IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151065

A Voice-First Personal Finance Assistant Using Tool Calling and a Locally Served Fine-Tuned Small Language Model

Rishi Raj Sharma[1], MD Faiz[2], S. Madan[3], Tejashwini[4], Udaya Shankar Budagavi[5]

7th Sem B.E.(CS&AI), Ballari Institute of Technology and Management (BITM), Ballari, Karnataka-583104, India[1-4]

Assistant Professor, Department of Computer Science and Engineering - Artificial Intelligence, Ballari Institute of Technology and Management (BITM), Ballari, Karnataka 583104, India[5]

Abstract: Large language model assistants can answer questions about personal finances, but can also make mistakes, upload sensitive data or run actions without human approval. FIN-AI is a voice-first personal finance assistant for Indian users with local data storage, rupee amounts, loan instalments, and lakh/crore units. A model is only used to select a tool, calculations are carried out by Python, and answers are phrased with templates; model-proposed modifications are confirmed via Proceed cards. Queries follow a three-tier pipeline: offline intent classifier, a locally hosted Qwen3-1.7B with QLoRA, and cloud LLMs as fallback. Training lessons are created from the real tool registry; the first model was fine-tuned on a 4 GB laptop GPU and later ones on a free cloud T4. In 67 hand-written questions, tool-selection accuracy increased from 71.6% (untuned) to 91.0% after the first fine-tuning (exact McNemar p = 0.0002); the final version scored 89.6% (Wilson 95% interval 80.0–94.8%). The fast path handled 98 of 105 read-only questions offline. Development evaluations are presented, and the data sent to external AI, speech, and market-data services is detailed.

Keywords: Voice assistant, Personal finance, Tool calling, Small language model, QLoRA, Query routing.

I. INTRODUCTION

For many people, dealing with everyday money is hard: in a widely quoted survey, just about 30% of those in the U.S. could answer three simple questions about compounding, inflation and diversification correctly [1]. Meanwhile, the proportion of adults in developing economies making or receiving digital payments grew from 35% in 2014 to 57% in 2021, while in India over 80 million adults made their first digital merchant payment during COVID-19 [2]. Spoken interfaces can help: with 58 non-literate and semi-literate adults, no one finished a money transfer using a text interface, and a spoken-dialog interface required less than half the time of a rich-multimedia one [3]. Indian users must be supported with assistance for amounts spoken in lakh/crore, loan instalments (equated monthly instalments, EMIs), Indian English voices and Hindi-English (Hinglish) confirmations like “haan ji”.

Such an assistant is easy to create with large language models (LLMs) but difficult to trust. LLMs can generate cohesive but inaccurate text [4]; prior to fine-tuning, the 1.7 billion parameter model used in this work answered “The price of Reliance is currently 1250 INR.” without invoking any tool. Unlike cloud-only assistants that send personal records off the device, small language models can be fine-tuned at low cost and run locally on a device [5].

This paper presents FIN-AI, a voice-first web application that covers income and expenses, budgets, savings goals, bills and EMIs, stock holdings (NSE, BSE and US exchanges) and currency conversion. Spoken questions are transcribed using automatic speech recognition (ASR), and answers are read aloud using text-to-speech (TTS). For data questions, the model (or the offline classifier) chooses one of 32 typed backend tools, and Python services are used to calculate the answer, while fixed templates are used to formulate the answer; the exceptions are specified in Section III-B. Questions go through an offline intent classifier, a locally served small model fine-tuned with quantized low-rank adaptation (QLoRA) and a cloud LLM; changes proposed by a model are subjected to confirmation on a confirmation card, with one exception (Section III-D). The contributions are:

- A three-tier router that combines an offline intent classifier, a locally deployed QLoRA-tuned Qwen3-1.7B model with a ranker-built tool menu, and a cloud fallback model, plus a circuit breaker that rests a failing local model (Sections III and IV);

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

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151065

- Registry-driven lesson generation and a low-cost QLoRA recipe on a 4 GB laptop GPU (which spills into system memory) and a free Colab T4 (Section IV);
- A small-sample evaluation: hand-written tool-selection accuracy rose from 48/67 untuned to 61/67 after the first fine-tune (exact McNemar p = 0.0002, although the runs also differed in setup), while later versions show no significant change on that set (Sections V and VI);
- An explicit description of the privacy boundary and of the threats to validity (Sections III-F and VII).

