TACI BULLETIN · SCIENTIFIC FORUM · RESEARCH ARTICLE

A Memory-Augmented, Retrieval-Grounded Large Language Model Framework for Personalized Digital Assistance in Higher Education

Dr. Maria Aslam

Department of Computer Science, Tahseen Ahmad Cheema Institute (TACI), Bahawalpur, Pakistan
Corresponding author: maria.aslam@taci.edu.pk · ORCID: 0000-0002-5376-013X

ABSTRACT

Large language models (LLMs) have made conversational digital assistants broadly capable, yet most deployments remain generic: they forget users between sessions, answer from parametric knowledge that may be outdated or fabricated, and seldom act on the documents and routines of a specific workplace. These limitations are acute in higher-education institutions, where faculty, students and administrators combine academic writing, research reading, correspondence and scheduling, frequently across English, Urdu and Roman Urdu. This paper proposes a memory-augmented, retrieval-grounded LLM framework for a personalized digital assistant that addresses three gaps identified in the literature: persistent yet user-controlled personalization, grounding in institutional and personal documents, and safe execution of everyday tasks. The framework integrates seven layers: multimodal input, intent and context management, a consent-governed personal memory store, a retrieval-augmented knowledge layer, an LLM reasoning and tool-orchestration engine, a guardrail layer for grounding and privacy, and a text-and-voice output layer, all operating under an explicit governance layer. A relevance function ranks stored memories by semantic similarity, recency and importance before they are composed into the model context. The design is examined through five representative academic and administrative scenarios, and a multi-dimensional evaluation protocol is specified covering task success, factual grounding, personalization fidelity, efficiency, usability and privacy leakage. The framework offers institutions a practical blueprint for assistants that are personal, verifiable and accountable, and establishes the basis for empirical evaluation in a Pakistani higher-education setting.

Keywords: *large language models; personalized assistant; retrieval-augmented generation; long-term memory; natural language processing; higher education; human–computer interaction*

1. Introduction

The transformer architecture [1] and the subsequent scaling of language models [2] have transformed natural language processing from a set of narrow tasks into a general capability. Instruction tuning with human feedback [3] made these models usable as conversational assistants that can answer questions, draft text, summarize documents and reason over multi-step problems, and recent surveys document the speed and breadth of this progress [4], building on decades of foundational work in speech and language processing [19].

Education and academic administration are among the domains with the most to gain. LLMs can support writing, explanation, feedback and routine correspondence, while also raising concerns about accuracy, bias, privacy and over-reliance [5]. In practice, three limitations prevent current assistants from becoming dependable personal aides. First, most assistants are effectively stateless: personalization is confined to what fits in the current prompt, so preferences, ongoing projects and past decisions must be restated repeatedly [6], [7]. Second, answers drawn from the model's parametric knowledge can be outdated or fabricated, a phenomenon widely documented as hallucination [8]. Third, being useful often requires *acting* (sending an email, setting a reminder, generating a document), which calls for controlled integration with external tools [9], [10].

These limitations are sharper in higher-education institutions in developing countries. Faculty frequently combine teaching, research and administrative duties; institutional knowledge lives in circulars, policies and templates rather than on the open web; users switch between English, Urdu and Roman Urdu; and expectations

of privacy are high because conversations involve students, colleagues and unpublished work. A generic assistant that neither remembers its user nor knows the institution, and that cannot show where its answers come from, is poorly matched to this environment.

This paper addresses that mismatch by proposing a memory-augmented, retrieval-grounded LLM framework for personalized digital assistance in higher education. Its contributions are as follows:

(1) A structured review of LLM personalization, long-term memory, retrieval grounding and tool use, culminating in an explicit statement of the research gap (Section 2).

(2) A seven-layer framework that unifies consent-governed long-term memory, document retrieval and tool orchestration under guardrails and a governance layer (Section 3).

(3) A memory-relevance formulation, a context-composition scheme and a request-processing algorithm that make the design reproducible (Sections 3.3–3.4).

(4) A scenario-based design analysis and a multi-dimensional evaluation protocol for empirical validation (Section 4).

Section 5 discusses implications, risks and limitations, and Section 6 concludes the paper.

