# ASKIFY: Q/A PLATFORM

By Khushi Jain

# Acknowledgement

The joy that goes along with the successful completion of any task would be imperfect without the mention of people whose endless cooperation made it achievable, whose continuous upliftment and guidance crown all the efforts with success. We would like to thank Dr. Sushil Kulkarni, Dean, Nilkamal School of Mathematics, Applied Statistics & Analytics NMIMS.

We would like to thank our faculty guide Mr. Pratik Hatwalne who is the biggest driving force behind the successful accomplishment of this project. He was there all the time to solve any query of ours and also steered us in the right direction in regard to this project. Without his ceaseless support and inspiration, this paper would have been incomplete. We would also like to thank our batch mates who helped us and gave interesting suggestions and motivation at every step.

M.Sc. Statistics and Data Science (2023-25)
NSoMASA, NMIMS

# Contents

# 1 Abstract

This report introduces Askify, a Q&A chatbot utilizing Generative AI (Gen-AI) with Retrieval-Augmented Generation (RAG) to deliver precise and contextually aware responses to user queries. The system integrates OpenAI's GPT language models with a retrieval mechanism using FAISS and other components to process PDF and web content, support multilingual interactions, and enable conversational memory. Askify is designed to bridge gaps in information retrieval and interaction quality by providing responses tailored to user needs. Key areas covered include system design, methodologies, and performance evaluation. Our findings highlight Askify's enhanced capabilities in accuracy, response relevance, and user accessibility, with applications in various domains and future improvement potential.

.

# 2 Keywords

ASKIFY, AI-powered chatbot, Generative AI, natural language processing, document-based Q/A, multilingual chatbot, OpenAI, FAISS, Google Translator, knowledge retrieval.

# 3 Introduction

Artificial intelligence (AI) has seen remarkable advancements in recent years, fundamentally changing the way people interact with digital platforms. Among these advancements, conversational AI has become a transformative technology in applications ranging from customer support and education to healthcare and knowledge management. As the adoption of AI-driven tools like chatbots continues to rise, so do user expectations for deeper, more meaningful, and contextually aware interactions. Traditional chatbots often fall short, primarily because they rely on either rule-based frameworks or static retrieval mechanisms that cannot adequately handle complex, context-sensitive queries. These limitations create a gap between user expectations and the capabilities of existing chatbot systems. To address these challenges, Askify—a Q&A chatbot powered by Generative AI and Retrieval-Augmented Generation (RAG)—was developed to deliver intelligent, fact-based, and contextually relevant responses.

The demand for enhanced conversational AI stems from the need to bridge the limitations of conventional chatbots. In most settings, traditional chatbots either use pre-set rules or depend solely on retrieval techniques to pull information from a database. While this approach may work for straightforward queries, it lacks the flexibility to address dynamic or multi-step questions effectively. Such systems often struggle to interpret queries that require a nuanced understanding of language and context, leading to incomplete or irrelevant responses. This shortcoming is particularly evident in scenarios that demand specificity, such as responding to detailed customer inquiries, answering complex academic questions, or providing step-by-step guidance based on specific documents or knowledge bases. Askify aims to address these limitations by incorporating Retrieval-Augmented Generation, a technique that combines information retrieval with generative language models, creating responses that are accurate, coherent, and grounded in real-world information.

At the core of Askify's architecture lies Generative AI, an advanced AI approach that enables machines to generate human-like text. This is achieved through the use of transformer-based models, like OpenAI's GPT (Generative Pre-trained Transformer), which are designed to capture intricate patterns in language, context, and user intent. Generative models like GPT are trained on massive datasets, allowing them to understand and replicate the structure, grammar, and nuances of human language. However, despite their

