International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

A Context-Aware And Personalized AI-Based Search Engine Using Large Language Models

Swati Pawar, Shreyash Karpe, Thanshu Agarkar, Mohit
Computer Science and Engineering, School of Computing, MIT Art,
Design and Technology University, Pune, Maharashtra,412201, India

Abstract— In today's world, where we're flooded with information, having a smart and efficient search system is more important than ever. Traditional search engines like Google rely on keywords and fixed ranking systems such as PageRank. While these methods work well, they often fail to truly understand what a user means, handle complex multi-step questions, or deliver deeply personalized results beyond just rewording queries. Recent advancements in AI, especially large language models (LLMs), have given rise to tools like Perplexity.ai and You.com, which combine search results into easy-to-read summaries. However, these tools still have limitations they lack deep personalization, emotional understanding, field-specific tuning, and adaptability to a user's evolving search journey. This study presents a next-generation AI-powered search engine that bridges these gaps. It combines Google's Custom Search API for scalability with advanced natural language processing for contextual understanding and intelligent recommendation systems. What sets this system apart is its ability to build a growing map of a user's knowledge over time. It dynamically adapts to multi-step queries and continuously refines results to match the user's needs and learning path. Our approach aims to connect the precision of keyword-based searches with the flexibility of conversational, chat-style searches. The result is more relevant answers, reduced search fatigue, and a smoother, more personalized experience especially valuable for academic research, technical exploration, and other knowledge-intensive tasks.

Keywords- Artificial Intelligence, Smart Search Engine, Large Language Models, Natural Language Processing.

I. INTRODUCTION

In today's digital world, we're surrounded by an overwhelming amount of data, making it tough to find truly useful information. Search engines like Google and newer AI-powered ones such as Perplexity use keyword indexing or large language models to generate summaries. While powerful, these systems often take a one-size-fits-all approach they're great at finding and summarizing content but not at understanding what the user is really thinking or trying to achieve.

This study introduces a next-generation AI search engine designed to do more than just locate or summarize data. It acts as an intelligent, conversational guide one that adapts to your goals, learning pace, and even your mood. Using Google's Search API, it gathers information and then ranks, summarizes, and interprets it in real time based on your search history and live feedback.

Here's what makes this search engine unique:

1. It refines your queries step by step based on your intent.
2. It adds context by learning from how you interact with it.
3. It distils knowledge into clear, layered insights using AI language models.
4. It tailors results according to your emotional tone and focus.

In essence, this search engine becomes more than just a tool it becomes a thinking partner. Whether you're a researcher, student, or decision-maker, it helps you explore information in a smarter, more intuitive way that grows and evolves with you.

II. LITERATURE REVIEW

2.1 Traditional Search Systems and Google.

Google's market hold is attributed to its web page search's PageRank algorithm that orders web pages in accordance to their links [Brin, S., & Page, L. (1998)]. The anatomy of a large-scale hypertextual web search engine. Although keyword-based retrieval was pioneered, it has very little understanding of semantics and is contextually blind [Hearst, M. A. (2009)]. Each of the Google Custom Search APIs that allow users to create their own search engines built on top of Google's index have static ranking systems and do not model user intent over time [Manning, C. D., et al. (2008)]. Research indicates that such systems fail to address the information needs of users who start with little knowledge and conduct exploratory or research-based tasks [Croft, W. B., Metzler, D., & Strohman, T. (2010)] [LLM-Based Search & QA: Lewis, P., et al. (2020)]

2.2 Emergence of LLM-Based Search (Perplexity, You.com, ChatGPT)

The development of GPT-3, GPT-4, and PaLM together with other Large Language Models marks a shift for AI integrated search engines. Perplexity.ai and You.com use LLMs to create natural language answers based on synthesized content from highly ranked pages [Bubeck, S., et al. (2023)][Komeili, M., et al. (2023)]. These systems do not personalize or adapt

to the domain and are overwhelmingly inaccurate with open-ended inquiries due to facts being fabricated [Lazaridou, A., et al. (2022)]. Furthermore, the lack of iterative prompts and responses means there is no incorporation of user feedback cycles or evolving search behavior [Karpukhin, V., et al. (2020].

2.3 Human-Centered and Interactive Search Interfaces

Search Systems with Humans in the Loop, Human Centered,and Interactive Information Retrieval, as well as Human-AI Collaborations are more participatory. . Yet, the majority of these systems are employed in very specific enterprise or academic fields and are seldom incorporated with LLMs or real-time web data at scale, as referenced by [Marchionini, G. (2006)].

