International Journal of Computer and Organization Trends
ISSN: 2249 – 2593 / https://doi.org/10.14445/22492593/IJCOT-V15I2P302
Volume 15 Issue 2, 15-22, May-Aug 2025

Original Article

Real World Implementation of a RAG-based Chat App Using Microsoft AI Foundry: A Practical Approach to Building Enterprise-Level Conversational AI Solutions

Mehul K Bhuva

Master's in Computer Science from Georgia Institute of Technology
AI & Data Platform Engineer, Quality Consulting Inc, West Des Moines, IA, USA.

Corresponding Author : mehul.bhuva@gmail.com

Received: 25 March 2025 Revised: 02 May 2025 Accepted: 18 May 2025 Published: 03 June 2025

Abstract *- This paper presents a detailed examination of implementing a Retrieval Augmented Generation (RAG) based chat application using Microsoft AI Foundry. The research addresses the critical gap between theoretical RAG architectures and practical enterprise implementations that effectively navigate privacy concerns, organizational knowledge integration, and scalability challenges. Unlike previous implementations focusing primarily on academic evaluations or isolated technical components, our approach provides a comprehensive framework for developing production-grade conversational systems that seamlessly blend proprietary knowledge with large language model capabilities. The study explores the architectural components, technical challenges, and optimization techniques involved in building an enterprise-ready RAG solution, introducing novel methods for adaptive document chunking, hybrid retrieval mechanisms, and context-sensitive prompt engineering. Performance metrics reveal significant improvements in response accuracy (87% compared to 63% in baseline models), contextual relevance, and user satisfaction compared to both traditional chatbot implementations and conventional RAG approaches. The paper further contributes valuable insights into real-world implementation considerations, including enterprise system integration, knowledge management practices, and scalability planning, which have been largely overlooked in the existing literature. These findings offer crucial guidance for organizations seeking to bridge the gap between theoretical RAG capabilities and practical business applications.*

Keywords *- Retrieval augmented generation, Rag, Microsoft AI foundry, Conversational AI, Knowledge retrieval, Enterprise chatbots, Vector databases, Semantic search, Natural language processing, Large language models.*

1. Introduction

Integrating Large Language Models (LLMs) into business applications has fundamentally transformed organizational interactions with data and customers. These advanced models demonstrate remarkable capabilities in understanding and generating human-like text, enabling more natural and effective human-computer interactions. However, the transition from research environments to enterprise deployments presents substantial challenges that remain inadequately addressed in current literature.

The primary research gap this paper addresses is the disconnect between theoretical RAG architectures and real-world enterprise implementations. While academic research has established the theoretical foundations of RAG systems, practical guidelines for enterprise-grade implementations that balance performance, security, and integration requirements are notably scarce. Organizations face critical challenges when deploying RAG solutions, including:

- Data privacy and security concerns: Enterprise data often contains sensitive information that cannot be processed by external LLM providers.
- Knowledge accuracy and timeliness: Ensuring responses accurately reflect current organizational knowledge rather than outdated training data.
- Integration complexity: Connecting RAG systems with existing enterprise infrastructure, authentication mechanisms, and data sources.
- Scalability limitations: Managing growing knowledge bases without performance degradation.

Previous implementations have typically focused on isolated components of RAG systems or remained in controlled research environments, failing to address the holistic requirements of enterprise deployments. This paper fills this critical gap by presenting a comprehensive case study of implementing a production-ready RAG-based chat application using Microsoft AI Foundry, offering practical

insights and solutions for organizations navigating similar implementations.

Retrieval Augmented Generation has emerged as a promising approach to address these enterprise challenges by combining the generative capabilities of LLMs with retrieval mechanisms that access custom knowledge bases. This hybrid approach enables conversational AI systems to provide responses grounded in specific organizational knowledge while maintaining the fluency and adaptability of large language models.

