International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

NEXIS: A Voice-Driven, Context-Aware AI Workspace for Document-Grounded Conversational learning

Mohammed Mehdi[1], Mohammed Ahad Fazal[2], Bhavya C[3]
Student, Department of BCA, St. Francis College, Bengaluru[1]
Student, Department of BCA, St. Francis College, Bengaluru[2]
Assistant Professor, Department of BCA, St. Francis College, Bengaluru[3]
Email Id: bhavya.c@stfranciscollege.edu.in

Abstract:

Studying or working across multiple documents and notes often involves repetitive searching, switching between files and re-explaining context for every follow-up question. NEXIS is a Windows-based desktop environment and this problem is solved by using real time and contextual voice conversation. The user creates a "context" through the act of writing, copying and importing data into NEXIS including pdf, docx, txt and mark down documents and arranging them into an organized library with folders, tags and full text search.

Voice-enabled interaction works using the Gemini Live API, which enables live listening and talking, interruption capabilities, and auto-reconnect, enabling users to have a natural conversation that is based on what the users chose rather than a textual one, which would need text search. The transcripts and session data will be stored locally using SQLite database that includes full-text functionality. The credentials to access the API are stored securely using the key ring package in Python, which uses the Windows Credential Manager for storage. NEXIS also includes study-related work flow features such as Quiz-Me self-checks and Teach-Back coaching, as well as pomodoro focus sessions.

NEXIS makes it possible to use conversational speech over conventional text searches as a way of investigating, refining, and retrieving information from one's documents and notes. The system is an example of the integration of real-time conversational AI, local context persistence, and the structured learning work flow in one desktop environment.

Keywords: Voice-To-Voice interaction, Context-Aware AI, Document-Grounded Conversation, Conversational Learning, AI Learning Assistant

1. INTRODUCTION:

Studying or working across several documents is rarely linear. A learner reading lecture slides, a research paper and personal notes repeatedly searches for passages, switches between files, and re-explains the context of the task every time a follow-up questions comes up. Large language models (LLMs) can answer such questions fluently [11][12], but a general-purpose assistant does not know which documents the learner is working with. It answers from its training data and can produce statements that sound plausible but are not supported by the same material, a failure known as hallucination [9],[10]. Two strategies are commonly used to tie answers to a user's documents retrieving relevant passages from a corpus [1],[2], or placing the documents directly in the long context of a modern model [4],[13],[14]. Either way, the interaction is usually typed chat, which keeps the repetitive cycle of typing a question, reading an answer and typing the next one.

Voice offers a more natural channel. Recent speech-language models support real-time spoken dialogue [16]-[19], but these advances are mostly demonstrated as general conversational agents. They are rarely combined with a personal document

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

library, local storage of past sessions, and the structured study practices that learner actually use.

This paper presents **NEXIS,** a Windows desktop environment that combines these elements. A user builds a "context" by writing, pasting and importing PDF, DOCX, TXT and Markdown files into an organized library with folders, tags and full-text search. The user then talks to the assistant about that context through the Gemini Live API [14], with live listening and speaking, interruption (barge-in), automatic reconnection and multilingual conversation. NEXIS does not chunk or retrieve the active context is supplied to the model in full. Documents and transcripts are stored in locally in SQLite, and API credentials are kept in the Windows Credentials Manager. On top of this core, NEXIS offers three study workflows Quiz-Me self-checks, Teach-Back coaching and Pomodoro focus sessions.

