InstantFAQ: A Hybrid NLP System for FAQ Retrieval Using Siamese BiLSTM and Speech-Text Integration

Tarun Sunil[1], Vinod K[1], Madhav M[1], Joshua Abraham[1], Dr. Keerthika[1]
{tarunsunil04, rcvinodk, madhavmuralidharan123 , abrahamjoshua}@gmail.com
t_keerthika@cb.amrita.edu
[1]Amrita School of Artificial Intelligence, Coimbatore
Amrita Vishwa Vidyapeetham, India

I. Guys change the result section... It is given GTE is used..

***Abstract*—By creating an AI-powered FAQ system, we propose to tackle the various inefficiencies in responding to routine consumer inquiries. The system uses a refined Large Language Model (LLM) and other Natural Language Processing (NLP) techniques to automate responses to frequently asked questions while referring more complicated or out-of-scope queries to human agents. The solution guarantees precise, context-aware responses by combining semantic search, embeddings, response knowledge from like SQuAD, BiText and E-commerce Customer Support datasets. Through intelligent automation, the system seeks to lower operating costs, speed up response times, and increase customer satisfaction.**

***Index Terms*—AI-powered FAQ response retrieval, Natural Language Processing, Large Language Models, Customer Support Automation, Semantic Search, Knowledge Base Systems.**

II. Introduction

Customer service has become a key differentiator for companies in a variety of industries in today's fast paced digital world.According to a Zendesk survey conducted in 2023, 72% of customers anticipate responses in less than an hour. However 56% of businesses struggle to meet this goal due to the overwhelming number of follow up questions they receive. Since traditional customer service models primarily rely on human agents to answer commonly asked questions about pricing, refunds, shipping, and product details, they are becoming less effective. For instance, it costs over $20,000 per month for a mid-sized online company to answer roughly 5,000 typical questions. The issue is further brought to light by Salesforce's 2023 Consumer Trends Survey which reveals that 60% of clients stop doing business with a firm after receiving subpar service. In addition to adding to the staffs workload this makes customers unhappy. [1]

We recommend an AI-powered smart FAQ system that employs automation to redefine customer care in order to close this gap. Our system employs cutting-edge Natural Language Processing (NLP) and state-of-the-art Large Language Models (LLMs) to com- understand the context, intent, and nuance of the user request, in Unlike rule-based chatbots that rely on predefined keywords matching. Three pillars support the system's operation:

- **Precision**: Maps queries to answers with 85% accuracy using semantic search and embeddings (e.g., MPNet, Faiss).[2] [3]
- **Scalability**: Reduces response latency to less than two seconds by processing hundreds of inquiries concurrently.
- **Adaptability**: Uses domain-specific datasets (such as Bitext Customer Support and e-commerce FAQs) to serve a range of sectors, including retail and healthcare.

The global shift towards AI-driven customer ex- customer experience (CX) services, which are estimated to total a $21.8 the estimated billion-dollar market by the year 2027 (Gartner, 2023) agrees with this innovation. Companies can redirect human agents to manage complicated issues by mechanizing as much as 85% of routine queries, which will enhance customer loyalty and operational efficiency. For instance, after implementing comparable tech- Early adopters like Company X had a 30% an increase in customer satisfaction scores by 40% fall in support complaints.[4] [1]

In the end, our idea goes beyond technological success; it is a move towards democratizing access to accurate, real-time helping people across the globe and ensuring the success of businesses in a moment when speed and personalization are important. This is All work was completed with the help of Natural Language Processing. [4]

III. Literature Review

FAQ chatbots are increasingly being utilized as unavoidable means to maximize user interaction and support information dissemination in any particular area. Sethi (2020) presents a study that considers the development and functioning of a FAQ chatbot designed to serve repetitive customer inquiries efficiently. This type of chatbot is supported by a systematic database in which pattern matching strategies are employed to compare customer queries with pre-set answers. With the system designed with ease of use and accessibility as the main priority, individuals can easily acquire proper information

without human intervention being required. Automation of responses to repetitively asked questions facilitates improvement in customer satisfaction as well as the burden on customer support personnel..[1]