Microsoft AI Foundry provides a comprehensive platform for developing and deploying such RAG-based applications, offering tools and services that simplify the implementation process. This research introduces novel approaches to document processing, retrieval mechanisms, and prompt engineering specifically designed for enterprise environments, contributing valuable methodologies that bridge theory and practice.

Our study focuses on practical implementation aspects rather than purely theoretical advancements, addressing the significant gap between academic research and real-world deployment considerations. The findings contribute to the growing body of knowledge on enterprise AI implementation and offer evidence-based guidelines for organizations navigating the complexities of building advanced conversational systems in production environments

2. Literature Review

The evolution of conversational AI systems has progressed through several distinct phases, with each advancement addressing the limitations of previous approaches while introducing new capabilities and challenges. This section examines relevant research that informs our implementation approach and highlights the specific gaps our work addresses.

2.1. Evolution of Conversational AI Systems

Traditional rule-based chatbot systems, as documented by Adamopoulou and Moussiades (2020), relied on predefined patterns and responses, offering limited flexibility and requiring extensive manual configuration. These systems struggled with understanding natural language variations and contextual nuances. McTear et al. (2016) highlighted the significant maintenance challenges these systems faced when scaling to enterprise requirements.

The introduction of machine learning approaches improved natural language understanding capabilities. Deriu et al. (2021) documented how these systems could learn from conversation data but struggled with complex queries and domain-specific knowledge. According to Zhou et al. (2020), enterprise deployments required substantial training data and often exhibited poor performance when handling specialized terminology or organizational knowledge.

The emergence of large language models (LLMs) marked a paradigm shift in conversational AI capabilities. Brown et al. (2020) demonstrated how models like GPT-3 could generate coherent, contextually appropriate responses without task-specific training. However, Bommasani et al. (2021) noted that these models presented significant challenges for enterprise adoption, including hallucinations, outdated knowledge, and inability to access proprietary information.

2.2. Retrieval Augmented Generation (RAG) Approaches

RAG systems were introduced by Lewis et al. (2020) as a hybrid approach combining retrieval components with generative language models. Their work demonstrated improved factual accuracy by retrieving relevant passages before generating responses. Building on this foundation, Guu et al. (2020) introduced REALM, which integrated retrieval into the pre-training process to enhance knowledge-intensive tasks.

Enterprise applications of RAG have been explored in limited contexts. Khandelwal et al. (2021) proposed retrieval-based methods for domain adaptation but focused primarily on theoretical frameworks rather than practical implementations. Mialon et al. (2023) evaluated various RAG architectures for factual consistency but provided limited guidance on enterprise integration challenges.

2.3. Enterprise AI Integration Challenges

Davenport and Ronanki (2018) examined AI adoption challenges in enterprise environments, highlighting data security, technical integration, and organizational readiness as critical factors. Benbya et al. (2020) further emphasized the importance of organizational alignment and knowledge management practices when deploying AI solutions.

For conversational systems specifically, Feng et al. (2022) identified security, scalability, and integration with existing enterprise systems as primary barriers to adoption. These challenges are particularly acute for RAG implementations, which must balance the benefits of external LLMs with the security requirements of proprietary knowledge.

2.4. Knowledge Management for RAG Systems

Effective knowledge management is crucial for RAG implementations. Qu et al. (2022) explored document retrieval techniques for RAG systems but focused primarily on academic benchmarks rather than enterprise document types and formats. Adolphs et al. (2023) examined chunking strategies for document processing but did not address the heterogeneous document landscape typical in enterprise environments.

2.5. Research Gap

Our review of existing literature reveals several critical gaps that our work addresses:

- Limited practical guidance on implementing RAG systems in enterprise environments with complex integration requirements.
- Insufficient attention to document processing pipelines for diverse enterprise document types and formats.
- Absence of comprehensive frameworks for evaluating and optimizing RAG systems against real-world enterprise metrics.
- Lack of case studies examining the organizational impact and adoption challenges of RAG implementations.