The main contributions are:

1. **A voice-first, document-grounded workspace** that links a user-built library to real-time spoken conversation using full-context grounding, so follow-up questions need no re-explaining.
2. **A local-first design** that stores documents and sessions in SQLite, provides FTS5 full-text search over the library [**and the transcripts**], and keeps credentials in the operating system's credential store.
3. **Integrated learning workflows** (Quiz-Me, Teach-Back, Pomodoro) that run on the same context rather than as separate tools.
4. **A primary study** with ten undergraduate students, in which participants learned the system after a brief introduction and valued its multilingual support (English, Hindi, Kannada, Urdu) and repeated, context-aware doubt clearing

2. RELATED WORK:

Grounding answers in documents: Retrieval-augmented generation (RAG) conditions a model on retrieved evidence, and surveys trace its evolution from simple to modular pipelines [1], [2]. Research systems add self-critique and adaptive retrieval [3]. An alternative is to skip retrieval and give a long-context model the whole text. Li et al compare the two on public benchmarks and find that long-context models outperform RAG on average when resources allow, while RAG remains much cheaper [4]. NEXIS follows the long-context route because a personal library is small enough to supply in full. Both routes have known weaknesses. Models may use information in the middle of a long input less reliability [5], and RAG pipelines fail in practice through missed context, extraction and ranking errors [6].

Real-time voice interaction: Speech recognition has been scaled through large weakly supervised training [15]. Newer systems target conversation Moshi models real-time speech-text dialogue [16], GPT-4o accepts and produces audio natively [17], and LLaMa-Omni and Mini-Omni generate speech while text is still being produced [18], [19]. These works focus on the model. NEXIS addresses the application layer. A hosted live speech model combined with a user-owned library, interruption handling, reconnection and local session storage.

LLMs in education: Surveys describe opportunities in tutoring, context generation and assessment, and raise concerns about accuracy and ethics [21], [23]. Evidence suggests that LLM-generated help can produce learning gains comparable to human-authored help in a controlled setting [22]. Ruffle&Riley is a conversational tutoring system built around a learning-by-teaching dialogue [20]. Related to Quiz-Me, retrieval practice and practice testing have been studied in university teaching [26], and LLM-generated practice questions have been tested in university courses [30]. Related to Teach-Back, Debbane et al. report challenges students face when teaching a virtual agent [27]. For focus sessions, a style of university students compared Pomodoro-style systematic breaks with self-regulated breaks [28]. NEXIS differs from these systems in grounding all of these activities in documents the learner chooses, with voice and local persistence in one environment.

Evaluating grounded assistants: Automatic frameworks such as RAGAs and ARES assess faithfulness and answer relevance for retrieval-based systems [24], [25]. Our study is qualitative and does not apply them (Section 10). We list quantitative evaluation as future work (Section 12).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

3. PROBLEM STATEMENT AND OBJECTIVES:

Problem statement. Let C be the text of the learner's active context, drawn from a library D of at documents and notes, and let u t be the spoken input at turn t with history H_{t-1}. The goal is to produce a spoken response r t that (i) is supported by C (ii) arrives quickly enough to feel conversational, (iii) allows the user to interrupt and the session to recover from connections failures, and (iv) leaves a record of the interaction on the user's machine. Typed or long-context [4], require typing and re-stating context. General voice assistants are fluent but not tied to a personal library [16], [17]. Study tools rarely combine both with local storage.

Objectives:

- **01 (context creation).** Let users build and organize a library by writing, pasting and importing PDF, DOCX, TXT and Markdown, with folders, tags and full-text search
- **02 (voice-first grounded dialogue).** Support continuous spoken conversation over the active context, with barge-in, multilingual use and automatic reconnection.
- **03 (local persistence).** Store contexts, sessions and transcripts locally in SQLite [with full-text search over the library and the transcripts].
- **04 (secure credentials).** Keep API credentials in the operating system's credentials store, not in configuration files or the database.
- **05 (learning workflows).** Provide Quiz-Me, Teach-Back and Pomodoro on the same context

4. NEXIS SYSTEM OVERVIEW

NEXIS is a single-user Windows desktop applications, built with PySide6 (Qt) in Python, for conversational study over a personal document library. A typical session follows five steps:

Build the context: The user creates contexts by typing notes in the built-in editor, pasting text, or importing files (PDF, DOCX, TXT, Markdown, log files up to 50MB each). Text is extracted on import and stored as a complete document.