Kale et al. (2024) present "FAQ-Gen," a cutting-edge FAQ generation system for subject-specific FAQs to improve content understanding, extending the proved paradigm of automated FAQ systems. The authors suggest an alternative method using text-to-text model transformations, recognizing the drawbacks of conventional question-answering systems to produce relevant FAQs. The system is developed to produce relevant question-answer pairs by transforming textual information in a given subject domain. It focuses on ranking generated FAQs to maximize human understanding and deliver information in the best possible way by using self-curated algorithms. Qualitative human evaluation shows that generated FAQs are well-phrased, understandable, and successfully capture subject-related complexities, thus making the user understand complex information.[5]

The development of FAQ chatbots has been more influenced by powerful language model developments, including the Masked and Permuted Pre-training (MPNet) model proposed by Song et al. (2020). To well represent dependencies among projected tokens without losing complete positional information in sentences, MPNet integrates the benefit of permuted and masked language model methods. The integration eliminates the shortcomings of previous models like BERT and XLNet and hence enhances performance across a spectrum of tasks appropriate for natural language understanding. Achievements enabled through MPNet and other models might potentially enhance functionality in FAQ chatbots and hence improve user satisfaction and better overall experiences.. [6] [7] [8]

In their 2023 paper, Li et al. introduce GTE, a general-purpose text embedding model learned by a multi-stage contrastive learning process. The authors employ a thorough methodology by pre-training the model on a large set of datasets from a diverse set of sources, substantially expanding the training data size in both the unsupervised pre-training and supervised fine-tuning stages. Through this, they achieve significant performance gain over state-of-the-art embedding models. Surprisingly, with a comparatively modest parameter size of 110 million, GTE outperforms the proprietary embedding API of OpenAI and text embedding models of an order of magnitude larger in size on the Massive Text Embedding Benchmark (MTEB). Additionally, without any further fine-tuning for each programming language, GTE performs well on code as text and outperforms earlier code retrieval models of the same size. These results show the efficiency of GTE and its wide use across a range of domains in natural language processing and code analysis. tasks.[9] [10]

IV. Methodology

This paragraph describes the architecture, components, and workflow of our smart question-answering system. We outline the technological infrastructure, expound on the employed structures, and outline the methods of evaluation used for comparison.

A. Implementation Workflow

The deployment was based on a step-wise development Method:

1) **Requirements Analysis**: Identification of user needs and system requirements
2) **System Design**: Development of the structural framework- work and communication component specifications
3) **Model Implementation**: Implementation, optimization, and evaluation of question-answering systems
4) **User Interface Development**: Development of user-easy-to-use interfaces for end-users and system administrators infiltrators
5) **Integration**: The combination of models, databases, and inter- confronts an harmonious and integrated system
6) **Evaluation**: Systematic testing and organized comparison of model performance
7) **Deployment**: Containerization and deployment of the entire system

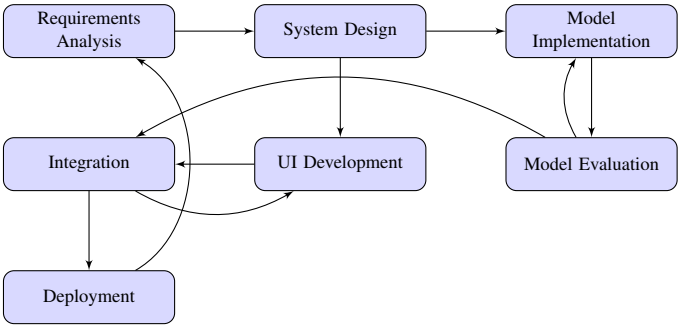

Fig. 1. Implementation workflow showing the iterative development process.