II. RELATED WORK

A. Assistants for Personal Finance

For mobile money, non-literate/semi-literate users strongly preferred designs that were non-text [3]. In a close published analog to FIN-AI's category model, word and character n-grams and a support vector machine [5] have been used to categorize short bank-transaction descriptions.

B. Tool Use and Grounding

Hallucination is one of the fundamental defects of LLMs, regardless of retrieval [6]. Toolformer demonstrates that language models can teach themselves to invoke APIs via simple text annotations [7], and τ-bench measures whether agents can follow complex domain policies and user constraints across multi-turn interactions [8]. FIN-AI avoids ungrounded calculation by delegating all arithmetic and database operations to deterministic Python tools.

C. Small Models and Efficient Fine-Tuning

For local data handling and low latency and light fine-tuning, small language models are increasingly used [4]. Low-Rank Adaptation (LoRA) reduces trainable parameter counts by freezing pre-trained weights and injecting low-rank decomposition matrices [9]; QLoRA quantizes the base model to 4-bit precision while retaining 16-bit adapter performance [10]. Qwen3-1.7B provides strong reasoning and tool-calling capabilities at edge-compatible model sizes [11], and TinyAgent demonstrates function-calling on device [12].

D. Query Routing

FrugalGPT cascades LLM APIs with a learned scorer, which achieves up to 98% cost reduction while matching or exceeding the accuracy of individual foundation models [13]. FIN-AI adapts this cascading approach to bridge an offline rule-and-intent classifier, a local QLoRA-adapted small model, and a cloud fallback.

E. Speech for Indian English

Trained on 680k hours of weakly supervised audio, Whisper performs well with diverse accents [14]. Evaluation on Indian-accented English benchmarks like Svarah shows that state-of-the-art ASR models still experience elevated word error rates on vernacular code-mixed phrases and regional phonetic variations [15], motivating FIN-AI's specialized vocabulary prompts and Hinglish confirmations.

F. Safety for Agents

Language model agents are prone to take risks without asking even when there are safety requirements [16]. In financial settings where transactions are irreversible, autonomous action execution poses high risk. FIN-AI addresses this through mandatory Proceed confirmation cards and defensive plan checks.

III. SYSTEM DESIGN

A. Requirements and Architecture

The following rules are observed in the system: a model selects a tool, Python calculates the answer, and a template phrases the words, with exceptions specified below. All financial databases remain strictly on the user's local machine.