2. Literature Review

2.1 Foundations of large language models

Self-attention allowed sequence models to be trained in parallel at unprecedented scale [1]. Brown et al. showed that a sufficiently large model can perform new tasks from a handful of examples supplied in the prompt [2], and Ouyang et al. demonstrated that reinforcement learning from human feedback aligns model outputs more closely with user intent, producing the instruction-following behaviour on which today's assistants rely [3]. Zhao et al. survey the resulting landscape of pre-training, adaptation, utilization and evaluation [4]. These advances, however, describe general-purpose models; they do not by themselves make an assistant aware of a particular user or institution.

2.2 Personalization of LLMs

Chen et al. review how LLMs are being adapted to individual users and identify the move from passive, recommendation-style personalization to assistants that converse, plan and act on a user's behalf, while noting open challenges in long-term user modelling and privacy [6]. The LaMP benchmark formalizes personalized text generation and classification and shows that retrieving relevant items from a user's profile and adding them to the prompt improves personalized outputs [7]. Prompt-based personalization is nevertheless bounded by the context window and offers the user little visibility or control over what the system "knows" about them.

2.3 Long-term memory for conversational agents

Several architectures extend LLMs with memory that persists beyond a single context window. MemGPT manages a hierarchy of memory tiers inspired by operating-system virtual memory, letting the model page information in and out as needed [11]. MemoryBank stores past interactions and user portraits and applies a forgetting mechanism inspired by the Ebbinghaus curve to keep memory relevant [12]. Park et al. retrieve memories for generative agents by combining relevance, recency and importance [13], an idea adopted in this paper's memory-ranking function. These systems demonstrate the value of persistent memory but are general-purpose and give limited attention to user consent, editability and institutional deployment.

2.4 Retrieval grounding and hallucination

Retrieval-augmented generation (RAG) conditions the model on passages retrieved from an external corpus, improving factuality on knowledge-intensive tasks and allowing knowledge to be updated without retraining [14]. Gao et al. survey the rapid evolution of RAG pipelines, including retrieval, augmentation and evaluation strategies [15]. Ji et al. show that hallucination remains a pervasive failure mode of natural language generation and that grounding in verifiable sources is among the most effective mitigations [8]. For an institutional assistant, retrieval over circulars, policies, templates and the user's own papers is therefore essential.

2.5 Tool use and agentic behaviour

Toolformer shows that language models can learn when and how to call external APIs [9], and ReAct interleaves reasoning traces with actions and observations, improving performance and interpretability on multi-step tasks [10]. These techniques underpin assistants that can draft and send messages or manage calendars, but they also introduce risk: an action taken on a wrong assumption has real consequences, which motivates human confirmation for consequential operations.

2.6 LLMs in education, and privacy

Kasneci et al. analyse the opportunities of LLMs for learners and educators alongside risks including inaccurate content, bias, privacy and dependence, and they argue for human oversight and competence building [5]. On the privacy side, Carlini et al. demonstrate that training data, including personal information, can be extracted from language models [16]. An assistant that stores personal memories must therefore minimize what it retains, let users inspect and delete it, and avoid sending sensitive data to external services unnecessarily. Speech interfaces, increasingly expected by users, are now practical through robust multilingual speech recognition [17].

2.7 Research gap

Table 1 summarizes the primary focus of representative work. Taken together, the literature reveals four gaps that this paper addresses:

- **G1, persistent and controllable personalization.** Memory architectures [11]–[13] and profile retrieval [7] personalize responses, but users rarely have explicit control over what is remembered, for how long and with what consent.
- **G2, unified grounding, memory and action.** Retrieval [14], [15], memory [11], [12] and tool use [9], [10] are mostly studied separately; few designs combine them in one assistant with guardrails between reasoning and action.
- **G3, institutional and multilingual context.** Existing systems are evaluated on general or English-only tasks, whereas higher-education users in Pakistan work with local institutional documents and switch between English, Urdu and Roman Urdu.
- **G4, holistic evaluation.** Evaluations typically report a single dimension such as accuracy; protocols that jointly measure task success, grounding, personalization, usability and privacy are scarce.

Table 1. Primary focus of representative approaches compared with the proposed framework (✓ primary focus; ~ partially addressed; – not addressed).