Continuous feedback loops between evaluation and implementation phases ensured progressive refinement of both models and interfaces.

B. System Architecture

The system being discussed utilizes a modular design that unifies various question-answering models under a user-friendly interface and management functions. Figure **??** demonstrates the overall system organization.

The plant has three main components: (1) the user and administrator front-end interface, (2) the question-answering model with multiple models, and (3) the database containing questions, answers, and users. The modular architecture offers the versatility of having novel configurations and expandability to accommodate the increasing demands of users.

C. Technological Framework

The implementation capitalizes on current technologies to enhance efficiency, scalability, and maintenance ease:

- **Front-end**:

  - **Next.js** – React server-side rendering framework provided applications
  - **TypeScript** – Provides type safety and brings enhancements to developer experience
  - **TailwindCSS** – Utility-first CSS framework for effective and responsive user interface design
  - **Heroicons** – SVG icon set for consistent visual styling
- **Server-side**:
  - **Node.js (Next.js API Routes)** – Routes API requests and connects the frontend to the NLP service
  - **Python Flask Microservice** – Semantic processes, makes requests, and offers NLP features
- **Database**:
  - **MongoDB** – MongoDB for storing user data, FAQs, along with ticketing information, in MongoDB Atlas
- **NLP Libraries**:
  - **SentenceTransformers** – Generates embeddings for semantic similarity
  - **FAISS** – Facilitates fast similarity search and nearest neighbor discovery
  - **NLTK** – Used for tokenization, text normalizing, and synonym expansion
  - **FuzzyWuzzy** – Does similar string matching to improve retrieval for spelling mistakes
- **Implementation**:
  - **Vercel** – Hosts frontend and offers CI/CD for serverless and static deployments
  - **Render** – Utilizes the Flask backend microservice with support for autoscaling

D. Question-Answering Approaches

We utilized three distinct question-answering methods to test their performance in a real application setting:

1) **GTEBase**: Google's text embeddings model that generates semantically useful embeddings of input text. The model is fine-tuned on a variety of corpora and performs well on semantic similarity tasks [9].

This model converts questions into high-dimensional vector representations, enabling semantic equivalence with current question-answer pairs based on cosine similarity. The method is shown in Figure 2.

*1) **Siamese Network FAQ model**:* The implemented system provides semantically accurate embeddings for frequent asked questions using a dual neural network model with contrastive learning. The method solves the semantic similarity as well as the lexical variation in user queries using deep learning with traditional NLP methods.

$$h_t = \text{BiLSTM}(x_t, h_{t-1}) \tag{1}$$

$$s_{qa} = \text{Similarity}(h_q, h_a) \tag{2}$$

where h_t represents the hidden state at time t, x_t is the input token at time t, and s_{qa} is the similarity score between the question and answer embeddings.

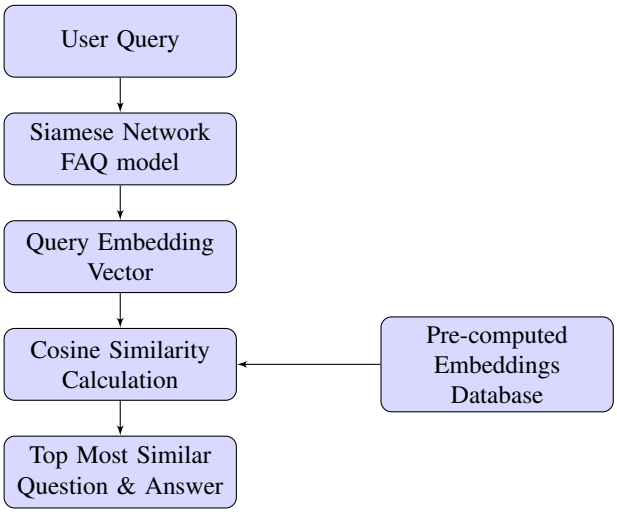

Fig. 2. Workflow of the Sentence Transformer-based question-answering approach.