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

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151065

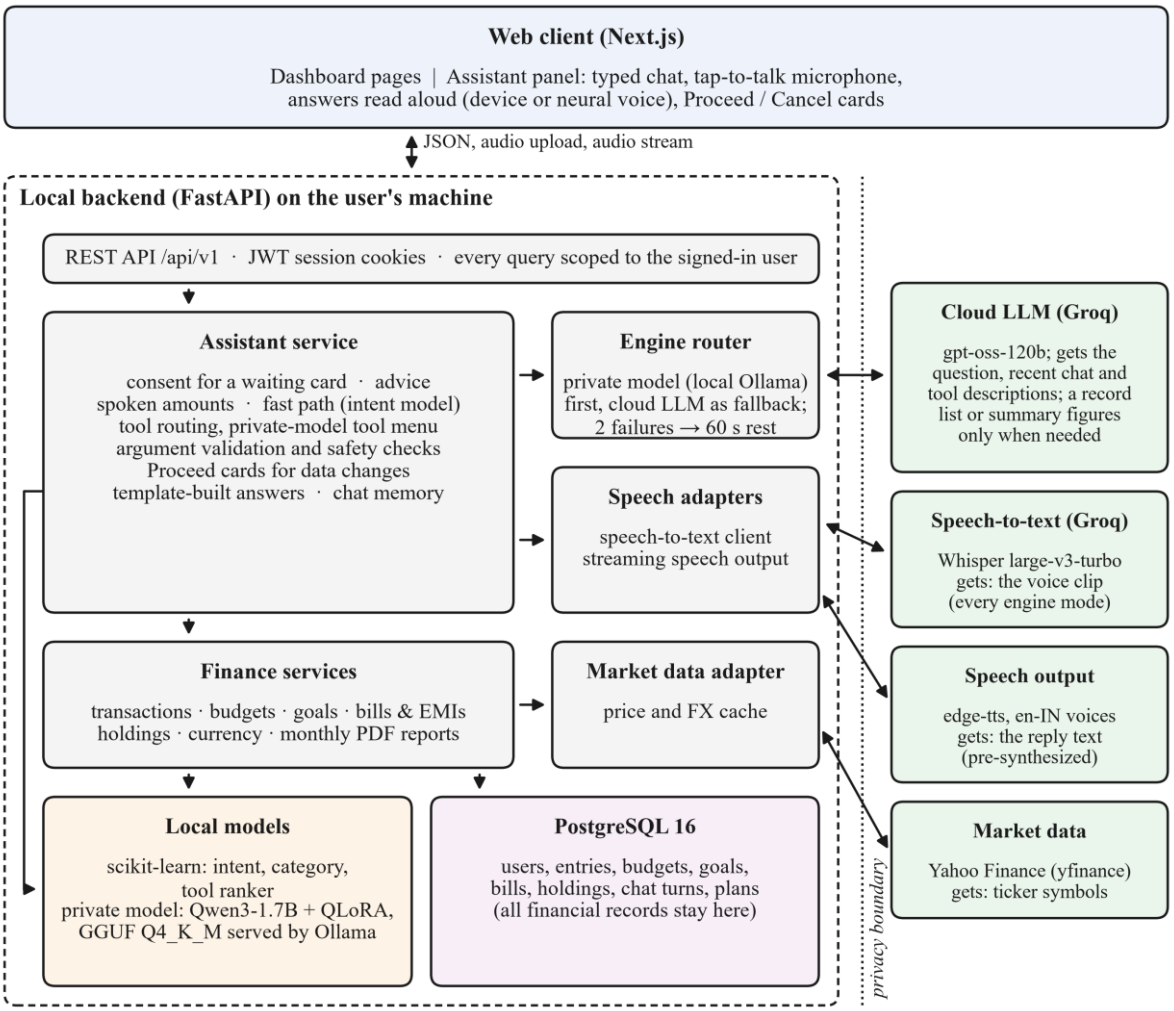


Fig. 1. FIN-AI architecture. The backend runs on the user's machine and keeps all financial records local. Voice questions are routed through an offline classifier, a locally served QLoRA-tuned small language model, or a cloud model.

B. Tools and Template-Built Answers

The money (7), plans (12), wealth (7) and changes (13) are 32 unique tools in four topic groups (some belong to two groups). Calculations are performed in Python, with answers composed of fixed templates. For instance, balance enquiries, spending summaries, and EMI breakdowns are computed directly by deterministic backend services rather than generative text.

Two kinds of model text, however, are provided. Private-model advice is only shown if it passes a groundness check against the user's real budget figures, and cloud advice is synthesized directly with conversational caution.

C. Question Routing and Engines

The sequence of the different checks is: first check for the waiting card, then the advice questions, followed by the fast path classifier. If the fast path matches, the tool executes immediately without any model call.

When in automatic mode, the cloud responds when a private reply calls a tool that is not offered or fails repeated generation checks. A circuit breaker monitors consecutive local timeouts and temporarily rests the local Ollama instance.

D. Proceed Cards and Safety Checks

The proposed changes become a series of steps to execute, with the card text generated by the backend rather than the language model. The user must explicitly inspect and click "Proceed" to commit financial state changes (e.g., logging an expense, adding a bill, setting a budget).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151065

A private model's plans are subjected to further checks, e.g. all numbers in a request must be used, amounts must not be zero or negative, and destructive operations require two-step confirmation.

E. Voice Pipeline and Conversation Memory

Recording is tap-to-talk (0.5 s to 30 s). The transcriptions are made with Groq's whisper-large-v3-turbo with a specialized finance vocabulary prompt. Speech responses are rendered through Microsoft edge-tts using Indian English voice profiles (en-IN-NeerjaNeural / en-IN-PrabhatNeural). Short-term conversational history retains up to 10 recent interactions within a 30-minute rolling window.

F. Privacy Boundary

Table I lists what leaves the machine in the submitted code. All financial records remain in the local SQLite database and are never transmitted to external servers without explicit user invocation.