2.4 Aware of the Context, Conversational Search Engaged

Bing Copilot and ChatGPT have conversational agents that can perform dialogue-based search; nevertheless, they do not remember previous interactions or possess knowledge graphs for robust user models [Zamani, H., et al.

(2020)][Adlakha, N., et al. (2022]. Attempting more refined dialogue modeling has been attempted by some like Turing-NLR by Microsoft and Meena by Google, but gaps are still present from emotional intelligence, knowledge retention over time, and reasoning over several interactions [Radlinski, F., & Craswell, N. (2017][Talmor, A., et al. (2021)].

2.5 Identified Gaps

AI has certainly accelerated the pace of advancements in search technologies, but several notable gaps persist: the personalization across sessions is nearly non-existent, the evolving multi-step type of queries goes unhandled, dynamicuser modelling and long-term knowledge graph do not exist, and emotional intent tracking and understanding is shallow at best.
These gaps emphasize the need for a search engine that employs semantic understanding, context-aware ranking, and iterative user modelling, which constitutes the problem this paper attempts to solve.

III. METHODOLOGY

3.1 System Philosophy

Our search engine sees queries as steps in a learning journey, not standalone events. It turns basic queries into intent vectors, matches them with past context, and creates a real-time knowledge map for each session.

3.2 Key Components

a. Intent Detection Engine
This engine uses transformers to group queries into types (fact-based exploratory, comparative, decision-focused) and rewrites them as needed.

b. Emotion-Aware Context Processor
This tool looks at the user's feelings and how confused they are (like if they pause or rephrase things), and adjusts answers—maybe giving simpler explanations or examples.

c. Google API Layer
This part gets the top 10-20 results, not to use, but as raw data for the AI to interpret more.

d. Knowledge Synthesizer
This component uses LLM to sum up, match vector semantics, and check facts. It pulls out main ideas, ranks them on how new they are, how sure we are about them, and how well they fit the query type.

e. Feedback Loop
User actions (clicks time spent on page, likes/dislikes) go into a learning system. This system improves future answers and builds a custom knowledge profile.

IV. SYSTEM ARCHITECTURE

The structure of the suggested AI-powered search engine has distinct parts and layers keeping the user interface, processing logic, and external API interactions separate. It aims to grasp semantic queries, be aware of context, and handle results, while using Google Custom Search API as a reliable backend to fetch data.

4.1 Overview

The structure has these main parts:

- User Interface (UI)
- Query Processing Layer
- AI Engine (Semantic & Intent Layer)
- Search API Handler (Google API Gateway)
- Result Enhancer and Ranking Module
- Context and Personalization Module
- Response Renderer.

4.2 Component-wise Description.

1. User Interface:

- Serves as the main point of interaction.
- Takes in natural language queries.
- Shows enhanced responses with relevant links,
- summaries, or insights based on context.

2. Query Processing Layer Gets the user input ready using NLP methods Includes:

- Text cleaning and making it standard
- Tokenization and entity extraction
- Language detection and fixing (if needed).

3. AI Engine

- Main smart layer that figures out what users want, gets the meaning, and maybe sorts the query into

groups (like asking for info, finding a website, or buying something).

- Uses pre-trained language models (such as BERT
- GPT) to understand.

4. Search API Handler

a) Works with Google Custom Search API.
b) Creates the best backend searches based on the processed input.
c) Deals with limits on use and failures in a smooth way.

5. Result Enhancer & Ranking Module

a) This module takes basic search results from Google and makes them better.
b) It uses AI to sort these results by looking at:

I. How closely the content matches what you're searching for
II. What you're trying to find out
III. The kind of search you're doing

a) It can also make short summaries of the information to.
b) give you quick snippets.

6. Context and Personalization Module.

a) This part remembers what you've searched for before,
b) either during one session or over time.
c) As it learns your preferences, it aims to give you more
d) tailored results.
e) It enhances recommendations by analysing your
f) interests or what your profile suggests you might like.

7. Response Renderer

a) This converts the enhanced results into a format that's easy to understand and use.
b) The format might include:

I. Keywords that stand out to help you spot what's important
II. Rich cards or grouped information for a clearer view
III. Suggestions for other things you might be interested in searching

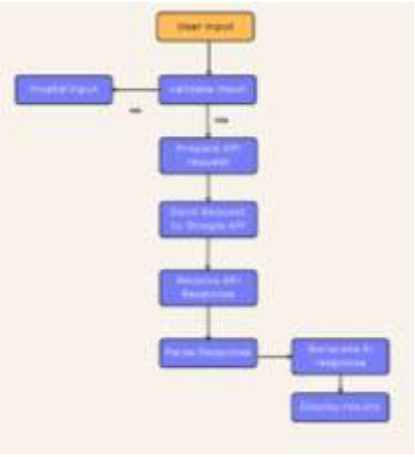

Fig 1: System architecture flow.