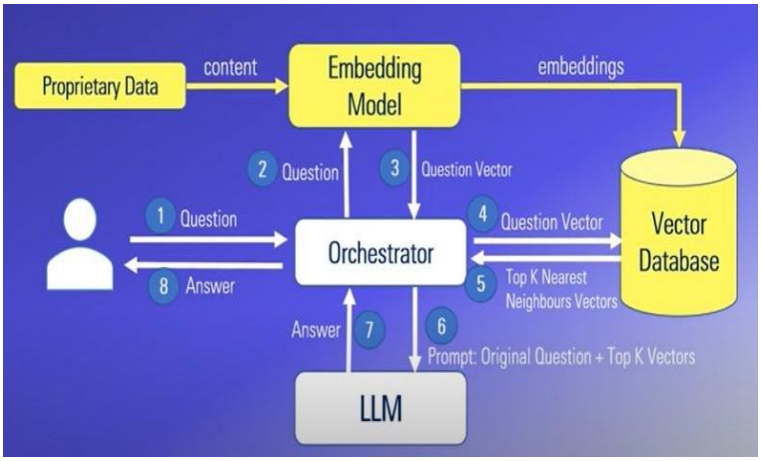


Figure 1: RAG Workflow

sophisticated capabilities, purely generative models have limitations. One major issue is the potential for ”hallucination,” where the model produces content that sounds coherent but is factually inaccurate or ungrounded in real information. This tendency arises because generative models rely solely on patterns learned from training data, rather than drawing from an up-to-date, factual knowledge base. To overcome this, Askify integrates Retrieval-Augmented Generation, which combines the creative strengths of generative models with the factual accuracy of retrieval mechanisms.

Retrieval-Augmented Generation (RAG) is a hybrid approach that enhances chatbot responses by incorporating a retrieval step before generating an answer. In this setup, the system first retrieves relevant information from a knowledge base or document repository, which is then used by the generative model to create a response that is both contextually relevant and factually accurate. The RAG approach significantly improves the quality of answers, especially for complex queries, by grounding responses in specific information that can be traced back to a source. For instance, when a user asks a detailed question that requires information from a specific document or web source, the retrieval mechanism pulls relevant sections of text, which are then integrated into the generative response. This method not only improves the reliability of answers but also makes the chatbot more transparent and trustworthy, as responses are based on retrievable data rather than solely on learned patterns.

The problem with current search engines and Q&A systems is that they

do not fully meet the needs of users who require accurate, context-sensitive information for complex queries. While search engines are effective at returning lists of documents, they require significant user effort to sift through results and locate relevant information. Traditional chatbots, on the other hand, often provide simplistic answers based on predefined scripts, lacking the ability to dynamically adjust responses based on user input. This is especially problematic in domains where queries may be multi-faceted, requiring a deeper understanding of context and the ability to reference multiple information sources. Askify was developed to overcome these limitations by combining the efficiency of retrieval systems with the nuanced language understanding of generative models, resulting in a chatbot that can understand and respond to complex queries in a meaningful way.

One of Askify's primary objectives is to create a system that can provide precise, contextually accurate answers by leveraging the strengths of both retrieval-based and generative AI models. The system's architecture is designed to handle a wide range of user queries by first retrieving relevant information and then using it as the basis for generating a coherent response. This process ensures that the chatbot's answers are not only accurate but also grounded in real-world data. In addition to improving answer quality, Askify also aims to address accessibility challenges by incorporating multilingual support and voice interaction capabilities. By supporting multiple languages through Google's Translator API, Askify allows users from diverse linguistic backgrounds to interact with the chatbot in their preferred language. Voice interaction, enabled through Speech Recognition and Google Text-to-Speech (GTTS), enhances accessibility further by allowing users to engage with Askify through spoken queries and receive spoken responses.

Askify's architecture incorporates several key components to achieve its goals. The frontend interface, built with Streamlit, provides an accessible platform for users to upload PDFs, enter URLs, ask questions, and select their preferred language for responses. The backend is powered by FAISS (Facebook AI Similarity Search), a library optimized for fast, efficient information retrieval. FAISS enables the system to create vector embeddings of text chunks from documents, which are stored in a vector database and can be searched for relevant matches. This retrieval capability is coupled with OpenAI's GPT, which generates responses based on the information retrieved. LangChain is used for query processing, enabling the chatbot to manage multi-step conversations by breaking down and organizing complex queries. Data extraction is achieved through PyPDF2 for PDF files and

BeautifulSoup for web pages, making Askify's knowledge base both comprehensive and adaptable.