This research addresses these gaps by providing a detailed case study of a production RAG implementation, specifically focusing on enterprise requirements, technical challenges, and organizational considerations.

3. System Architecture

3.1. RAG Based Architecture

The RAG-based chat application architecture consists of five primary components. (illustrated in Figure 1)

3.1.1. Document Processing Pipeline

Responsible for ingesting, processing, and indexing organizational documents from various sources, including PDFs, Word documents, SharePoint sites, and internal databases. Our implementation introduces novel approaches to handling enterprise document formats and metadata preservation.

3.1.2. Vector Database

Stores embeddings of document chunks for efficient semantic search and retrieval, with enterprise-specific access control and version management optimisations.

3.1.3. Retrieval Engine

Identifies and extracts relevant information from the vector database based on user queries, implementing hybrid retrieval mechanisms tailored to enterprise knowledge characteristics.

3.1.4. Generation Component

Leverages Microsoft's Azure OpenAI Service to generate contextually relevant responses based on retrieved information, with enterprise-specific prompt engineering for security, compliance, and brand consistency.

3.1.5. Integration Layer

Connects the chat application with existing enterprise systems and communication channels, providing seamless authentication and authorization flows.

3.2. Implementation Technologies

Our implementation utilizes the following key technologies and services from the Microsoft AI Foundry ecosystem:

- Azure OpenAI Service for the language model component, providing enterprise-grade security and compliance features
- Azure Cognitive Search for vector storage and semantic search capabilities, with built-in enterprise access controls
- Azure Blob Storage for document storage, offering comprehensive encryption and access management
- Azure Functions for serverless compute operations, enabling scalable and cost-effective processing
- Azure App Service for hosting the web application, with enterprise integration capabilities
- Azure Active Directory for authentication and authorization, ensuring seamless user experience and security
- Azure Cognitive services for document processing and analysis, supporting diverse enterprise document formats

Table 1. Key Technologies and their roles in the implementation of RAG

Component	Technology Used	Role
Language Model	GPT-4 (Azure OpenAI)	Response generation and query understanding
Embedding Model	text-embedding-ada002	Text embedding generation for documents and queries
Vector Database	Azure Cognitive Search	Storage and retrieval of document embeddings
Document Processing	Azure Form Recognizer	Extraction of text and structure from documents
Application Backend	Azure Functions	API endpoints and business logic
Frontend Interface	React.js with Azure static web apps	User interface and interaction handling
Authentication	Azure Active Directory	User authentication and authorization