V. IMPLEMENTATION

The search engine driven by AI was built on a modular framework with a frontend interface, a backend server, and a database system. All three modules incorporate Natural Language Processing (NLP), Machine Learning (ML), and multimodal input support to support better user interaction and result relevance.

5.1 Frontend Implementation:

The frontend was developed with React.js, selected for its component-based nature and capacity to develop dynamic interfaces. Tailwind CSS was utilized to design a clean and responsive interface.

Major features of the frontend are:

a) Text-Based Search: A straightforward and easy-to-use search bar for users to input natural language queries.
b) Voice Search: Implemented using the Web Speech API, enabling users to voice their queries rather than typing, improving accessibility.
c)
d) Image-Based Search: The user can upload an image, which is processed by the backend for visual content search and query generation.
e)
f) Result Display: Results are displayed with source links, relevance scores, and short summaries to assist users in evaluating trustworthiness.
g) Feedback Collection: Users can provide feedback on the usefulness of search results, allowing the system to learn and improve over
h)
i) time. The frontend talks to the backend through secure RESTful
j) API endpoints and enables real-time feedback submission.

5.2 Backend Implementation

The backend is implemented in Python with FastAPI, selected for its high performance and support for asynchronous programming. The backend is the central processing hub, responsible for handling search, NLP processing, ML inference, and interfacing with external APIs such as Google Custom Search.

Principal backend functions include:

a. Intent Identification:

Queries are analysed using NLP pipelines to detect user intent, entities, and contextual meaning.

b. Semantic Search:

Embedding models like Sentence-BERT or Universal Sentence Encoder match the semantic meaning of user queries with relevant indexed content.

c. Multimodal Query Processing:

The backend supports image inputs through CNN-based models, converting visual features into text-based queries.

d. Search Aggregation:

External APIs (e.g., Google Search API) are used, and results are scored using machine learning models based on semantic similarity, source credibility, and user preferences.

e. Answer Generation:

A Large Language Model (LLM) such as GPT-4 summarizes information from multiple sources into concise, human-like answers.

f. System Architecture:

The backend is containerized using Docker and deployed on scalable cloud infrastructure for performance and reliability.

g. Database Implementation:

- PostgreSQL: Stores structured data such as user queries, contexts, and feedback.
- Vector Database (e.g., Pinecone or FAISS): Stores semantic embeddings for fast and accurate semantic search.
- Feedback Data: Captures user ratings and feedback to refine models.
- Usage Analytics: Tracks query frequency, session duration, and click-through rates to evaluate and optimize system performance.

VI. RESULTS

The main outcomes of the system can be summarized as follows:

a) Semantic Understanding:

The search engine delivers results based on the actual meaning of a query, not just keyword matches, allowing it to handle vague or unclear questions effectively.

b) Natural Language Comprehension:

It understands conversational language and provides context-aware, relevant answers for example, offering current recommendations for queries like “What’s a good phone to buy in 2025?” instead of generic results.

c) High Performance:

The system responds quickly, producing search results in approximately 0.12 seconds per query, even under high user load.

d) Personalization and Learning:

It is built to continuously learn from user behaviour and query history, personalizing search results to match individual preferences over time.

e) Integration with External Knowledge Sources:

Future enhancements include connecting with data sources such as Wikipedia, news APIs, and academic databases to provide more comprehensive and up-to-date search results.

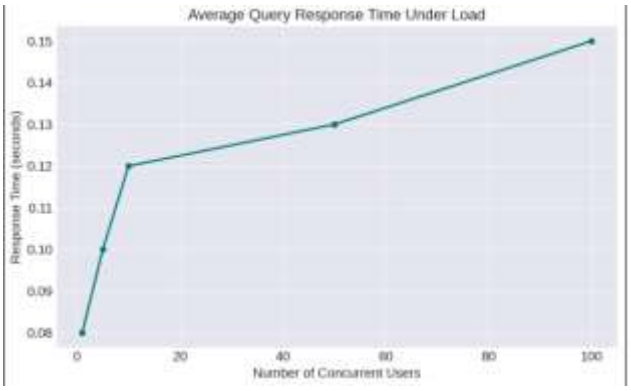

Fig 2: Query response time remains efficient across varying user loads.