Approach	Long-term user memory	Document grounding	Task / tool execution	Consent & privacy controls	Higher-education context
GPT-3, InstructGPT [2], [3]	–	–	–	–	–
LaMP [7]	~	~	–	–	–
MemGPT [11]	✓	~	~	–	–
MemoryBank [12]	✓	–	–	–	–
RAG [14], [15]	–	✓	–	–	–
Toolformer, ReAct [9], [10]	–	~	✓	–	–
Kasneci et al. [5]	–	–	–	~	✓
Proposed framework	✓	✓	✓	✓	✓

3. Proposed Framework

3.1 Design requirements

Six requirements, derived from the gaps above and from the needs of academic and administrative users, guide the design:

- **R1 Personalization:** remember stable preferences and ongoing work across sessions, under user control.
- **R2 Verifiability:** ground factual answers in retrievable sources and cite them.
- **R3 Safe action:** execute tasks through tools, with explicit confirmation before consequential actions.
- **R4 Accessibility:** accept text and voice and handle English, Urdu and Roman Urdu input.
- **R5 Privacy and accountability:** minimize stored data, allow inspection and deletion, and log every action.
- **R6 Deployability:** run on hosted or open-weight models with modest institutional infrastructure.

3.2 System architecture

Figure 1 presents the architecture. A request enters through the multimodal input layer (L1), which transcribes speech and detects language. The intent and context manager (L2) classifies the request, decomposes multi-step tasks and maintains session state. The reasoning engine (L5) then draws on three resources: the personal memory store (L3), the knowledge-retrieval layer (L4) and a tool layer for email, calendar and document generation. Every candidate response passes through guardrails (L6) before reaching the output layer (L7), and a governance layer spanning the system provides consent management, audit logging and human confirmation. Table 2 summarizes the role of each layer.

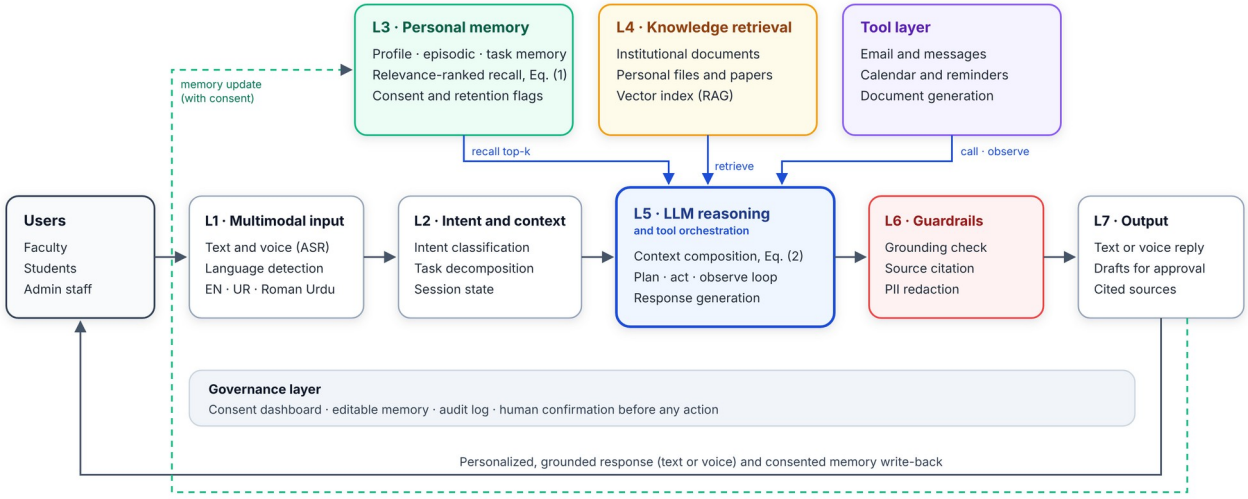

Figure 1. Architecture of the proposed memory-augmented, retrieval-grounded personalized assistant. Memory (L3), knowledge retrieval (L4) and tools feed a single reasoning engine (L5); guardrails (L6) and a governance layer control what is returned, acted upon and remembered.