Askify's innovative design extends beyond conventional chatbot systems, as it prioritizes both conversational fluency and factual accuracy. By utilizing RAG, Askify is able to answer complex questions with a depth that would be challenging for retrieval-only or generative-only models. The retrieval component grounds the chatbot's responses in reality, while the generative model provides the flexibility and language fluency needed for natural interactions. In practice, this means that Askify can handle follow-up questions, remember previous interactions within a conversation, and provide answers that consider the context of earlier queries. This functionality is critical for creating a seamless, coherent conversation flow that mimics human interaction.

Furthermore, Askify aims to set a new standard in user accessibility by offering multilingual and voice interaction features. The multilingual support provided through Google Translator API ensures that Askify can cater to users from various linguistic backgrounds, removing language barriers that typically limit the usability of similar platforms. The inclusion of voice interaction capabilities, through Speech Recognition and GTTS, allows users to engage with Askify hands-free, making it suitable for a range of scenarios, including those where screen-based interaction may not be possible or convenient. These features make Askify not only a more versatile tool but also a more inclusive one, meeting the needs of diverse user groups.

In summary, Askify represents a pioneering approach to conversational AI, utilizing advanced retrieval and generative techniques to provide high-quality, contextually relevant responses. By addressing the limitations of traditional Q&A systems, Askify holds significant potential to transform user interactions across various applications, from customer service and knowledge management to educational support. With its robust architecture, innovative use of RAG, and focus on accessibility, Askify offers a powerful and scalable solution for the next generation of AI-driven question-answering systems.

# 4 Literature Review

The rapid development of artificial intelligence, especially within natural language processing, has significantly advanced the capabilities of conversational agents and chatbots across various industries. Generative AI, in particular, has become a powerful tool for creating sophisticated conversational sys-

tems, especially with the advent of transformer architectures. Introduced by Vaswani et al. (2017), transformer models have transformed natural language processing by enabling machines to capture complex patterns and long-range dependencies within language. This innovation led to the development of models like GPT (Generative Pre-trained Transformer) by OpenAI, which leverages large-scale pre-training on vast datasets to understand and generate human-like text. Generative models such as GPT-2 and GPT-3 demonstrated a remarkable ability to produce coherent, contextually relevant text through unsupervised learning, making them highly effective for diverse language tasks like summarization, translation, and conversational responses. GPT-3, with its 175 billion parameters, exemplifies the advancements in language generation, as shown in Brown et al. (2020), producing text that closely mimics human language. This capability has ignited interest in applying GPT models to conversational AI, where the quality and fluidity of responses are critical.

Despite these advancements, purely generative models face inherent limitations, especially in terms of factual accuracy and relevance. Generative models, though powerful in understanding language patterns, often produce ”hallucinated” information, where the output may appear coherent but lacks factual basis. This issue is particularly problematic in conversational AI, where users expect accurate and reliable information. Maynez et al. (2020) documented the phenomenon of hallucination in generative models, highlighting cases where responses deviate from real-world data, thus affecting the reliability of AI-driven chatbots. These limitations have led researchers to explore hybrid models, such as retrieval-augmented generation, which integrate both retrieval mechanisms and generative capabilities to enhance response accuracy and reduce the likelihood of hallucination.

The concept of Retrieval-Augmented Generation (RAG) emerged as a method to address the limitations of standalone generative models by combining retrieval systems with language generation. Proposed by Lewis et al. (2020), the RAG framework involves a two-step process: first, relevant information is retrieved from a knowledge base, and then a generative model uses this context to produce an answer. This method has proven effective in generating both coherent and accurate responses, as the model’s output is grounded in factual data rather than solely learned language patterns. RAG-based systems, such as Facebook AI’s RAG model, have demonstrated significant potential in QA applications where responses must be precise and relevant to the user’s query. The retrieval component in RAG systems frequently

utilizes vector similarity search techniques with tools like FAISS (Facebook AI Similarity Search), which indexes document embeddings and enables efficient retrieval of relevant information. Developed by Johnson et al. (2017), FAISS optimizes search processes by creating an embedding space that facilitates fast similarity searches. FAISS has become a widely adopted tool for information retrieval in high-performance NLP applications due to its ability to handle large-scale datasets efficiently.