TABLE I WHAT LEAVES THE MACHINE, PER OUTSIDE SERVICE, IN THE SUBMITTED CODE (OPTIONAL GOOGLE SIGN-IN CAN ALSO TRANSMIT AN AUTHENTICATION TOKEN)

Service	What is sent, and when	Modes
Groq chat, gpt-oss-120b (tool choice)	When the cloud answers: rules, date, tool descriptions, the question (≤ 500 chars), ≤ 6 chat messages from the last 30 min (may contain the user's figures), a note on the previous tool; only if a record change is likely, ≤ 8 latest entries, 6 unpaid bills and 6 goals (tagged E1/B1/G1)	Automatic, Cloud only
Groq chat (advice)	When the cloud gives advice: a snapshot of month totals, top 5 categories, bill totals, next bill, ≤ 4 goals, ≤ 4 budgets at ≥ 80% and holding cost; the question; no chat history	Automatic, Cloud only
Groq Whisper large-v3-turbo	Every voice question: the audio clip (≤ 10 MiB, ≤ 30 s) and a fixed vocabulary prompt	all, incl. Private only
Microsoft edge-tts (en-IN voices)	Every reply: its speech text (≤ 1,000 chars), pre-synthesized whatever the voice mode or switch	all
Yahoo Finance (yfinance)	Cache misses: ticker symbols and currency pairs; never quantities or buy prices	all
Ollama, private model (local)	Rules, records, history, advice snapshot, tool menu and question; stays at the default 127.0.0.1 address (not enforced)	Automatic, Private only

IV. MODELS AND TRAINING

A. Lightweight Classifiers

The models are the category, the intent and the ranker, and all three share the fast path pipeline. The category model maps transaction notes to budget categories; the intent model classifies user queries into read-only intents versus conversational prompts; and the ranker filters available tools to fit small-model context windows.

The two training sets are generated with injected noise (case changes, typos, messy spacing): 60k notes are generated from merchants, places, everyday items, and shapes in short bank descriptions [5], and 44,996 are questions from fixed templates and a synonym grammar, where it is intentional that ask_ai is the largest class (34%) so that sending work to an LLM is the safe default [17]. Suggested categories are not given if the confidence is below 0.60. The fast path answers offline only for a confident read-only label with no action word (68 action words including Hinglish), no loose number, no two joined topics (checked only in Automatic mode) and a bills rule that reserves bill and EMI questions for the bills tool:

$$\text{offline}(q) = [\hat{k} \in \mathcal{K}_r] \wedge [p(\hat{k} \mid x_q) \geq \tau] \wedge \neg A(q) \wedge \neg N(q) \wedge \neg T(q) \wedge \neg B(q, \hat{k}) \qquad (1)$$

where K_r is the set of the eight read-only intent labels, $\tau = 0.80$, A, N, T and B are the four guards and k is the most probable label in the intent model. The tool ranker is a one-vs-rest logistic regression, which generates the independent scores s_t(q) for use in building the private model's menu M(q), similar to how TinyAgent retrieves tools [12].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

DOI: 10.17148/IJARCCE.2026.151065

B. Registry-Driven Lesson Generation

A lesson is one supervised example: a system message, optionally a records list or snapshot, conversation history, user query, and target tool call formatted as JSON. Training examples are automatically synthesized from the live tool registry schema to guarantee syntactically valid function signatures.

C. QLoRA Fine-Tuning and Serving

The private model is the Qwen/Qwen3-1.7B [11] model with QLoRA [10]: the frozen weights are quantized to 4-bit NF4 with double quantization, and LoRA adapters [9] are trained on all 7 attention + MLP projections.

With $r = 16$ and $\alpha = 32$, this gives 17,432,576 trainable parameters (1.003%), where r is the LoRA rank and α its scaling factor. The training is done with paged 8-bit AdamW, cosine schedule, batch size 1, 16-step accumulation and gradient checkpointing, and only the answer tokens are used for the loss.

All tokens are silently graded by the collator for v1, a bug that was discovered later, but only about 2–4% of the tokens are graded for v2–v4. All previous training is included in each of them – v4 contains everything v3 had and more. Version 1 was trained using the laptop GPU, with system memory fallback when 4 GB of dedicated GPU memory was exceeded and trained once without any explicit seed value passed. Versions 2–4 were trained using a free Colab T4 with fp16 with each version being trained only once with no explicit seed value passed. Adapters were merged and converted to GGUF Q4_K_M (approx. 1,056 MiB) with llama.cpp and served by Ollama at temperature 0.2. The examined configuration is C4: finai-v4 + prompt v4, the short menu, up to 10 history messages and 1 retry after "That is not on my list.", which is only activated by machine-local settings (the committed defaults run no private model).