Table 2. Layers of the proposed framework, their functions and representative techniques.

Layer	Function	Representative techniques
L1 Multimodal input	Capture text and voice; detect language	Speech recognition [17]; language identification; Roman Urdu normalization
L2 Intent and context	Classify intent; decompose tasks; track session	Prompted intent classification; dialogue state tracking
L3 Personal memory	Store and recall profile, episodic and task memory	Embedding store; relevance ranking, Eq. (1); consent and retention flags
L4 Knowledge retrieval	Ground answers in institutional and personal documents	Chunking and vector index; RAG [14], [15]
L5 Reasoning and orchestration	Compose context; plan, act and observe; generate response	Instruction-tuned LLM [3]; ReAct-style loop [10]; function calling [9]
L6 Guardrails	Verify grounding; cite sources; protect privacy	Claim-to-source matching [8]; PII detection and redaction
L7 Output	Deliver text or speech; present drafts for approval	Text-to-speech; approval workflow

Layer	Function	Representative techniques
Governance	Consent, transparency and accountability	Memory dashboard; audit log; confirmation gates

3.3 Personal memory model

The memory store holds three types of item: *profile memory* (stable preferences such as writing style, citation format or preferred language), *episodic memory* (summaries of past interactions and decisions) and *task memory* (ongoing projects, deadlines and templates). Each memory item m carries an embedding e_m, a timestamp t_m, an importance score $\iota_m \in [0, 1]$ assigned at write time, and a consent flag $c_m \in \{0, 1\}$. Following the retrieval principle of Park et al. [13], the relevance of item m to query q at time t is

$$S(m \mid q, t) = \alpha \cdot \cos(e_q, e_m) + \beta \cdot \exp(-\lambda(t - t_m)) + \gamma \cdot \iota_m \tag{1}$$

where $\alpha + \beta + \gamma = 1$ weight semantic similarity, recency and importance, and λ controls the rate of decay. Only items with $c_m = 1$ are eligible, and the top-k items by S form the recalled set M_k. Items whose relevance stays below a threshold for longer than a retention period T are archived, and the user may view, edit or delete any item through the governance dashboard, which addresses requirement R5 and gap G1.

The context passed to the model is then composed as

$$C = [\, \Pi \,;\, P_u \,;\, M_k \,;\, D_k \,;\, H_r \,;\, q \,] \tag{2}$$

where Π is the system policy, P_u the user profile summary, M_k the recalled memories, D_k the top-k retrieved document passages, H_r the recent dialogue turns and q the current request. Ordering policy first and the request last keeps the instructions stable while giving the model the most recent information at the point of generation.

3.4 Request-processing algorithm

Algorithm 1 specifies how a single request is processed, combining grounding, memory and tool use with explicit checkpoints for safety.

Algorithm 1. *Personalized, grounded request processing*

Input: request q (text or voice) from user u; **Output:** grounded response or approved action

```
if q is speech then q ← ASR(q); detect language ℓ and normalize Roman Urdu
intent, subtasks ← IntentManager(q, session state)
M_k ← top-k memories by S(m | q, t), Eq. (1), with c_m = 1
D_k ← top-k passages retrieved from institutional and personal corpora
C ← compose context, Eq. (2)
for each subtask do
    plan next step with LLM(C); if a tool is required then
        if action is consequential (send, schedule, publish) then request user confirmation
        execute tool; append observation to C
r ← LLM(C)
Guardrails: verify each factual claim in r against D_k; attach citations; flag unsupported claims; redact PII
deliver r in language ℓ as text or speech
propose memory updates; store only items the user permits (c_m = 1); write audit log
```

3.5 Guardrails, privacy and governance

Guardrails operate on every response. Factual statements are matched to retrieved passages; supported statements are cited, and unsupported ones are either removed or clearly marked as the model's own suggestion, reducing the risk of hallucination [8]. Personally identifiable information is detected and redacted before any call to an external model service, mitigating the leakage risks documented by Carlini et al. [16]. Consequential actions, such as sending an email or creating a calendar event, are always presented as drafts for approval. A governance dashboard lets the user inspect, correct or delete what the assistant remembers, and an audit log records every action, supporting institutional accountability.