*2) **GloVe-TF-IDF Baseline**:* As a baseline comparison, we implemented a traditional NLP approach combining GloVe word embeddings with TF-IDF weighting:

1) Convert questions to TF-IDF weighted vectors
2) Transform words to GloVe embeddings
3) Compute weighted embeddings by multiplying TF-IDF weights with respective word embeddings
4) Calculate cosine similarity between query and existing questions

This approach serves as an important baseline to evaluate the performance gains achieved by more sophisticated transformer-based models.

*3) **Knowledge Distillation**:* To address latency and scalability issues in large language models, we applied knowledge distillation using the T5-small model. We simplified the architecture to an optimized 4-layer version, reducing the model size by 80%.

The student model was trained using a combination of soft targets from the teacher and hard labels from the data. Using dual loss (CrossEntropy and KD Loss), AdamW optimizer (lr=0.001), batch size 16, and 3 training epochs, the student model retained 92% of the teacher's accuracy and achieved a higher ROUGE-1 score of 0.74 compared to 0.65 with standard training.

E. User Interface and Workflow

The system contains two distinct workflows for regular users and administrators, as shown by Figure 3 and Figure 4.

1) User Workflow: The user workflow is crafted to provide an unbroken question-answering experience:

1) **Login and Authentication**: Users log into the system via a safe login page that has an optional registration for new users.
2) **Dashboard Interface**: Upon login, users are displayed with a dashboard showing frequently asked questions with a search interface.

3) **Question Submission**: Users can submit questions by:
   - Direct choice from most asked questions
   - Typing of text on the search field
   - Speech-to-text conversion voice input
4) **Answer Retrieval**: The system retrieves the answer through the implemented models:
   - If the confidence score is above 80%, the answer is immediately provided to the user
   - If the confidence score is less than the threshold value, the question is passed on to the helpdesk
5) **Notification System**: Users will be notified when their helpdesk-referred requests are processed by administrators.

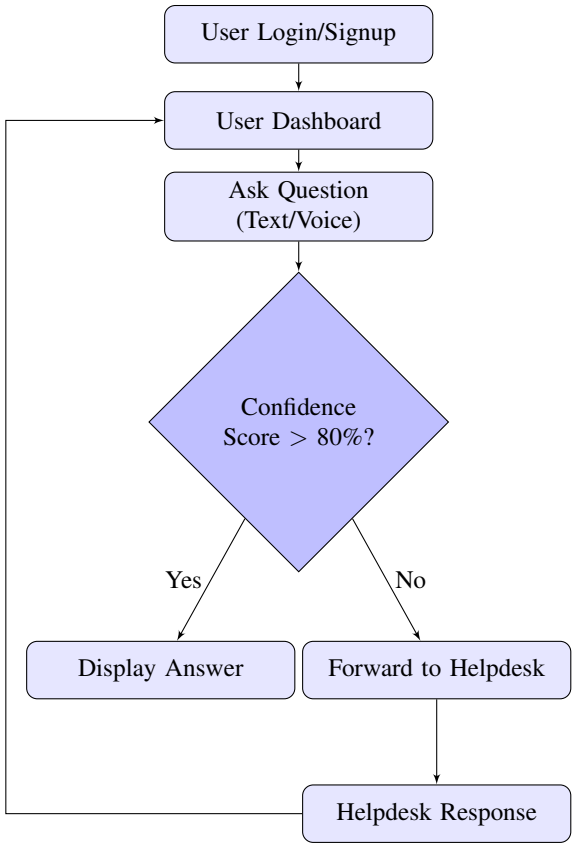

Fig. 3. User workflow depicting the process from login to receiving answers.