Organize the library: Contexts are placed in folders and given tags. A search box queries context names, content and tags with SQLite FTS5 and returns up to 25 results by default.

Activate a context: The user marks a context as active. Its full text becomes the grounding material for the conversation.

Talk: In the voice panel, the user speaks to the assistant, which answers aloud about the active context. The user can interrupt at any time, switch to another context mid-session, change the response and transcript languages, and rely on automatic reconnection if the connection drops. Transcripts of each turn are saved locally.

Study: On the same context, three workflow tools are available: Quiz-Me (adaptive self-check questions), Teach-Back (the user explains a concept and receives feedback) and a Pomodoro timer. The Result panel also offers text actions such as Explain, Summarize, MCQs and Viva Questions.

5. SYSTEM ARCHITECTURE AND COMPONENTS

NEXIS is organized as five layers , each implemented as a separate package.

- **Presentation layer.** PySide6 widgets: main window, context workspace and library, voice panel, quiz panel, Teach-Back panel and study timer.
- **Ingestion layer.** An extractor converts supported files to plain text (pypdf for PDF, python-docx for DOCX, built-in decoding for TXT, Markdown and log files).
- **Storage layer.** A SQLite database in WAL mode with foreign keys holds contexts, folders, sessions, messages, settings, a cache of AI outputs and saved insights, FTS5 indexes support search.
- **Conversation layer.** A bridge component builds the Gemini system instruction from the active context and manages context switching. A live session module handles microphone capture, streaming, playback, interruption, reconnection and the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

language-consistency component (Section 6).

- **Learning layer.** Quiz and Teach-Back modules call the Gemini text model. The Pomodoro module is a local timer with no network dependency.

Table 1: Components and technologies

Component	Technology	Role
GUI	PySide6(Qt)	Panels for library, voice, quiz, Teach-Back, timer
Import	Pypdf python-docx, Python text decoding	Extract text from PDF, DOCX, TXT, MID, LOG
Storage	SQLite (WAL) with FTS5	Contexts, folders, transcripts, settings
Voice I/O	Sound device	Microphone capture and speaker playback
Model access	Google-genai SDK	Gemini Live (audio) and text models
Credentials	Python keyring, Windows Credential Manager	Store the API key outside code and database
Study timer	Local module	Python Pomodoro phases and counters

Everything except a model call stays on the user's machine, and the API key is never stored in the database or configuration files.



Fig. 1. Layered architecture: presentation, ingestion, storage, conversation (Gemini Live, language consistency), learning, with keyring/Credential Manager beside storage.

Table 2: Live session configuration

Parameter	Value
Live model	Gemini-2.5-flash-native-audio-latest
Text model	Gemini-flash-latest (with fallbacks)
Input audio	PCM 16-bit mono, 16 kHz, 320-sample blocks(~20ms)
Output audio	PCM 16-bit, mono, 24kHz
Voice	Algenib
Activity detection	Automatic (server defaults)
Interruption	START_OF_ACTIVITY_INTERRUPTS + local audio flush
Context comprehension	Sliding window
Initial retries	3 attempts (0.75s, 1.75s, delays)
Runtime reconnects	Up to 5, exponential backoff, cap 15s
Transcription	Live input and output, user transcript cleaned by text model

6. CONTEXT-AWARE DOCUMENT-GROUNDED CONVERSATION

Full-context grounding: Many grounded assistants use RAG; documents are split, embedded and indexed, and the top-ranked passages are retrieved for each question [1], [2]. NEXIS instead stores each document whole and places the entire active context in the system instruction of the live session. The model therefore sees everything the user has chosen, and follow-up questions need no retrieval step. Formally, if C is the active context and H_{t-1} the conversation so far, the model at turn t responds conditioned on (C_H_{t-1}, u t). The instruction also tells the model to [paste the grounding wording from bridge.py, e.g., answer from the context and say when it is not covered]. This is practical because modern models accept long inputs [13], [14], and Li

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

et al. report that long-context prompting can outperform retrieval on average when its higher cost is acceptable [4]. The results in [13], [14] describe the model family, not the limit of the specific live model used here.

