous characters. Next, Arabic-specific diacritics (tashkeel) are removed to reduce orthographic variation. We then eliminate common stopwords using a comprehensive Arabic stopword list, thereby retaining only content-bearing terms Finally, each remaining token undergoes lemmatization via a pretrained Arabic NLP pipeline, which converts inflected forms to their base lemmas. This sequence of normalization steps produces clean, consistent textual inputs, ensuring that subsequent embedding and retrieval stages operate on semantically meaningful units.



E-commerce Site (B.TECH)
Data Scraping
Data Cleaning & Preprocessing (Remove HTML, normalize Arabic)
Arabic Embeddings AraBERT, Open-AI
Vector DB + Metadata Storage Qdrant
Query Vectorization
Top-k Product Retrieval
LLM + RAG Prompt Builder (OpenAi)
Chatbot Interface (Streamlit)
Response in Arabic

Figure1: Arabic Chatbot Pipeline

C. Embedding

To obtain dense semantic representations of Arabic product information, we utilize a pre-trained BERT-based Arabic encoder. Input texts (both product names and specification descriptions) are first tokenized and mapped into token embeddings via Open-AI and Arabic BERT model. We then apply mean pooling across the token dimension, weighted by the attention mask, to produce a fixed-length vector for each text instance. This process generates 768-dimensional embeddings that capture contextual and morphological nuances of Arabic, which are subsequently stored in the vector database for efficient similarity search and retrieval.

D. Indexing

It constitutes the initial stage of our pipeline, in which product embeddings and associated metadata are organized for efficient semantic retrieval. We instantiate a vector database collection with two distinct 768-dimensional vector fields, one for product specifications and one for product names, both using cosine distance as the similarity metric as shown below figure 2:-

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\cdot\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Figure2: Cosine Similarity

Each product is represented by its specification and name embeddings alongside metadata (e.g., item name, price, link, category) and assigned a globally unique identifier. Once the collection schema is defined, all product vectors and payloads are batch-uploaded to the database. At query time, the user's query is embedded using the same embedding model, and a top-k nearest-neighbor search is executed against the indexed vectors. The result is a ranked list of products whose semantic representations most closely match the user's intent, forming the basis for prompt construction in the subsequent RAG response generation.

E. LLM and RAG

To generate contextually grounded Arabic responses, the system employs a two-stage Retrieval-Augmented Generation (RAG) process coupled with a large language model (LLM). First, the user's query is normalized and transformed into a dense semantic vector using the same Arabic-specific preprocessing and embedding pipeline applied to the product catalog. This query embedding is then submitted to the vector database, which returns the top-k most semantically similar product entries along with their metadata (name, price, link). The retrieved items are formatted into a structured prompt, listing each product in order of relevance.

Next, this prompt is combined with a tailored system instruction and a set of few-shot Arabic examples to steer the LLM's output towards concise, accurate, and domain-appropriate recommendations. The system instruction defines the assistant's role as an expert in electrical appliance description, specifies the criteria for selecting and ordering products, and outlines the expected response format. The few-shot examples illustrate ideal user–assistant exchanges for refrigerator and washing machine inquiries. Finally, the fully assembled message sequence is sent to LLM, which produces a fluent, context-aware answer in Arabic, explaining why the recommended products best match the user's needs.

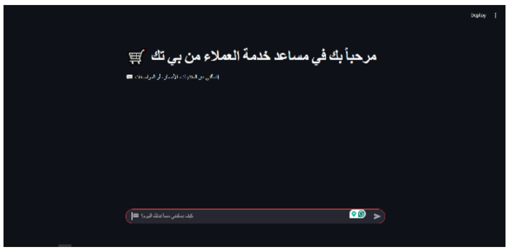
F. Chatbot UI

The chatbot interface is designed with a focus on usability and native Arabic support, offering a conversational environment that mimics modern messaging applications. The layout provides a clean and centered design, featuring a welcoming message and prompt-based interaction to guide the user. Message history is preserved across interactions, ensuring continuity and context awareness. The assistant responds in real time, and user inputs are dynamically processed and appended to the conversation stream. Visual enhancements and clear separation between user and assistant messages improve readability and user engagement, making the chatbot intuitive and accessible to a broad audience.