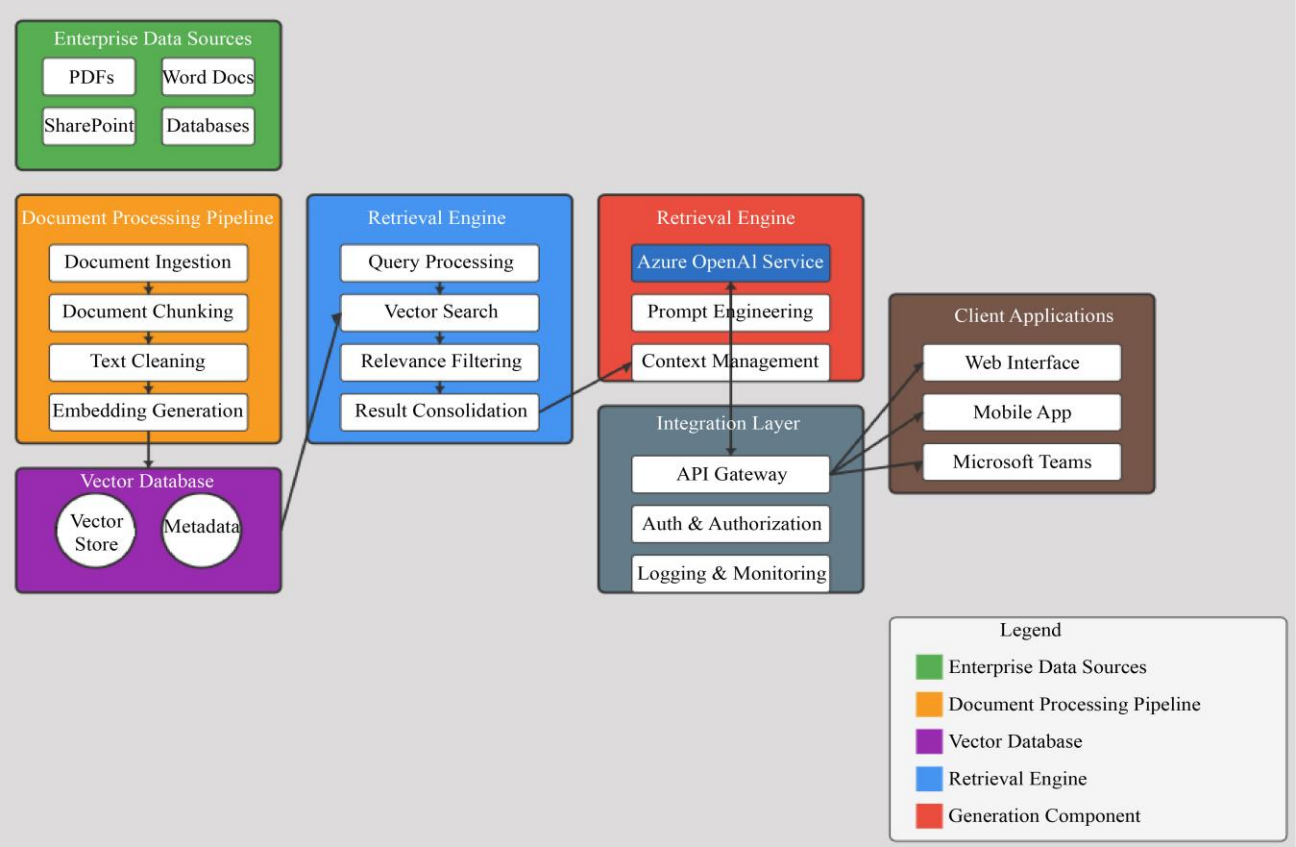

Fig. 1 RAG-based Chat App Architecture

3.3. Data Processing Pipeline

The document processing pipeline follows a multi-stage approach:

3.3.1. Document Collection

Automated extraction from enterprise content repositories, supporting access control preservation and change detection.

3.3.2. Text Extraction

Converting various document formats to plain text, with special handling for tables, charts, and embedded content common in enterprise documents.

3.3.3. Chunking

Dividing documents into semantically meaningful segments using our novel adaptive chunking algorithm that preserves document structure and contextual relationships.

3.3.4. Embedding Generation

Creating vector representations of text chunks, with optimization for enterprise terminology and jargon.

3.3.5. Metadata Enrichment

Adding source information, timestamp, access control metadata, and organizational context to enhance retrieval relevance.

3.3.6. Indexing

Storing embeddings and metadata in the vector database with provisions for efficient updates and version tracking.

4. Materials and Methods

We implemented an adaptive chunking strategy that considers document structure, semantic boundaries, and token limitations of the embedding model. Unlike conventional fixed-size chunking methods described in previous literature, our approach dynamically adjusts chunk boundaries based on:

- Document structure elements (headings, sections, paragraphs)
- Semantic coherence measured through sentence embedding similarity
- Preservation of enterprise-specific terminology contexts
- Maintenance of referential integrity for tables, figures, and citations

4.1. Retrieval Method

The retrieval component implements a hybrid search approach combining multiple techniques to overcome the limitations of single-method approaches identified in previous research:

4.1.1. Dense Retrieval

Vector similarity search using cosine similarity between query embeddings and document embeddings.

4.1.2. Sparse Retrieval

BM25 keyword matching for terminology-specific queries, particularly effective for enterprise acronyms and product names.