3.6 Implementation considerations

The framework is model-agnostic. It can be realized with a hosted instruction-tuned LLM accessed through an API or with an open-weight model served on institutional hardware, a vector database for memory and documents, a speech-recognition model for voice input [17], and connectors to the institution's email and calendar services. Delivered as web and mobile clients over a single API, the design suits institutions with modest infrastructure, consistent with requirement R6.

4. Design Evaluation

4.1 Scenario-based analysis

To examine how the framework behaves across typical workloads, five representative scenarios were analysed (Table 3). Each scenario traces which layers are engaged and how personalization changes the outcome compared with a generic assistant.

Table 3. Representative academic and administrative scenarios, layers engaged and the effect of personalization.

Scenario	Example request	Layers engaged	Effect of personalization and grounding
S1 Formal correspondence	"Draft an email to the Dean requesting approval for next week's workshop."	L2, L3, L5, Tools, L6, L7	Applies the user's usual tone and signature; recalls workshop details from task memory; sent only after approval
S2 Voice reminder	Voice note in Roman Urdu: "Kal 10 baje faculty meeting yaad dilana."	L1, L2, L5, Tools, L7	Transcribes and normalizes Roman Urdu; creates a calendar reminder at the user's confirmation
S3 Research summarization	"Summarize this paper and compare it with the two I read last week."	L3, L4, L5, L6	Retrieves the earlier papers from episodic memory; cites passages; flags unsupported comparisons
S4 Academic writing	"Improve this paragraph and format the references."	L3, L5, L6	Uses the stored citation style and discipline-specific vocabulary; preserves the author's voice
S5 Administrative document	"Prepare a notice for the mid-term schedule."	L4, L5, Tools, L6, L7	Retrieves the institutional template and current calendar; generates a draft notice for review

Across the scenarios, memory removes the need to restate context (S1, S3, S4), retrieval ties answers to institutional or scholarly sources (S3, S5), and the confirmation gate keeps the user in control of actions (S1, S2, S5). The scenarios also show that the layers are complementary: none of the five could be served well by memory, retrieval or tool use alone, which supports the integrated design called for by gap G2.

4.2 Evaluation protocol

To address gap G4, Table 4 defines a multi-dimensional protocol for empirical validation. A within-subject study is proposed with faculty, students and administrative staff, comparing the full framework against an ablated baseline that uses the same LLM without memory and retrieval. Personalization is assessed in the spirit of LaMP [7], and usability with the System Usability Scale [18].

Table 4. Proposed evaluation dimensions, metrics and methods.

Dimension	Metric	Method
Task success	Completion rate; expert quality rating (1–5)	Scripted tasks drawn from scenarios S1–S5
Factual grounding	Citation precision and recall; unsupported-claim rate	Manual verification of claims against sources [8]
Personalization fidelity	Preference-adherence rate; pairwise preference over baseline	Profile-conditioned tasks, LaMP-style comparison [7]
Efficiency	Time on task versus manual work; 95th-percentile latency	Timed sessions and system logs

Dimension	Metric	Method
Usability	System Usability Scale score	Post-session questionnaire [18]
Privacy and safety	PII leakage rate; consent-compliance rate; unconfirmed-action count	Red-team prompts and audit-log review [16]

5. Discussion

The framework reframes a personalized assistant as an accountable system rather than a single model. Persistent memory makes the assistant progressively more useful, retrieval makes its answers checkable, and tools make it able to act, yet each capability introduces a matching risk. Memory can become stale or over-fitted to past behaviour, retrieval can surface the wrong passage, and tools can execute an action on a mistaken premise. The guardrail and governance layers are the design's answer to these risks: they keep the user informed of what the assistant knows, where its answers come from and what it is about to do.

Several trade-offs merit attention in deployment. Richer context improves relevance but increases cost and latency; the relevance function of Eq. (1) and a fixed k bound this growth. Aggressive personalization may narrow the information a user encounters, so the system should distinguish preferences of style from preferences of substance. In academic settings, the assistant must support rather than replace authorship: drafts are suggestions, sources must be verified, and institutional policies on academic integrity apply [5].