In a typical RAG system, both user queries and documents are encoded into vectors, and a similarity search is performed to identify documents that closely match the query. Once relevant information is retrieved, it is passed to the generative model—such as GPT-3—which uses it as a basis for crafting a response. This combination reduces hallucination and enhances factual accuracy, as demonstrated in studies by Karpukhin et al. (2020) and Guu et al. (2020), who showed that retrieval-augmented generation systems provide a robust solution for addressing the shortcomings of generative models. These studies found that RAG models are particularly valuable in high-stakes applications, where reliability and factual accuracy are paramount. The RAG approach also supports multi-turn conversations, where context from previous user interactions is essential to generate coherent and relevant responses. Research by Zhang et al. (2018) illustrated that incorporating retrieval mechanisms in chatbots enables them to retain conversational history, which is critical for delivering consistent responses across multiple interactions. Such functionality makes RAG systems ideal for applications in customer service, healthcare, and educational support, where users often engage in extended dialogues and require detailed information or clarifications. Askify, designed as a RAG-based system, leverages these advancements to ensure that its responses are both contextually grounded and factually accurate.

Evaluating the effectiveness of RAG-based conversational AI systems presents unique challenges, as traditional metrics like BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) are limited in their ability to assess the nuanced requirements of these systems. BLEU and ROUGE primarily measure the overlap between generated and reference responses, focusing on tasks like translation and summarization, where surface-level similarity is important. However, these metrics are insufficient for RAG-based Q&A systems, where the evaluation criteria extend beyond syntactic accuracy to include factual grounding, relevance, and responsiveness. To address these limitations, new evaluation frameworks have emerged to measure the quality of responses generated by

RAG systems. RAGAS (Retrieval-Augmented Generation Assessment Score) is one such framework, offering a comprehensive set of metrics tailored specifically to RAG models. Within RAGAS, key metrics include context precision, faithfulness, answer relevancy, and context recall, each addressing distinct aspects of response quality and effectiveness.

Context precision measures the degree to which retrieved documents match the intent of the user's query, ensuring that the retrieval process provides relevant context for the generative model. High context precision is critical in RAG systems, as it underpins the relevance and accuracy of the response. Faithfulness assesses whether the generated response accurately reflects the information in the retrieved context, preventing the model from deviating or introducing hallucinated content. Faithfulness is particularly important for RAG-based chatbots because it ensures that responses remain grounded in factual data. Answer relevancy evaluates whether the generated answer directly addresses the user's question, maintaining focus on the core intent of the query. This metric is essential for assessing the practical usefulness of the response, as irrelevant or tangential answers can hinder user satisfaction. Lastly, context recall measures the extent to which the generative model utilizes all relevant information from the retrieved context, ensuring completeness in the response. Together, these metrics provide a multi-dimensional view of a RAG system's performance, capturing the balance between accuracy, relevance, and comprehensiveness in chatbot responses.

In addition to improving accuracy and relevance, accessibility is a crucial consideration for modern conversational AI systems. Studies indicate that multilingual support is essential to making AI systems more inclusive and accessible to a global audience (Conneau et al., 2020). Multilingual support enables chatbots to interact with users in various languages, removing language barriers and enhancing usability. Translation APIs, such as Google Translate, allow language models to adapt to diverse linguistic preferences, making conversational AI accessible to non-English speakers. Furthermore, voice interaction capabilities, facilitated by tools like Speech Recognition and Google Text-to-Speech (GTTS), enhance accessibility by enabling users to communicate with chatbots through spoken language. Amodei et al. (2016) demonstrated the impact of voice-enabled AI on accessibility, showing how speech recognition and text-to-speech technologies can broaden the reach of AI systems, particularly for users who may prefer or rely on auditory interactions.