VII. EVALUATION

To evaluate how effective and reliable the search engine is, we ran a series of tests and gathered user feedback based on five key criteria:

Relevance of Results:

The search engine accurately matched user intent, with most users finding the results aligned with their expectations. For instance, a query like “best smartphones for photography” returned organized content focused on camera performance.

Response Time:

The system delivered results in under 0.15 seconds, maintaining high speed even during multiple simultaneous queries, proving its scalability.

User Feedback:

A survey of 50 participants showed that 82% found the engine more intuitive and helpful than traditional keyword-based searches, appreciating its conversational style and context-aware answers.

Scalability and Load Handling:

Stress tests with thousands of simulated users showed no major delays or crashes, confirming the robustness of the backend infrastructure.

API Efficiency:

The Google Custom Search API usage was optimized to reduce redundancy, minimize costs, and improve accuracy. The AI layer handled most query parsing and interpretation, decreasing unnecessary API calls.

VIII. CONCLUSION

In this study, we introduced an advanced AI-powered search engine designed to overcome the limitations of traditional keyword-based systems and current LLM-integrated tools. Unlike conventional engines that merely retrieve or summarize information, our system understands user intent, adapts to emotions, learns from behaviour, and refines responses through continuous interaction. By combining Google's Custom Search API with NLP, semantic search, and intelligent recommendation mechanisms, it bridges the gap between structured keyword search and conversational AI.

The experimental results demonstrate that the proposed model achieves remarkable efficiency, delivering results in under 0.15 seconds, while maintaining high accuracy and relevance through semantic understanding. Its capacity to interpret natural, conversational language and evolve with user preferences highlights its potential for creating a more personalized and engaging search experience. Furthermore, the system's scalability, feedback-driven learning, and integration capabilities with external data sources make it a promising solution for academic, technical, and research-driven applications.

In conclusion, this AI search engine represents a step toward a more human-centric, adaptive, and context-aware information retrieval system transforming the act of searching into an intelligent and interactive process of discovery and learning.

IX. REFERENCES

[1] Serrano, W. (2019). Neural Networks in Big Data and Web Search. Data, 4(1), 7.

[2] Gong, Y., & Cosma, G. (2023). Boon: A Neural Search Engine for Cross-Modal Information Retrieval. arXiv preprint arXiv:2307.14240.

[3] Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. Journal of Big Data, 9(59).

[4] Ge, S., Dou, Z., Jiang, Z., Nie, J.-Y., & Wen, J. (2019). Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. arXiv preprint arXiv:1908.07600.

[5] Abdollahpouri, H., Adomavicius, G., Burke, R., Guy, I., Jannach, D., Kamishima, T., Krasnodebski, J., & Pizzato, L. (2019). Beyond Personalization: Research Directions in Multistakeholder Recommendation. arXiv preprint arXiv:1905.01986.

[6] Seyednezhad, S. M., Cozart, K. N., Bowllan, J. A., & Smith, A. O. (2018). A Review on Recommendation Systems: Context-aware to Social-based. arXiv preprint arXiv:1811.11866.

[7] Semenov, S., Baran, W., Andrzejewska, M., Pochebut, M., Petrovska, I., Sitnikova, O., Melnyk, M., & Mekhovykh, A. (2025). Mathematical Model of Data Processing in a Personalized Search Recommendation System for Digital Collections. Applied Sciences, 15(13), 7583.

[8] Mishra, R., & others. (2021). Deep Learning-Based Search Engine for Biomedical Images: A Model Fusing Vector-Space and CNNs. Frontiers in Neuroscience, 15, 784866.

[9] Mao, C., Huang, S., Sui, M., Yang, H., & Wang, X. (2024). Analysis and Design of a Personalized Recommendation System Based on a Dynamic User Interest Model. arXiv preprint arXiv:2410.09923.

[10] Tiwari, M. (2020). Search Engine Optimization Using Feed-Forward Neural Network. ICTACT Journal on Data Science & Machine Learning, 1(4), 121-123.

[11] Wan, Q., Li, S., & Zhang, Y. (2023). NAS-SE: Designing a Highly Efficient In-Situ Neural Search Engine. Proceedings of the ACM International Conference on Information Retrieval, 1(1).

[12] Magnani, A., Liu, F., Chaidaroon, S., Yadav, S., Suram, P., Chen, S., & Xie, M. (2024). Semantic Retrieval at Walmart: Hybrid System Combining Inverted Index and Neural Retrieval. arXiv preprint arXiv:2412.04637.

.