Switching context: When the user changes the active context mid-sessions, NEXIS sends up to 8,000 characters of the new context to the running live sessions instead of restarting it, so the conversation continues without a break.

Trade-offs: Longer instructions increase cost and may increase latency [4], and model may use information in the middle of long inputs less reliably [5]. Text must be extracted first, and pypdf cannot read scanned PDFs, so extraction quality remains a failure point (compare [6]). NEXIS gives no passage-level citations [7], and a context larger than the model limit cannot be supplied. Other features use only the beginning of the context. Grounding reduces but does not eliminate hallucinations [9], [10] and this study did not measure faithfulness (section 12).

7. DATA PERSISTENCE AND SECURITY

Local storage: NEXIS keeps all persistent data in one SQLite database on the user's machine, with write-ahead logging (WAL) and foreign-key enforcement enabled. The schema is versioned, so upgrades can migrate existing data. Transcripts text in messages is the version cleaned by the language-consistency step (section 6), so it may differ from the exact words spoken.

Full-text search. SQLite FTS5 provides keyword search over contexts fts, which covers context names, content and tags and returns up to 25 contexts by default. Database triggers keep the indexes synchronized with their source tables on every insert, update and delete. This search is separate from grounding the model receives the full active context (section7), not search results.

Credential security: Access to Gemini requires an API key. NEXIS stores it with Python's keyring package, which uses the Windows Credential Manager, under the service name Nexis_VI and the username gemini api key. The key is not written to the database, settings, logs or application files.

8. LEARNING-ORIENTED FEATURES

NEXIS adds study workflows on the same context that grounds the voice conversation. Quiz-Me, Teach-Back and the text actions call the Gemini text model; Pomodoro is a local timer (Fig.2 shows the Quiz-Me and Teach-Back loops).

Quiz-Me: Practice testing benefits university learning [26], and LLM-generated practice questions have been tested in university courses [30], in NEXIS:

1. Gemini receives the active context (up to 14,000 characters) and returns a JSON question bank of at least 12 questions, or the quiz length plus four if larger (default length: eight). Each question has text, topic, difficulty and an ideal answer.
2. Difficulty starts at easy, rises after a correct answer and falls after an incorrect one.
3. The learner answers by typing or speaking, and Gemini judges the meaning and gives feedback.
4. A final report shows the percentage score, strong and weak topics, and a per-question record. Incorrect questions can be retired.

The report is held only for the sessions. Because a language model writes and grades the questions, errors are possible [9], [10], and its grading was not compared with human grading section (section12).

Teach-Back: Learning by teaching underlines conversational tutoring systems [20]. In NEXIS

1. The learner asks about the active context (up to 12,000 characters), and Gemini explains it in the chosen language.
2. The learner explains it back, typed or spoken; spoken capture is silent and does not interrupt.
3. Gemini evaluates conceptual understanding, including across languages, and returns a 0 to10 score with strengths, missing concepts, corrections and an improved explanation. The result is not saved.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

Participants in one study ranked teaching a virtual agent as their preferred setup [27], so whether learners accept explaining to an AI is open and was not addressed in our study.

Voice answers: The Gemini Live API transcribes spoken answers. A language-consistency step then cleans and rephrases the transcript (section 6). Since this step is also a language model, the graded text may differ from what the learner said.

Text actions and summaries: From the results panel, learners can run explain, summarize, MCQs, viva questions. Results are saved in the insights table. Summaries are type-aware and stored with a text hash, so stale ones are detected when content changes; reusable results are cached in ai cache. Automatic naming asks Gemini for a two-to-six-word title, falling back to the first meaningful words if Gemini is unavailable, and is saved only when the user presses Save.