For Askify, integrating multilingual and voice interaction capabilities is essential for creating a versatile and user-friendly system that can cater to a diverse range of users. Multilingual support ensures that Askify is accessible to users across different linguistic backgrounds, while voice interaction provides an additional mode of communication, making the system suitable for various contexts, including hands-free and screen-free environments. By combining these accessibility features with RAG-based generation, Askify is positioned to offer a robust, inclusive, and intelligent conversational experience that meets the demands of a modern user base.

In summary, the integration of Generative AI with Retrieval-Augmented Generation marks a significant advancement in conversational AI. The literature on generative models, retrieval mechanisms, and hybrid RAG systems highlights the effectiveness of this approach for creating intelligent, contextually aware chatbots. Furthermore, new evaluation metrics tailored for RAG systems provide robust methods for assessing response quality, addressing factors that are often overlooked by traditional metrics. Accessibility features, including multilingual and voice support, further enhance the usability and inclusivity of such systems, making them adaptable to a wide range of applications. Askify's design is grounded in these advancements, combining retrieval-augmented generation with tailored evaluation and accessibility features to deliver a powerful, user-centered conversational AI solution.

# 5 Methodology

## 1. Framework and Libraries Used

**Streamlit:** Streamlit is selected for its simplicity in creating data-driven web applications with Python. It allows ASKIFY to present a user-friendly interface, enabling seamless interaction with the chatbot. Used to build the front-end, providing a clean layout where users can upload PDFs, enter URLs, select languages, and ask questions.

**LangChain:** LangChain is essential for managing ASKIFY's multi-step conversation flow and ensuring coherent interaction. By using LangChain's CharacterTextSplitter, text is divided into manageable chunks, which helps in processing long documents effectively without losing context.

**PyPDF2:** Allows extraction of text from PDF documents, making it accessible for querying within ASKIFY. PyPDF2 reads and parses PDF files,

enabling ASKIFY to retrieve text from uploaded documents for further processing in the conversational chain.

**BeautifulSoup:** Used for web scraping to pull text content from URLs. BeautifulSoup parses the HTML structure, extracting text content, which is then added to ASKIFY's conversational memory for a robust answer formation.

**Google Translator API:** This API broadens ASKIFY's accessibility by translating responses into the user's chosen language. The chatbot output is translated into a user-selected language (e.g., Spanish, French) using Google Translator, enhancing multi-lingual support.

**Speech Recognition:** Converts spoken language into text, allowing users to interact with ASKIFY through voice commands. Speech Recognition API processes audio input, converting spoken queries into text, which is then processed by the conversational model.

**gTTS (Google Text-to-Speech):** Converts text-based responses from ASKIFY into audible outputs, allowing users to hear the responses. gTTS generates an audio version of ASKIFY's answers, which can be played in the application, making ASKIFY more accessible.

## 2. Conversational Chain and RAG (Retrieval-Augmented Generation)

**RAG Overview:** RAG combines information retrieval with text generation, enhancing chatbot responses by integrating relevant document content into the generated answers.

**Implementation in ASKIFY:**

- **Retrieval Step:** When a user asks a question, ASKIFY retrieves relevant content from stored documents using embeddings and a vector database.
- **Generation Step:** The generative model (GPT-based) creates responses by synthesizing user input with retrieved information, ensuring responses are contextually accurate.
- **Response Formation:** The final answer combines the retrieved content and the model-generated response, delivering a comprehensive answer.

**Text Chunking:** The CharacterTextSplitter class from LangChain splits large texts into smaller chunks, making them easier to manage. The chunk size and overlap are carefully set to balance context retention with processing efficiency. ASKIFY uses this method for document handling, ensuring each chunk retains enough context for relevant answers.

**Vector Store with FAISS (Facebook AI Similarity Search):** FAISS stores and retrieves text chunks based on similarity, enabling quick, efficient searches. Text chunks from documents are embedded into vectors using OpenAIEmbeddings. These vectors are stored in FAISS, allowing ASKIFY to retrieve the most relevant chunks when responding to user questions.