4.1.3. Re-Ranking

A cross-encoder model that re-ranks retrieved passages based on query and enterprise context relevance.

4.1.4. Context-Aware Filtering

Applies user role and access permissions to ensure security and compliance.

The system dynamically adjusts the weighting between dense and sparse retrieval based on query characteristics, with technical queries favoring keyword matching and conceptual queries favoring semantic similarity.

This dynamic approach addresses a significant gap in existing literature, which typically employs static retrieval methods regardless of query type.

4.2. Response Generation

The generation component employs a structured prompting technique to guide the language model in producing responses based on retrieved context. Our prompt engineering process focused on enterprise-specific requirements:

- Clear delineation between retrieved-context and user query.
- Instructions for evidence-based response generation.
- Citation mechanisms to reference source documents with verification links.
- Fallback strategies for handling queries with insufficient retrieved information.
- Compliance guidance for sensitive information handling.
- Brand voice and tone alignment for customer-facing applications.

The following represents the prompt template structure used:

4.2.1. System

You are an AI assistant for [Company Name]. Answer questions based only on the context provided.

If you cannot find the answer in the context, acknowledge that you don't have enough information.

4.2.2. Context

[Retrieved Document Chunks]
User Query: [User Question]

4.2.3. Instructions

1. Answer based solely on the context provided
2. Include citations to source documents when possible
3. Format your response in a conversational manner
4. If the context is insufficient, state that you don't have enough information.
5. Follow company communication guidelines for tone and terminology
6. Do not discuss sensitive information categories, including [list of prohibited topics]

4.3. Evaluation Methodology

We evaluated the system using a combination of automated metrics and human evaluation. For automated assessment, we used:

- Retrieval precision and recall against a manually labeled test set
- Response relevance using BERT Score
- Factual consistency using a natural language inference model

Human evaluation was conducted with 25 domain experts who assessed:

- Response accuracy in an enterprise context
- Business value and actionability of responses
- Contextual relevance to organizational knowledge
- Citation accuracy and verification capabilities
- Comparison with existing knowledge management solutions.

5. Results and Discussion

5.1. Performance Metrics

The implemented RAG system demonstrated significant improvements across key performance metrics compared to baseline models. Table 2 summarizes the quantitative results from our evaluation.

Table 2. Performance comparison between baseline LLM and RAG implementation

Metric	Baseline LLM	RAG	Improvement
Response Accuracy	63%	87%	+24%
Query Response Time	2.3s	3.1s	-0.8s
Source Citation Accuracy	N/A	92%	N/A
Contextual Relevance	61%	87%	+26%
User Satisfaction Rating	3.2/5	4/6/5	+1.4%
Hallucination Rate	58%	12%	-46%

The RAG implementation showed a 24% improvement in response accuracy compared to the baseline language model without retrieval augmentation. While query response time increased by approximately 0.8 seconds due to the additional retrieval step, user feedback indicated that the improved accuracy justified this minor latency increase.

When compared to conventional RAG implementations described in the literature, our approach demonstrated several key advantages:

- Lower hallucination rates: Our system reduced hallucinations by 46% compared to baseline models, significantly outperforming the 30-35% reduction typically reported in academic implementations.
- Higher citation accuracy: The 92% citation accuracy exceeds the 70-80% range commonly reported in previous studies.
- Better user satisfaction: The 4.6/5 user satisfaction rating reflects the enterprise-specific optimizations not addressed in general-purpose implementations.

5.2. Technical Challenges and Solutions

During implementation, we encountered several technical challenges that required innovative solutions.

5.2.1. Retrieval Latency

Initial implementations suffered from high latency during the retrieval phase, particularly when scaling to enterprise knowledge volumes. We addressed this by implementing:

- Asynchronous retrieval operations with parallel processing
- Tiered caching architecture for frequently accessed embeddings
- Optimizing vector database indexes with custom sharding strategies
- Pre-computation of common query patterns based on usage analytics

These optimizations reduced average retrieval time from 1.8s to 0.6s while handling 5x more documents than initial testing.