Pomodoro: The timer uses a 25-minute focus phase, a 5-minute short break and a 15-minute long break after four focus phases. Users can start, pause, reset, skip and customize focus and short break lengths. In a study of university students, self-chosen breaks were linked to more fatigue and distraction and lower concentration and motivation than systematic breaks, with no difference inn task completion [28]. This motivates a structured schedule but was not tested on NEXIS. Pomodoro works offline.

Context limits: Each feature receives a different amount of context. Only the voice session uses the full context, all others use the beginning of the text, with no chunking or retrieval, so they miss the rest of long documents.

Table 3: **Context supplied to each feature**

Feature	Model	Context used
Voice conversation	Gemini Live	Full active context
Context switch mid-session	Gemini Live	First 8,000 characters of the new context
Quiz-Me	Gemini text	First 14,000 characters
Teach-Back	Gemini text	First 12,000 characters
Summaries, actions, questions	Gemini text	First 16,000 characters
Automatic naming	Gemini text	First 8,000 characters
Pomodoro	None (local)	Not applicable

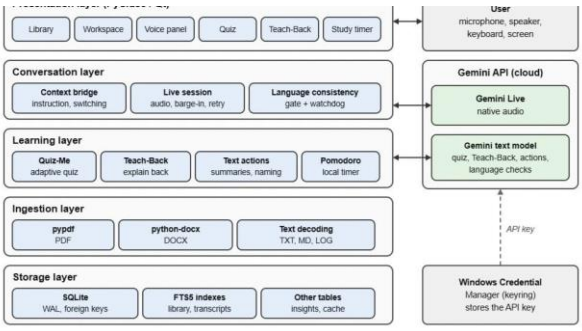


Fig. 2. Quiz-Me loop (context, question bank, question, answer, grading, difficulty update, report) and Teach-Back loop (question, AI explanation, learner's evaluation, feedback).

9. EXPERIMENTAL EVALUATION

We conducted a preliminary, exploratory user study to examine whether first-time users could learn and use NEXIS with a brief introduction, and what they found useful or difficult. The study was qualitative, with no control condition and no timed tasks. Its aim was early feedback, not performance measurement.

Participants: Ten third-year undergraduate students from the BCA and Engineering departments took part. They were recruited through the team's personal contacts and acquaintances (convenience sampling). Participation was voluntary. Verbal consent was obtained, and no names or personal data were recorded. Participants were told that their voice

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

and the document text are sent to Google's Gemini service during use.

Materials: Each participant received the NEXIS executable (version 1.1, build date 8 September 2026) and installed it on their own Windows machine. Participants supplied their own Gemini API key (bring-your-own-key) and entered it in the application's settings. They used their own study notes as documents.

Procedure: A team member gave a brief verbal tutorial covering the main features. Participants then explored NEXIS on their own for about 50 minutes without fixed tasks. They created or imported contexts, held voice conversations about them, and tried Quiz-Me, Teach-Back and the Pomodoro timer.

They also deleted a context and uploaded it again, and changed the conversation language during a session. Languages used were English, Hindi, Kannada, Urdu.

Data collection and analysis: After the session, a team member collected face-to-face feedback using open questions such as "What did you find useful?" and "What was difficult or confusing?" Responses were noted by a team member during the conversation. We grouped the comments into themes and counted how many participants raised each one. We collected no numerical ratings, timing data or answer-accuracy data, and applied no statistical tests.

10. LIMITATIONS AND FUTURE WORK

Limitations:

Context handling. NEXIS does not chunk, embed or retrieve. Only the voice session sees the full context; other features read just the first 8,000 to 16,000 characters, so they miss the rest of long documents. Models may also use information in the middle of long inputs less reliable [5], and long-context prompting costs more than retrieval [4]. There are no passage-level citations, so answers are hard to verify [7]. Grounding reduces hallucination but does not eliminate.