V. EXPERIMENTAL SETUP

A. Question Sets and Their Provenance

The private model is tested on 67 handwritten single questions (the headline set), 32 reported bill requests, 31 developer questions, 15 new-skill questions and 32 developer-written chats; the classifiers are tested on 126 expense notes and 121 plus 68 intent questions. Handwritten scores are development set scores; no blind test set; overlap between training and test data may enhance scores.

B. Metrics and Statistical Tests

For a one-tool item in the hand-written, developer and new-skill sets, the first call must name the tool that is expected; for a multi-tool item and all reported and chat items, all the tool names called in the reply must match the multiset of the item; for a no-tool item, any reply that does not include a call is considered successful, including "That is not on my list.". There is no check or save of arguments. Wherever it differs, a strict (non-accepting "That is not on my list.") variant is reported. Headline small-set scores carry a Wilson 95% interval [18].

The exact two-sided McNemar test [19] is used for paired comparisons, b for items correct only in the first run, c for items correct only in the second run.

Only same day / same code pairs are clean version tests, untuned / v1, etc. are indicative.

C. Protocol and Environment

The evaluation harness invokes the private model via the app's own Ollama adapter, creates v4 menus and history using the app's functions, but hard-codes the date, omits the record list, uses the non-fast path, omits cloud hand over and safety checks, and only times the model call. Decoding was done at temperature 0.2 without a seed, and each configuration was run only once per set (twice for v3 and v4). The same prompt and same tool list were used for both the untuned and v1 runs, but the context window (an unrecorded Ollama default for the untuned run, 4,096 for the v1 run) and the Ollama version and model file differed.

VI. RESULTS AND DISCUSSION

A. Classifiers and the Fast Path

Both classifiers show the gap between generated and hand-written data that is expected when test items are similar to the training items. The accuracy of the category model was 99.32% for the generated notes and 97.62% for the handwritten notes, with 76.98% for the first hand-written run, before the word lists were expanded; one remaining error is confident: "hostel mess and electricity dues" was classified as Rent instead of Bills (0.9827). The intent model achieved 99.94%

DOI: 10.17148/IJARCCE.2026.151065

accuracy on generated questions and 97.06% on the holdout set, up from 82.35% before the synonym grammar was added. The fast path did not answer any of the 84 hand-written LLM-bound questions offline (upper Wilson bound 4.4%), but answered 98 of the 105 read-only questions (93.3%, 86.9–96.7), all using the correct tool; this is a development result, where the guard words have been chosen with the holdout set in mind. The ranker gave all the necessary tools on the menu in 187 of 192 (97.40%) hand-written questions with the shipped menu rule.

B. Private Model Across Versions

Table II lists the versions of the private model. The configuration under study (C4) scored 60/67 = 89.6% (80.0–94.8) on the hand-written set, 29/32 on reported requests, 15/15 on new skills (13/15 strict), 31/31 on developer questions and 32/32 on chats.

TABLE II TOOL-SELECTION ACCURACY OF THE PRIVATE MODEL: RIGHT/N (ACCURACY, WILSON 95% INTERVAL IN %). ONLY TOOL NAMES ARE SCORED; V1'S DEVELOPER AND NEW-SKILL RUNS DATE FROM 2026-10-05; V4 IS THE CONFIGURATION UNDER STUDY (C4).

Version (run date)	Hand-written (n = 67)	Reported (n = 32)	New skills (n = 15)	Developer (n = 31)	Chats (n = 32)
Untuned Qwen3-1.7B (09-29)	48/67 (71.6) 59.9–81.0	–	–	25/31 (80.6) 63.7–90.8	–
v1 (10-01/05)	61/67 (91.0) 81.8–95.8	–	10/15 (66.7) 41.7–84.8	30/31 (96.8) 83.8–99.4	–
v2 (10-07)	61/67 (91.0) 81.8–95.8	21/32 (65.6) 48.3–79.6	10/15 (66.7) 41.7–84.8	31/31 (100) 89.0–100	–
v3 (10-07)	62/67 (92.5) 83.7–96.8	29/32 (90.6) 75.8–96.8	13/15 (86.7) 62.1–96.3	30/31 (96.8) 83.8–99.4	–
v3 re-run (10-10/11)	61/67 (91.0) 81.8–95.8	29/32 (90.6) 75.8–96.8	12/15 (80.0) 54.8–93.0	30/31 (96.8) 83.8–99.4	29/32 (90.6) 75.8–96.8
v4, short menu (10-11)	60/67 (89.6) 80.0–94.8	29/32 (90.6) 75.8–96.8	15/15 (100) 79.6–100	31/31 (100) 89.0–100	32/32 (100) 89.3–100