IV. Results

To validate the feasibility and usability of our Arabic e-commerce chatbot, we deployed the complete pipeline including data ingestion, RAG backend, and the Streamlit-based front end and conducted a hands-on demonstration with stakeholders.

The Streamlit interface in Figure 3 delivers a responsive, chat-style experience in native Arabic.



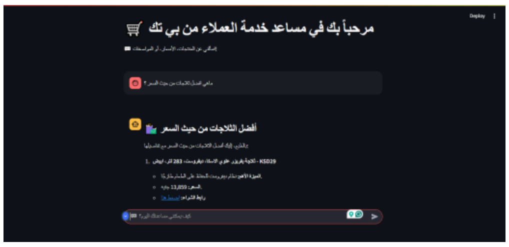
مرحباً بك في مساعد خدمة العملاء من بي تك 🛒

اسألني عن المنتجات، الأسعار، أو المواصفات

كيف يمكنني مساعدتك اليوم؟

Figure3: Arabic Chatbot UI

The deployed chatbot successfully handles a wide range of product-related queries including direct comparisons, "best overall" and "most cost-effective" recommendations and efficiently answers all frequently asked questions (FAQs) in fluent Arabic allowing users to scroll back through their session as shown in figure 4 below:



مرحباً بك في مساعد خدمة العملاء من بي تك 🛒

اسألني عن المنتجات، الأسعار، أو المواصفات

ما هي أفضل ثلاجات من حيث السعر؟

أفضل الثلاجات من حيث السعر

يسعدني إليك أفضل الثلاجات من حيث السعر مع تفاصيلها

1. ثلاجة بفريزر علوي [illegible] 283 لتر، ابيض - KSD29
   - الميزة الأهم: نظام ديفروست [illegible]
   - السعر: 13,859 جنيه
   - رابط الشراء: اضغط هنا

كيف يمكنني مساعدتك اليوم؟

Figure4: Call Center Query

The deployed chatbot successfully handles a wide range of product-related queries including direct comparisons, "best overall" and "most cost-effective" recommendations and efficiently answers. The GenAI-powered Arabic chatbot service achieved highly efficient and accurate responses when guided by the few-shot prompting strategy.

Table 1 presents the results of our manual qualitative evaluation. Each system output was independently assessed and categorized into four classes: Correct, Relevant, Partial, and Wrong. The review process involved subjective judgment based on domain-specific criteria. Out of 24 evaluated outputs, the majority (18) were fully correct, while a small portion demonstrated partial relevance or minor errors. This analysis highlights the model's strong performance in producing accurate and contextually appropriate responses.

cleaning and normalization, embedding generation using AraBERT and OpenAI models, and efficient semantic retrieval with Qdrant, all orchestrated through a Retrieval-Augmented Generation (RAG) framework. The system is exposed via a Streamlit interface that delivers real-time, context-aware responses in fluent Arabic, supporting product queries, comparisons, "best value" recommendations, and FAQs.

Evaluation Result	**Count**
Correct	18
Relevant	3
Partial	2
Wrong	1
Total	**24**

Table1: Qualitative Evaluation

V. Conclusion

In this work, we presented the design and implementation of a GenAI-powered Arabic chatbot tailored for e-commerce customer support. Our modular pipeline integrates web scraping, Arabic text

Although a formal end-to-end evaluation is reserved for future work, preliminary demonstrations and stakeholder feedback indicate that the chatbot offers high relevance, low latency, and seamless user experience. The few-shot prompting strategy successfully guides the underlying LLM to generate concise, accurate recommendations aligned with user intent.

I. Future Work

Future directions include a comprehensive user study to quantify retrieval accuracy, in-depth performance benchmarking, expansion to Arabic dialects, and integration of voice and multimodal inputs. By bridging advanced Arabic NLP techniques with state-of-the-art generative models, our solution lays the groundwork for scalable, high-quality customer support across Arabic-speaking e-commerce platforms.

References

[1] Kalva R. Optimizing E-commerce Platforms with GenAI-Driven DevOps and LLMOps: A Scalable Framework for Enhanced User Experience. *Journal of Artificial Intelligence, Machine Learning and Data Science*, 2024; **2(4)**:1782–1788.

[2] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W, Rocktäschel T, Riedel S, Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv preprint arXiv:2005.11401v4 [cs.CL], 12 Apr 2021.