Text extraction: Documents are converted using pypdf python-docx and plain-text decoding. Scanned PDFs, complex tables and figures are not handled, and poor extraction silently weakens grounding [6]. Supported formats are PDF, DOCX, TXT, Markdown and log files (50 MB limit).

Cloud dependence: NEXIS needs internet and the user's own Gemini API key, and audio and context are sent to the provider. Latency was not measured. Quota errors end a session, and a model alias may change behavior over time.

AI-judged features: Quiz-Me and Teach-Back grading can be wrong [9], [10] and was not compared with human grading. Transcripts are rephrased by a model, so stored text may differ from what was said. Results are not saved, so there is no learning history.

Evaluation scope: Ten participants from one institution gave informal feedback. We measured no latency, accuracy or learning outcomes, so results are early indications only [22].

Future work:

Hybrid context handling: add retrieval for large libraries, choosing by size and cost [4].

1. **Attributed answers:** passage-level citations [7] and automatic evaluation [24], [25].
2. **Learning history:** Store results, track weak topics, schedule repeat testing [26], [30].
3. **Security:** encrypt the database; consider an offline mode [15], [16], [18].
4. **Extraction:** OCR and the table parsing.
5. **Larger evaluation:** per-language measurements and learning outcomes over weeks [22], [23] [29]
6. **Portability:** Other desktops and Android (requested by four participants).
7. **Saved audio explanations:** repayable labelled recordings (suggested by the five participants).
8. **Physical-note learners:** printable summaries and quizzes.

11. CONCLUSION:

Studying across several documents usually means searching, switching files and re-explaining context for every follow-up question. This paper presented NEXIS, a Windows desktop environment that

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

replaces that cycle with spoken, multilingual conversation grounded in a user-built library. NEXIS supplies the learner's active context to the Gemini Live API in full, supports interruption and automatic reconnection, and stores documents and sessions locally in SQLite with full-text search, while keeping the API key in the Windows Credential Manager. On the same context, it adds Quiz-Me self-checks, Teach-Back coaching and Pomodoro focus sessions.

Our design deliberately avoids chunking and retrieval. For a personal library, this removes a class of retrieval failures [6] and relies on the long-context abilities of current models [4], [13], [14], at the cost of higher context use and limits on very large documents. In a preliminary study, ten undergraduate students used NEXIS for about 50 minutes after a brief introduction. All ten found its repeated doubt clearing, while participants also asked for mobile support and saved audio explanations, and some preferred traditional study. The study was small and informal, so we treat these results as early indications.

NEXIS shows that real-time conversational AI, local context storage and structured study workflows can be combined in a single desktop tool. Its limits, especially cloud dependence, the lack of source citations and the partial use of long documents by some features, together with the quantitative evaluation still to be done, define a clear agenda for future work.

REFERENCES:

[1] F. Biwer, W. Wiradhany, M. G. A. oude Egbrink, and A. B. H. de Bruin, "Understanding effort regulation: Comparing 'Pomodoro' breaks and self-regulated breaks," *British Journal of Educational Psychology*, vol. 93, no. S2, pp. 353–367, Aug. 2023, doi: 10.1111/bjep.12593.

[2] A. Debbané, K. J. Lee, J. Tse, and E. Law, "Learning by Teaching: Key Challenges and Design Implications," *Proc ACM Hum Comput Interact*, vol. 7, no. CSCW1, Apr. 2023, doi: 10.1145/3579501.