Limitations. The present work contributes a framework, formal components and an evaluation protocol rather than experimental results; its effectiveness must be confirmed through the study outlined in Section 4.2. Performance will also depend on the underlying model and on the quality of speech and language resources for Urdu and Roman Urdu, which remain more limited than for English.

6. Conclusion and Future Work

This paper proposed a memory-augmented, retrieval-grounded LLM framework for personalized digital assistance in higher education. Building on a review of LLM foundations, personalization, long-term memory, retrieval and tool use, it identified four gaps and responded with a seven-layer architecture, a memory-relevance function, a context-composition scheme, a request-processing algorithm and explicit guardrail and governance mechanisms. Scenario analysis showed how the layers combine to serve real academic and administrative tasks, and a multi-dimensional protocol was defined for empirical evaluation.

Future work will implement the framework as web and mobile clients for faculty, students and staff at TACI, carry out the proposed user study, and extend language support for Urdu and Roman Urdu, including speech. Further directions include on-device processing of sensitive memories, adaptive tuning of the weights α, β and γ from user feedback, and integration with institutional information systems.

Declarations

Author contributions: M. Aslam conceived the framework, conducted the literature review and wrote the manuscript.

Conflict of interest: The author declares no conflict of interest. The author serves as an Associate Editor of TACI Bulletin and was not involved in the editorial handling of this manuscript.

Funding: This research received no external funding.

Data availability: No new data were generated in this study.

Ethics: The user study proposed in Section 4.2 will be conducted with informed consent and institutional approval.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.

[4] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou *et al.*, "A survey of large language models," arXiv:2303.18223, 2023.

[5] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, art. 102274, 2023, doi: 10.1016/j.lindif.2023.102274.

[6] J. Chen, Z. Liu, X. Huang, C. Wu, Q. Liu, G. Jiang *et al.*, "When large language models meet personalization: Perspectives of challenges and opportunities," *World Wide Web*, vol. 27, no. 4, art. 42, 2024, doi: 10.1007/s11280-024-01276-1.

[7] A. Salemi, S. Mysore, M. Bendersky and H. Zamani, "LaMP: When large language models meet personalization," in *Proc. 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024, doi: 10.18653/v1/2024.acl-long.399.

[8] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu *et al.*, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, art. 248, 2023, doi: 10.1145/3571730.

[9] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro *et al.*, "Toolformer: Language models can teach themselves to use tools," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[10] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. International Conference on Learning Representations (ICLR)*, 2023.

[11] C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica and J. E. Gonzalez, "MemGPT: Towards LLMs as operating systems," arXiv:2310.08560, 2023.

[12] W. Zhong, L. Guo, Q. Gao, H. Ye and Y. Wang, "MemoryBank: Enhancing large language models with long-term memory," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 19724–19731, 2024, doi: 10.1609/aaai.v38i17.29946.

[13] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proc. 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023, doi: 10.1145/3586183.3606763.

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[15] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi *et al.*, "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.

[16] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee *et al.*, "Extracting training data from large language models," in *Proc. 30th USENIX Security Symposium*, pp. 2633–2650, 2021.

[17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. 40th International Conference on Machine Learning (ICML)*, PMLR 202, pp. 28492–28518, 2023.

[18] J. Brooke, "SUS: A 'quick and dirty' usability scale," in P. W. Jordan, B. Thomas, B. A. Weerdmeester and I. L. McClelland (eds.), *Usability Evaluation in Industry*, London: Taylor & Francis, pp. 189–194, 1996.

[19] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. (draft), Stanford University, 2024.

About the Author

Dr. Maria Aslam is Assistant Professor and Head of the Department of Computer Science, and Director IT, at Tahseen Ahmad Cheema Institute, Bahawalpur. She received her PhD in Computer Science from The Islamia University of Bahawalpur in 2025, and her MS and BS in Computer Science with Gold Medals. She worked as a research assistant on the Erasmus+ SAFE-RH project with the University of the West of Scotland. Her research spans healthcare AI, large language model applications, AI agents and GPU computing, with publications in IEEE Access, Scientific Reports and Frontiers in Pharmacology. She is a PHEC-nominated AI Expert and Trainer, a peer reviewer for Frontiers in Microbiology, and founder of MarXus AI Nexus.