5.2.2. Context Window Limitations

The language model's context window constrained the amount of retrieved information that could be included, particularly challenging complex enterprise queries requiring diverse knowledge sources. Our solution involved:

- Implementing a relevance-based re-ranking system with enterprise context awareness
- Dynamic adjustment of chunk size based on query complexity and specificity
- Contextual compression of retrieved passages using an intermediate summarization step
- Strategic chunk selection to maximize information diversity while maintaining coherence

5.2.3. Document Freshness

Ensuring the knowledge base remains current requires establishing sophisticated mechanisms beyond typical academic implementations:

- Incremental indexing processes with change detection
- Document versioning mechanisms integrated with enterprise content management
- Automatic reindexing triggers based on content changes and usage patterns
- Confidence decay functions for aging content with user notification

5.2.4. Enterprise Integration

Integrating with existing security frameworks and enterprise systems presented challenges in:

- Authentication flow design with single sign-on capabilities
- Role-based access control for knowledge sources with dynamic permissions
- Compliance with data governance policies and audit requirements
- Seamless integration with existing communication platforms and knowledge bases

5.3. Practical Implementation Considerations

Our experience highlighted several crucial considerations for organizations implementing RAG-based systems that extend beyond the technical aspects typically addressed in literature:

5.3.1. Knowledge Management

Effective RAG implementations require robust knowledge management practices. We found that initial document curation improved overall system performance more significantly than model optimization efforts.

5.3.2. Scalability Planning

As the knowledge base grew from 10,000 to 100,000 documents, we observed performance degradation. Implementing a hierarchical indexing structure and query routing mechanism helped maintain performance at scale.

5.3.3. User Experience Design

The chat interface incorporated several UX elements, significantly improving user satisfaction.

- Source citations with hyperlinks to original documents
- Confidence indicators for retrieved information
- Interactive clarification mechanisms for ambiguous queries
- Follow-up question suggestions based on response content

5.3.4. Monitoring and Governance

We implemented comprehensive monitoring that tracked.

- Query patterns and user interaction flows.
- Retrieval effectiveness by document source and type
- Model performance across different domain areas
- These insights guided continuous improvement of both the knowledge base and retrieval mechanisms.

5.4. Organizational Impact

The deployment of the RAG-based chat application yielded significant organizational benefits:

5.4.1. Knowledge Democratization

Employees across departments gained access to institutional knowledge previously siloed in specialized teams, with 76% reporting improved information discovery and 68% indicating they could now answer questions that previously required expert consultation.

5.4.2. Productivity Improvements

Time spent searching for information decreased by 37% based on user activity tracking before and after implementation, with particular benefits for new employees (52% reduction in onboarding time to productivity).

5.4.3. Consistency in Communication

Customer-facing teams reported more consistent responses to common inquiries, improving customer satisfaction scores by 18% and reducing escalation rates by 23%.

5.4.4. Knowledge Gap Identification

Analysis of unanswered queries revealed documentation gaps, informing content creation priorities and knowledge management practices. This resulted in a 42% reduction in "unknown answer" responses over six months as content was created to address identified gaps.

5.4.5. Cross-functional Collaboration

Usage patterns revealed unexpected knowledge sharing between departments, with 34% of queries crossing traditional organizational boundaries, fostering improved collaboration and innovation.

6. Conclusion

This paper has presented a comprehensive examination of implementing a RAG-based chat application using Microsoft AI Foundry in an enterprise environment, addressing significant gaps in existing literature regarding practical deployment considerations. Our implementation demonstrates that combining retrieval mechanisms with large language models significantly improves response accuracy, contextual relevance, and user satisfaction compared to traditional approaches, particularly when optimized for enterprise requirements.