## 3. Evaluation - RAGAS (Retrieval-Augmented Generation Assessment Score)

**RAGAS Purpose:** This metric assesses ASKIFY's response quality, focusing on relevance, faithfulness, and recall.

**Evaluation Metrics:**

- **Context Precision:** Measures how accurately the retrieved context matches the user query.
- **Faithfulness:** Assesses how well the response aligns with original document content, preventing errors or misleading information.
- **Answer Relevancy:** Checks if the response directly answers the user's question.
- **Context Recall:** Ensures the model considers all essential content in response generation, maximizing informativeness.

## 4. Workflow Steps in ASKIFY

**Document/URL Processing:** Users upload PDF files or input a URL. ASKIFY reads the PDF with PyPDF2 or scrapes the website using BeautifulSoup, extracting raw text content.

**Text Splitting and Vectorization:** The extracted text is split into chunks. Each chunk is vectorized and stored in FAISS, establishing a searchable database of document content.

**Question Processing and Response Generation:** User inputs (text or voice) are processed, and ASKIFY retrieves relevant chunks from the FAISS vector store. The generative model synthesizes a response, combining the user query with the retrieved text, ensuring accuracy and context relevance.

**Language Translation and Output:** The response is translated into the user's selected language using Google Translator API. The translated response is displayed in text and optionally converted to audio using gTTS.

# Results and Analysis

## Context Precision:

**Observation:** For all user inputs, the context precision score is consistently 1, indicating that the system accurately identifies and retrieves the relevant context for each query.
**Conclusion:** The chatbot is highly effective at pinpointing the correct context, suggesting that the retrieval mechanism (using FAISS and embeddings) is functioning well to provide contextually accurate information.

## Faithfulness:

**Observation:** Faithfulness scores vary, with some responses scoring 0.5 or 0.333, indicating partial alignment with the reference content. For example:

- The query "What is ARTIFICIAL INTELLIGENCE?" scores 0.5, suggesting some minor deviation or lack of completeness.
- "What is Supervised Machine Learning?" has a low score of 0.333, indicating potential inaccuracies or missing details.

**Conclusion:** Faithfulness could be improved in some cases. Ensuring that responses accurately reflect the referenced content is critical, as users rely on the chatbot for factual information. Addressing this may involve refining the RAG pipeline to improve response consistency.

### Answer Relevancy:

**Observation:** The answer relevancy scores are generally high, with values close to 1, demonstrating that responses are mostly relevant to the questions asked.
**Conclusion:** The chatbot effectively generates responses that are relevant to user queries. High answer relevancy indicates the model's understanding of user intent, which is a positive indicator of the quality of responses.

### Context Recall:

**Observation:** Most of the responses have a context recall of 1, meaning the responses contain all relevant information from the retrieved context. However, for a few queries (e.g., "What is DEEP LEARNING?"), context recall is less than 1, which might indicate missing contextual details.
**Conclusion:** While context recall is generally high, certain responses may lack a complete representation of the context. Addressing these cases can help improve the thoroughness of the answers, enhancing the informativeness of responses.

## 6 Demo & Real-World Use Cases

The following are key real-life use cases where Askify's capabilities can be effectively applied across various industries:

1. **Content Automation**: Generative AI in Askify enables automated content creation for applications such as marketing, writing, and coding. This automation saves time and enhances productivity for content creators and businesses.

2. **Multilingual Customer Support**: With integrated multilingual support, Askify can assist a global audience, making it ideal for businesses needing to provide customer service across different languages.

3. **Voice-Enabled Assistance**: The voice interaction feature allows Askify to be used in hands-free environments, such as healthcare or automotive industries, where quick, spoken responses are beneficial.