The only significant change in the 67 is from the untuned model to v1 (48/67 to 61/67; b = 0 and c = 13; exact p = 0.0002), and 12 of those 13 gains were questions that were answered by the untuned model without invoking any tool. There is also the issue that both runs used different context windows, and different versions and model files of Ollama, meaning this is a result that is consistent with a fine-tuning effect, but not a clean causal estimate. Later steps on the 67 are not significant: v1 to v2, v2 to v3 and the same-code v3 re-run to v4 all give p = 1.0 [19]. On the reported set, v2 to v3 is significant (21/32 to 29/32, p = 0.0078), but is a fix check: all 8 gains needed a tool added after training started for v2, and the set shaped v3's training. On the chats, v4 leads only by count (32/32 against 29/32, p = 0.25).

C. Run-to-Run Variation and Error Analysis

On two dates, the same v3 with the same inputs did 62 and 61 of 67; on all identical-input repeats 4 of 312 answers (1.28%, 0.50–3.25) differed, about one point on the 67. Errors are mainly systematic with 11 of the 13 different v3 misses being repeated with the same incorrect tool. Of the shipped v4's seven hand-written misses, 3 are "That is not on my list.", 2 chose the wrong tool and 2 are destructive choices (deleting a bill, and deleting everything for "wipe everything i logged this month", which the plan check refuses). Most of the failures of the untuned model were in answering a question with no tool call (14 of 19), while v1 answered no tool-needing question without a call, so most of the improvement (48→61) resulted from v1 calling a tool when one was required.

VII. LIMITATIONS AND THREATS TO VALIDITY

A. Validity of the Measurements

Only the tool names chosen by the private model are scored (arguments are not checked and not stored and on any no-tool item any answer not containing a call is considered correct). The context window, the Ollama version and the model file are confounding variables in the untuned–v1 comparison. All private-model scores come from a single run without a

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

DOI: 10.17148/IJARCCE.2026.151065

seed; each version was trained once, and version rows differ in registries (25, 31 and 32 tools), different tool wordings, different prompts, different tool offers, different context windows, and a v4-only retry.

B. Test Sets and Decisions

The Wilson intervals are optimistic as none of the sets have a recorded independent author and all were used for design decisions, and several sets overlap with lesson sets; removing the overlapping item barely changes the headline (v4: 59/66 = 89.4%). Version 4, which had to be no worse on the 67, and better on a team-written conversation test before being switched on, also failed that switch rule; it was switched on using a later "about equal or better" rule set with developer chats.

C. Not Evaluated

No user study was conducted, the effectiveness of the Proceed cards and safety checks beyond tests is unknown, the end-to-end Automatic pipeline was not measured and also the question sets on the paper were not used for the cloud model.

D. Privacy

Voice clips are transmitted to Groq Whisper, reply texts to edge-tts in all engine modes; cloud advice is not checked for numbers; one plain new record from the cloud can be saved without having a card; a network-level check was not recorded.

VIII. CONCLUSION AND FUTURE WORK

In this paper, a new voice-first personal-finance assistant, named FIN-AI, was introduced, with the language models generally selecting the tools used in the answer, the code performing the calculations on these tools, and the data changes proposed by the language models needing to be confirmed by the user, apart from the few stated exceptions. In the case of the 1.7B model (laptop, without fast path and hand over in the cloud), the expected tool was selected for 60/67 (approx. 90%) of the hand-written single questions. The only significant change on the 67 was at the first fine-tune (48 to 61, a confounded comparison) and all evaluations were performed on development sets.

The work to be done includes blind multi-author test sets frozen before scoring, including out-of-scope questions [17]; argument level scoring and repeated running; same day comparison of the base model and the fine-tuned model with equal context windows; cloud runs and end-to-end runs on the same test sets; replaying incorrect change calls in the plan checks; real-speech word error rate on Indian-accented English, for example on Svarah [15]; and gating speech pre-synthesis based on the voice mode and the "Voice output" switch.