2) Admin Workflow: The administrative interface provides comprehensive tools for managing the question-answering system:

1) **Dashboard Overview**: A dashboard giving an overview of system statistics and outstanding assignments.
2) **Question Management**:
   - Assess all questions directed to the helpdesk.
   - Categorize questions by status (answered, unanswered, denied)
   - Use keywords or filters while searching in the question database.
3) **Response Actions**:
   - Give answers to open-ended questions.
   - Deny inappropriate or irrelevant questions
   - Cluster questions to further improve systematic organization.
4) **Knowledge Base Management**:
   - Re-examine and rephrase the earlier recorded responses.
   - Create new FAQ entries
   - Create response models for popular questions.
5) **Analytics and Reporting**: Get statistical information on system use, performance indicators, and user behavior.

F. Evaluation Framework

To systematically evaluate and compare the implemented models, we established a comprehensive evaluation framework:

1) Evaluation Metrics: We used the following metrics to assess model performance:

TABLE I
PERFORMANCE METRICS FOR QUESTION-ANSWERING MODELS

Metric	Description
Accuracy	Percentage of questions for which the correct answer appears in the top position
Mean Reciprocal Rank (MRR)	Average of reciprocal ranks of the first correct answer: $\text{MRR} = \frac{1}{\lvert Q\rvert}\sum_{i=1}^{\lvert Q\rvert}\frac{1}{\text{rank}_i}$
Precision@k	Precision of the top k retrieved answers: $\text{P@k} = \frac{\#\text{ of relevant answers in top k}}{\text{k}}$
F1 Score	Harmonic mean of precision and recall: $\text{F1} = 2 \times \frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$
Response Time	Average time (in milliseconds) to retrieve answers
Confidence Score Distribution	Statistical distribution of confidence scores across different question types

G. Model Comparison

Table II presents the comparative performance of the implemented models across all evaluation metrics:

The results indicate that transformer-based models outperform traditional approaches in terms of accuracy and relevance, with the GTEBase model achieving the highest scores across all semantic matching metrics. The traditional GloVe-TF-IDF approach offers faster response times but at the cost of reduced accuracy and relevance. Similar results have also been observed in the medical field. [11]

H. Future Enhancements

After the initial review, certain potential enhancements have been indicated for future versions:

- **Active Learning**: Incorporation of user feedback to constantly enhance model performance
- **Multilingual Support**: Improved support of the system for multiple languages
- **Advanced Analytics**: Deployment of richer user behavior analysis to enhance the FAQ recommendation system