[3] Y. Chu *et al.*, "Qwen2-Audio Technical Report," Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.10759

[4] OpenAI *et al.*, "GPT-4o System Card," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.21276

[5] Gemini Team *et al.*, "Gemini: A Family of Highly Capable Multimodal Models," May 2025, [Online]. Available: http://arxiv.org/abs/2312.11805

[6] Z. Xie and C. Wu, "Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming," Nov. 2024, [Online]. Available: http://arxiv.org/abs/2408.16725

[7] Q. Fang, S. Guo, Y. Zhou, Z. Ma, S. Zhang, and Y. Feng, "LLaMA-Omni: Seamless Speech Interaction with Large Language Models," Mar. 2025, [Online]. Available: http://arxiv.org/abs/2409.06666

[8] G. Comanici *et al.*, "Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities," Dec. 2025, [Online]. Available: http://arxiv.org/abs/2507.06261

[9] Gemini Team *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," Dec. 2024, [Online]. Available: http://arxiv.org/abs/2403.05530

[10] R. Schmucker, M. Xia, A. Azaria, and T. Mitchell, "Ruffle&Riley: Towards the Automated Induction of Conversational Tutoring Systems," Nov. 2023, [Online]. Available: http://arxiv.org/abs/2310.01420

[11] A. Défossez *et al.*, "Moshi: a speech-text foundation model for real-time dialogue," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.00037

[12] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.04356

[13] N. F. Liu *et al.*, "Lost in the Middle: How Language Models Use Long Contexts," Nov. 2023, [Online]. Available: http://arxiv.org/abs/2307.03172

[14] W. Shi *et al.*, "REPLUG: Retrieval-Augmented Black-Box Language Models," May 2023, [Online]. Available: http://arxiv.org/abs/2301.12652

[15] J. Wei *et al.*, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Jan. 2023, [Online]. Available: http://arxiv.org/abs/2201.11903

[16] L. Huang *et al.*, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Trans Inf Syst*, vol. 43, no. 2, Jan. 2025, doi: 10.1145/3703155.

[17] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.11511

[18] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2312.10997

[19] G. Izacard *et al.*, "Atlas: Few-shot Learning with Retrieval Augmented Language Models," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2208.03299

[20] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.02155

[21] J. Glaser and T. Richter, "The Testing Effect in the Lecture Hall: Does it Transfer to Content Studied but Not Practiced?," *Teaching of Psychology*, vol. 52, no. 1, pp. 17–26, Jan. 2025, doi: 10.1177/00986283231218943.

[22] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling Large Language Models to Generate Text with Citations," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2305.14627

[23] J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking Large Language Models in Retrieval-Augmented Generation," Dec. 2023, [Online]. Available: http://arxiv.org/abs/2309.01431

[24] D. Zhang *et al.*, "SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities," May 2023, [Online]. Available: http://arxiv.org/abs/2305.11000

[25] A. Brasoveanu, M. Moodie, and R. Agrawal, "Textual evidence for the perfunctoriness of independent medical

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 11, CONFLUENCE - 2026

reviews," in *CEUR Workshop Proceedings*, CEUR-WS, 2020, pp. 1–9. doi: 10.1145/nnnnnnn.nnnnnnn.

[26] J. Pavão, R. Bastardo, A. G. Silva, and N. P. Rocha, "Usability and User Experience Assessment of Health Care Conversational Agents Using Validated Subjective Instruments: Systematic Review and Comparative Analysis," *JMIR Hum Factors*, vol. 13, 2026, doi: 10.2196/102911.

[27] Z. Ji *et al.*, "Survey of Hallucination in Natural Language Generation," Jul. 2024, doi: 10.1145/3571730.

[28] W. Fan *et al.*, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2405.06211

[29] P. K. Rubenstein *et al.*, "AudioPaLM: A Large Language Model That Can Speak and Listen," Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.12925

[30] Y. An, J. Liu, N. Acharya, and R. Hashmi, "Enhancing Student Learning with LLM-Generated Retrieval Practice Questions: An Empirical Study in Data Science Courses," Jul. 2025, [Online]. Available: http://arxiv.org/abs/2507.05629

(This work is licensed under a Creative Commons Attribution 4.0 International License.)