REFERENCES

[1] A. Lusardi and O. S. Mitchell, "The economic importance of financial literacy: Theory and evidence," Journal of Economic Literature, vol. 52, no. 1, pp. 5–44, 2014. https://doi.org/10.1257/jel.52.1.5

[2] A. Demirgüç-Kunt, L. Klapper, D. Singer, and S. Ansar, The Global Findex Database 2021: Financial Inclusion, Digital Payments, and Resilience in the Age of COVID-19. Washington, DC: World Bank, 2022. https://doi.org/10.1596/978-1-4648-1897-4

[3] I. Medhi, S. N. N. Gautama, and K. Toyama, "A comparison of mobile money-transfer UIs for non-literate and semi-literate users," in Proc. SIGCHI Conf. Human Factors in Comput. Syst. (CHI '09), New York: ACM, 2009, pp. 1741–1750. https://doi.org/10.1145/1518701.1518970

[4] F. Wang et al., "A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with LLMs, and trustworthiness," ACM Trans. Intell. Syst. Technol., vol. 16, no. 6, pp. 1–87, 2025. https://doi.org/10.1145/3768165

[5] S. García-Méndez, M. Fernández-Gavilanes, J. Juncal-Martínez, F. J. González-Castaño, and Ó. B. Seara, "Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus," IEEE Access, vol. 8, pp. 61642–61655, 2020. https://doi.org/10.1109/ACCESS.2020.2983584

[6] 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, pp. 42:1–42:55, 2025. https://doi.org/10.1145/3703155

[7] T. Schick et al., "Toolformer: Language models can teach themselves to use tools," in Adv. Neural Inf. Process. Syst., vol. 36, pp. 68539–68551, 2023. https://doi.org/10.52202/075280-2997

DOI: 10.17148/IJARCCE.2026.151065

[8] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan, "τ-bench: A benchmark for tool-agent-user interaction in real-world domains," in Proc. 13th Int. Conf. Learn. Represent. (ICLR 2025), 2025, pp. 9965–10017.

[9] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in Proc. 10th Int. Conf. Learn. Represent. (ICLR 2022), 2022. https://openreview.net/forum?id=nZeVKeeFYf9

[10] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," in Adv. Neural Inf. Process. Syst., vol. 36, pp. 10088–10115, 2023. https://doi.org/10.52202/075280-0441

[11] A. Yang et al., "Qwen3 technical report," arXiv preprint arXiv:2505.09388, 2025. https://doi.org/10.48550/arXiv.2505.09388

[12] L. E. Erdogan et al., "TinyAgent: Function calling at the edge," in Proc. 2024 Conf. Empirical Methods in Natural Language Processing: System Demonstrations, Miami: ACL, 2024, pp. 80–88. https://doi.org/10.18653/v1/2024.emnlp-demo.9

[13] L. Chen, M. Zaharia, and J. Zou, "FrugalGPT: How to use large language models while reducing cost and improving performance," Trans. Mach. Learn. Res., 2024. https://openreview.net/forum?id=cSimKw5p6R

[14] A. Radford et al., "Robust speech recognition via large-scale weak supervision," in Proc. 40th Int. Conf. Mach. Learn. (PMLR), vol. 202, 2023, pp. 28492–28518.

[15] T. Javed et al., "Svarah: Evaluating English ASR systems on Indian accents," in Proc. Interspeech 2023, 2023, pp. 5087–5091. https://doi.org/10.21437/Interspeech.2023-2588

[16] Y. Ruan et al., "Identifying the risks of LM agents with an LM-emulated sandbox," in Proc. 12th Int. Conf. Learn. Represent. (ICLR 2024), 2024, pp. 27031–27098.

[17] S. Larson et al., "An evaluation dataset for intent classification and out-of-scope prediction," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), Hong Kong: ACL, 2019, pp. 1311–1316. https://doi.org/10.18653/v1/D19-1131

[18] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," J. Amer. Statist. Assoc., vol. 22, no. 158, pp. 209–212, 1927. https://doi.org/10.1080/01621459.1927.10502953

[19] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," Psychometrika, vol. 12, no. 2, pp. 153–157, 1947. https://doi.org/10.1007/BF02295996

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