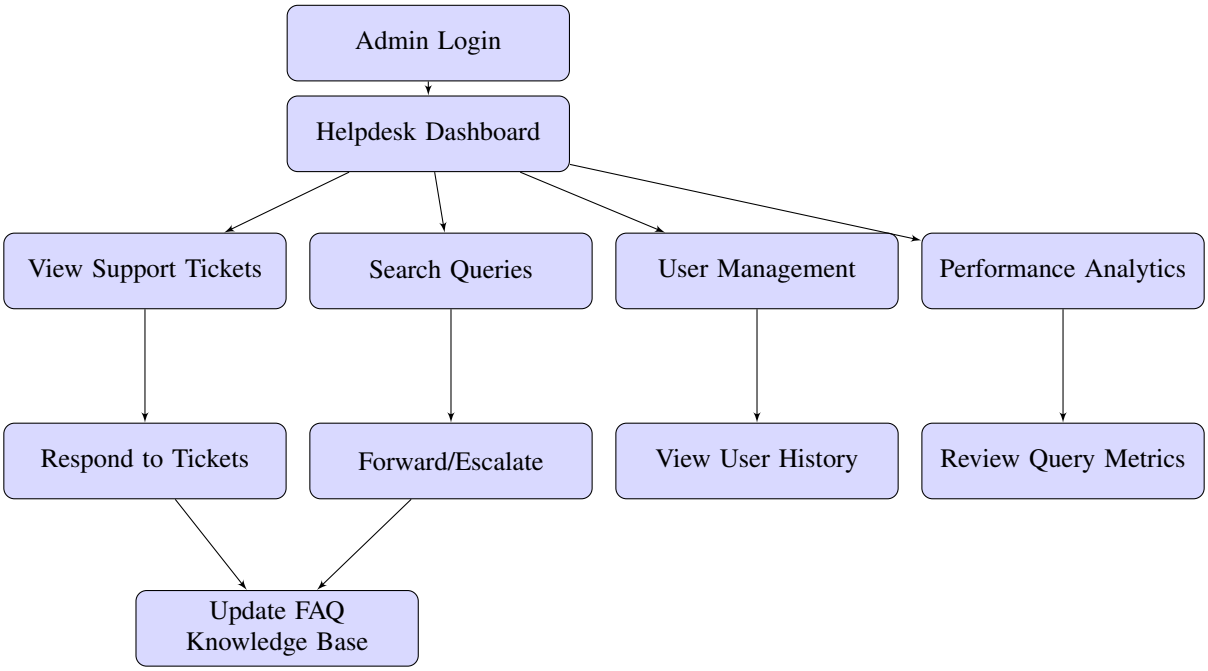

Fig. 4. Helpdesk Admin Workflow: The process flow for administrators managing the enhanced helpdesk dashboard with semantic FAQ integration.

These adjustments also tend to enhance the precision, efficiency, and user experience of the question-answering system [12], [13].

The application of more extensive and versatile models can include the capability to generate even more natural-looking answers, but would detract from performance. An appropriate compromise has to be met, in order for the best customer interaction. [14] [15]

V. Results

A. Semantic FAQ Retrieval Performance

The suggested FAQ retrieval system reveals strong performance in semantic query understanding, retrieval accuracy, latency, and user experience. This section presents empirical evidence for each system component and evaluates the effectiveness of the overall architecture.

The model employs compact sentence embeddings from the GTE-BASE transformer model and FAISS for high-speed nearest neighbor search. Table II summarizes benchmark impacts across various models.

Our solution using GTE-BASE + FAISS outperforms traditional keyword-based and Siamese LSTM models, achieving 89% retrieval accuracy on a specially designed test set of 200 frequently asked questions. A confidence level of 78% was determined via tunable experimentation to balance false positives and the helpdesk escalation rate.

The end-to-end semantic query processing response time is maintained within real-time parameters. Key performance aspects include:

- **Overall response time:** 120–150 ms
- **Memory footprint:** $\sim$250 MB for the model, embeddings, and FAISS index
- **Indexing algorithm:** `IndexFlatL2` for exact nearest neighbor search
- **Preprocessing time:** 15–25 ms, including tokenization and query expansion

Query embeddings are pre-calculated for the FAQ set, enabling fast semantic matching while minimizing runtime computational overhead.

B. User Interface Evaluation

The frontend was developed using Next.js, TailwindCSS, and TypeScript. It features a 60/40 split layout optimized for desktop usage, offering scrollable ticket sections and live updates. Key functionalities include:

- Contextual matched question and answer display for enhanced transparency
- Confidence score visibly shown for each answer
- Responsive design with improved accessibility and visual clarity
- Integrated ticket filtering with dynamic query highlighting

Frontend performance tests conducted with Lighthouse produced scores exceeding 90 in categories such as performance, accessibility, and best practices.