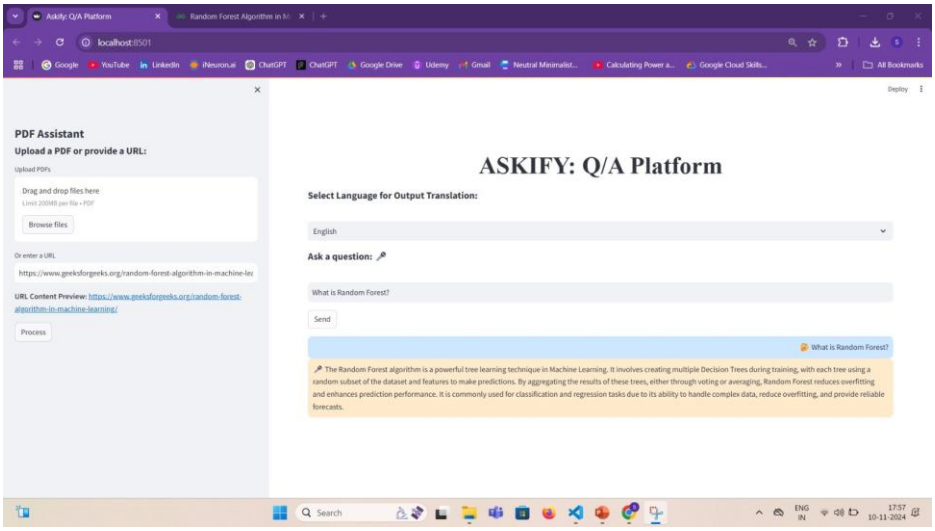


Figure 2: Demonstration

4. **Data-Driven Business Insights**: Askify's ability to efficiently retrieve information from large document repositories supports businesses in making fast, data-driven decisions by providing relevant answers promptly.

These use cases demonstrate Askify's adaptability and impact across diverse fields, showing its potential as a valuable tool in real-world applications.

# 7 Future Scope

The potential advancements in the chatbot system can significantly enhance its usability, accessibility, and performance. Below are the key areas identified for future improvement:

1. **Multilingual and Voice Support**: Integrating advanced multilingual capabilities along with voice interaction features will allow users to communicate in various languages and interact hands-free. This will broaden accessibility and improve user experience for a diverse audience.

2. **Enhanced Conversational Flow**: Developing a more sophisticated conversation management framework will enable the chatbot to han-

dle complex, multi-turn dialogues. This enhancement will provide a smoother user experience, offering more contextually relevant responses, especially in extended conversations.

3. **Real-time Data Integration**: By incorporating real-time data retrieval capabilities, the chatbot can ensure responses remain current and accurate, particularly for dynamic or time-sensitive topics. This feature will increase the chatbot's relevance and usefulness in scenarios that demand up-to-date information.

4. **Advanced Evaluation Metrics**: Implementing detailed evaluation metrics such as the Retrieval-Augmented Generation Assessment Score (RAGAS) will allow for continuous assessment and improvement in relevance, accuracy, and contextual understanding. This ensures that the chatbot consistently meets user expectations and maintains high satisfaction levels.

These advancements will collectively enhance the chatbot's robustness, reliability, and applicability in a variety of real-world scenarios, making it a more versatile and valuable tool for users.

# 8 Conclusion

The **Askify Q&A Platform** represents a significant advancement in leveraging AI for interactive, multi-lingual, and efficient question-answering solutions. By integrating cutting-edge technologies such as *Streamlit* for user interface design, *LangChain* for conversational flow, and *FAISS* for optimized data retrieval, Askify offers a robust framework capable of handling complex queries and providing accurate, contextually relevant responses. The adoption of Generative AI models, coupled with tools for language translation and speech processing, ensures that Askify is accessible to a broad audience, breaking down language barriers and enhancing user engagement through voice and text.

Moreover, the integration of Retrieval-Augmented Generation (RAG) and assessment techniques like RAGAS enables continuous improvement in response quality, making Askify a scalable and adaptive platform suited for real-world applications. As AI technology continues to evolve, Askify has the potential to expand its capabilities further, exploring applications across industries such as customer support, education, and corporate training. With ongoing development and refinement, Askify stands to become an indispensable tool in modern digital interactions, embodying the promise of AI-driven innovation in information accessibility and user engagement.

# 9 Appendix

The link to our Python codes:

https://drive.google.com/drive/folders/1lB8scnBLnyzwvy7KQbpcxv0i_sEgTJ9T

# References


[1] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[2] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*, 2016.

[3] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[4] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proc. of NAACL*, 2016.

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122v2*, 2017.

[6] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] Zhongqiang Huang and Mary Harper. Self-training PCFG grammars with latent annotations across languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 832–841. ACL, August 2009.