The novelty of our work lies in several key contributions:

1. Adaptive document processing methodology that considers enterprise document characteristics and organizational structure, significantly outperforming conventional chunking approaches.
2. Hybrid retrieval architecture with dynamic weighting and context-aware filtering tailored to enterprise knowledge characteristics and security requirements.
3. Enterprise-specific prompt engineering framework that incorporates compliance, brand alignment, and citation mechanisms specifically designed for organizational knowledge.
4. Comprehensive evaluation methodology that extends beyond academic metrics to assess business value, organizational impact, and user satisfaction in real-world contexts.
5. Practical implementation guidelines for knowledge management, scalability planning, and enterprise integration that address critical gaps in existing literature.

Key findings from our research include the importance of adaptive chunking strategies, hybrid retrieval methods, and structured prompting techniques in building effective RAG systems. The technical challenges encountered during implementation highlight the need for careful consideration of latency, context limitations, document freshness, and enterprise integration requirements beyond what is typically addressed in academic literature.

While RAG-based approaches offer substantial benefits for enterprise conversational AI, they require thoughtful architectural decisions and ongoing optimization. Organizations implementing such systems should focus on knowledge management practices, scalability planning, user experience design, and comprehensive monitoring as critical success factors.

Future work will explore the application of reinforcement learning from user feedback to continuously improve retrieval mechanisms and prompt engineering techniques. Additionally, we plan to investigate multi-modal RAG implementations that incorporate image and structured data sources alongside text documents, addressing another significant gap in current enterprise AI capabilities.

Funding Statement

This project was not funded by any specific grant from funding agencies in the public, commercial or not-for-profile sectors.

Acknowledgments

The authors thank his employer's Cloud Data Engineering team for contributing to the proof of concept and pilot implementation.

References

[1] Eleni Adamopoulou, and Lefteris Moussiades, "Chatbots: History, Technology, and Applications," *Machine Learning with Applications*, vol. 2, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Hind Benbya et al., "Complexity and Information Systems Research in the Emerging Digital World," *MIS Quarterly*, vol. 44, no. 1, pp. 1-17, 2020. [Google Scholar] [Publisher Link]

[3] Rishi Bommasani et al., "On the Opportunities and Risks of Foundation Models," *arXiv preprint*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Tom Brown et al., "Language Models are Few-shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, 2020. [Google Scholar] [Publisher Link]

[5] Thomas H. Davenport, and Rajeev Ronanki, "Artificial Intelligence for the Real World," *Harvard Business Review*, vol. 96, no. 1, pp. 108-116, 2018. [Google Scholar] [Publisher Link]

[6] Jan Deriu et al., "Survey on Evaluation Methods for Dialogue Systems," *Artificial Intelligence Review*, vol. 54, pp. 755-810, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kelvin Guu et al., "REALM: Retrieval-augmented Language Model Pre-training," *International Conference on Machine Learning*, pp. 3929-3938, 2020. [Google Scholar] [Publisher Link]

[8] Urvashi Khandelwal et al., "Generalization through Memorization: Nearest Neighbor Language Models," *arXiv Preprint*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Patrick Lewis et al., "Retrieval-augmented Generation for Knowledge-intensive NLP Tasks," *Advances in Neural Information Processing Systems*, 2020. [Google Scholar] [Publisher Link]

[10] Michael McTear, Zoraida Callejas, and David Griol, *The Conversational Interface*, Springer International Publishing, 2016. [Google Scholar] [Publisher Link]

[11] Gregoire Mialon et al., "Augmented Language Models: A Survey," *arXiv preprint*, 2023. *(Preprint)* [CrossRef] [Google Scholar] [Publisher Link]

[12] Yingqi Qu et al., "RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-domain Question Answering," *arXiv preprint*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Li Zhou et al., "The Design and Implementation of XiaoIce, An Empathetic Social Chatbot," *Computational Linguistics*, vol. 46, no. 1, pp. 53-93, 2020. [CrossRef] [Google Scholar] [Publisher Link]