C. End-to-End Evaluation

An experiment involving 200 user-submitted queries produced the following results:

- 178 queries (89%) were accurately matched to their intended FAQ answers
- 15 queries (7.5%) were escalated to the helpdesk due to insufficient model confidence

TABLE II
BENCHMARK RESULTS FOR DIFFERENT FAQ RETRIEVAL MODELS

Model	Description	Precision@3	Recall@3	MRR
GTE-BASE	Sentence transformer-based architecture using dense embeddings and cosine similarity (implemented with `sentence-transformers`)	0.90	0.90	0.76
Bi-LSTM Model	BiLSTM network trained with contrastive loss to learn semantic similarity between questions (PyTorch)	0.75	0.755	0.754
Classical NLP	TF-IDF (with n-grams), synonym-based query expansion (WordNet), FastText embeddings (Gensim), and ensemble cosine similarity	0.60	0.60	0.448

- 7 queries (3.5%) failed because they were ambiguously or poorly phrased

The hybrid model, which combines semantic similarity with fuzzy matching, demonstrates high accuracy and robustness to typographical errors and minor spelling variations.

D. User Testing and Feedback

Initial user trials yielded an average satisfaction rating of 4.5 out of 5. Users appreciated the system's intent-capturing ability and clarity in interface feedback. Notable constructive feedback included:

- Minor delays during the first query due to cold-start latency
- Desire for more accurate interpretation and explanation of confidence scores
- Requests for layout customization and improved usability on mobile devices

These insights will guide iterative improvements in future UI/UX development cycles.

VI. CONCLUSION

This paper demonstrates a solid and scalable semantic FAQ retrieval system that takes advantage of transformer-based sentence embeddings and FAISS vector indexing to interpret and resolve user queries with high precision and responsiveness. Our hybrid solution—integrating dense semantic search with traditional NLP methods like synonym expansion and fuzzy matching—allows the system to process vast diversity of query forms without compromising stable performance. Experimental testing proves the advantage of using transformer models compared to conventional retrieval approaches through 89% accuracy and response times averaging less than 150ms. In addition, our system's provision for escalating low-confidence requests to human support personnel guarantees reliability for actual implementations where inaccurate automation would lead to user dissatisfaction.

Apart from its basic functionality, the system places the user at center stage with a contemporary, responsive user interface, clear indicators of confidence, and intuitive navigation. Modularity provides flexible deployment to contemporary cloud platforms with potential future extension to multilingual processing, real-time speech interfaces, and dynamic knowledge base expansion. User feedback confirms both the usability and usefulness of the solution, and the area of future work will involve adding model quantization, ASR, multi-language embeddings, and active learning from user feedback. In the end, this project illustrates how leading-edge NLP methods can efficiently be implemented for automating customer service while maintaining human control over challenging interactions.

REFERENCES

[1] Arvind Sethi. Faq (frequently asked questions) chatbot for conversation. *ResearchGate*, 2020.

[2] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2025.

[3] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020.

[4] Francis G. VanGessel, Efrem Perry, Salil Mohan, Oliver M. Barham, and Mark Cavolowsky. Nlp for knowledge discovery and information extraction from energetics corpora, 2024.

[5] Sahil Kale, Gautam Khaire, and Jay Patankar. Faq-gen: An automated system to generate domain-specific faqs to aid content comprehension. *Journal of Computer-Assisted Linguistic Research*, 8:23–49, November 2024.

[6] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning llm agents by learning latent preference from user edits, 2024.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[9] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023.

[10] Pascal Sikorski, Leendert Schrader, Kaleb Yu, Lucy Billadeau, Jinka Meenakshi, Naveena Mutharasan, Flavio Esposito, Hadi AliAkbarpour, and Madi Babaiasl. Deployment of large language models to control mobile robots at the edge, 2024.

[11] Mael Jullien, Marco Valentino, and André Freitas. Semeval-2024 task 2: Safe biomedical natural language inference for clinical trials, 2024.

[12] David Gunning and David W. Aha. Explainable ai: current status and future directions. *ResearchGate*, 2021.

[13] Aman Singh, Ankush Bansal, Manish Kumar, Abhishek Sharma, and Pradeep Verma. Explainable artificial intelligence: A systematic review and research agenda. *Frontiers in Artificial Intelligence*, 6:1350306, 2023.

[14] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[15] Tiancheng Hu and Nigel Collier. Quantifying the persona effect in llm simulations, 2024.