Context Tuning for Retrieval Augmented Generation

Raviteja Anantha Tharun Bethi Danil Vodianik Srinivas Chappidi

Apple

Overview

Large language models (LLMs) have the remarkable ability to solve new tasks with just a few examples, but they need access to the right tools. Retrieval Augmented Generation (RAG) addresses this problem by retrieving a list of relevant tools for a given task. However, RAGâs tool retrieval step requires all the required information to be explicitly present in the query. This is a limitation, as semantic search, the widely adopted tool retrieval method, can fail when the query is incomplete or lacks context. To address this limitation, we propose Context Tuning for RAG, which employs a smart context retrieval system to fetch relevant information that improves both tool retrieval and plan generation. Our lightweight context retrieval model uses numerical, categorical, and habitual usage signals to retrieve and rank context items. Our empirical results demonstrate that context tuning significantly enhances semantic search, achieving a 3.5-fold and 1.5-fold improvement in Recall@K for context retrieval and tool retrieval tasks respectively, and resulting in an 11.6% increase in LLM-based planner accuracy. Additionally, we show that our proposed lightweight model using Reciprocal Rank Fusion (RRF) with LambdaMART outperforms GPT-4 based retrieval. Moreover, we observe context augmentation at plan generation, even after tool retrieval, reduces hallucination.

User Centric Persona Synthesis

We simulate realistic underspecified and implicit interactions using GPT-4 across various applications commonly found with digital assistants. The dataset is structured to encompass a diverse range of contexts, representing different synthetic user activities and interactions. A total of 791 unique personas were synthesized, covering seven key applications, shown in Table 1. The final dataset contained 4,338 train and 936 test data points.

Table: Distribution of Context and Tools generated by GPT-4 based User Centric Persona Synthesis.

Application	Avg. Context Items	Tools Count
Music	4.38	11
Google	9.57	10
Notes	2.23	9
Mail	2.93	8
PhoneCall	2.34	8
Calendar	5.63	7
Reminders	4.81	6


```
{
 "calendar": [
  {
   "category": "Work Event",
   "event_name": "Crop Production Meeting",
   "organizer": "redacted_email@domain.com",
   "participants": ["redacted_email@domain.com"],
   "location": "Main Farm",
   "start_time": "2023-09-28T08:15:30.283665",
   "end_time": "2023-09-28T10:15:30.283665",
   "status": "Confirmed",
   "notes": "Discuss the yield and soil health"
  },
  // More calendar events...
 ],
 "google_searches": [
  { "query": "Boxing matches to watch",
   "date": "2023-09-30T21:30:00.283665Z" },
  // More searches...
 ],
 "emails": [
  {
   "author": "redacted_email@domain.com",
   "recipients": ["redacted_email@domain.com"],
   "subject": "Farm Updates",
   "content": "Dear [Name], I would like to share some updates from the farm..."
   // Additional email fields...
  },
  // More emails...
 ],
 // Music History and Remainders...
}
```


Figure: Snippet of a Persona.

Retrieval Augmented Generation with Context Retrieval

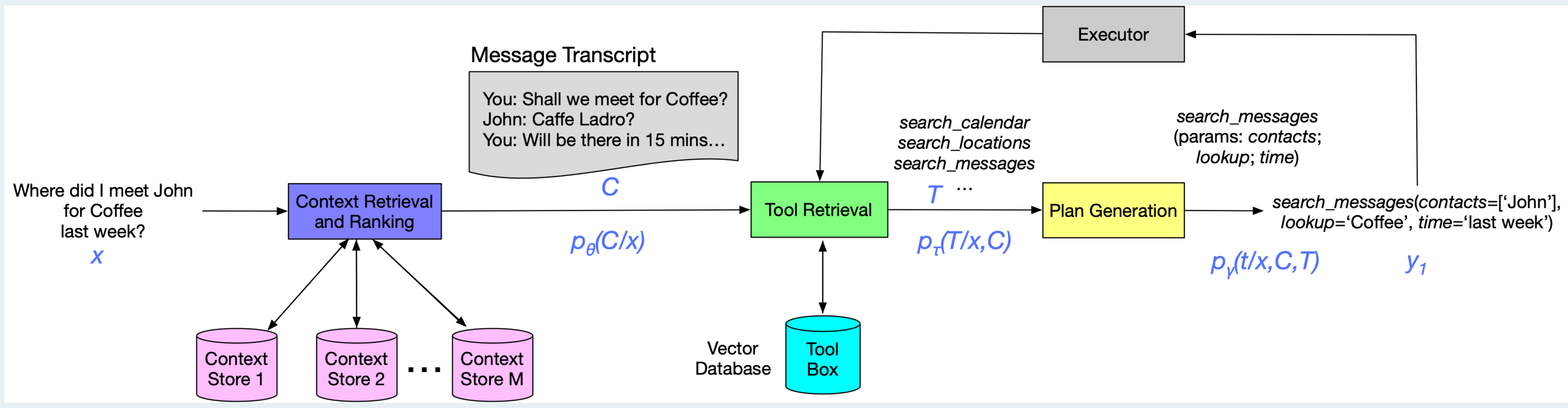

Figure: Context-tuned RAG pipeline illustrating end-to-end processing of a complex request with progressive plan generation.

Sample of Context-Seeking Queries

We generate CoT using GPT-4 to guide the planner in resolving tool ambiguity. Table 2 showcases examples of generated implicit queries alongside their corresponding CoT, context, and top-3 relevant tools.

Table: A sample of context-seeking or under-specified queries along with CoT produced by GPT-4. The columns for context and tools show labels for those retrieval tasks.

Implicit Query	CoT	Relevant Context	Top-3 Relevant Tools
When is my next guitar lesson?	Check the 'Calendar' for any upcoming guitar lessons. If not there, check 'Reminders' for any alerts set about the lesson.	The user has a reminder titled "Guitar Class"	['Reminders', 'Calendar', 'Notes']
I need to check my diet plan again.	I may have noted down the diet plan in 'Notes'. If not there, perhaps I saved a photo of it in 'Photos'.	The user has a note titled "Intermittent Fasting Plan." The user also has an image titled "Keto Diet."	['Photos', 'Notes', 'Mail']
I'm running late.	Check 'Calendar' for any scheduled meetings. If so, verify 'Maps' or 'Google Maps' to gauge current traffic situation and estimated time of arrival. Use 'Messages' or 'Messenger' or 'Mail' to inform the meeting attendees that you are "running late".	The user has an upcoming meeting titled "LLM Discussion" organized by "John Doe."	['Calendar', 'Mail', 'Messages']

Methodology

Our experiments train and evaluate tool retrieval and planning with and without context tuning.

Context Tuning To compare various context retrieval methods, we employ both text-based and vector-based retrieval baselines. We simulate different context stores by structuring context data per persona and train models to perform federated search. We use query and persona meta-signals, such as frequency, usage history, and correlation with geo-temporal features, to perform retrieval. We evaluate context retrieval using the Recall@K and Normalized Discounted Cumulative Gain (NDCG@K) metrics.

Tool Retrieval We employ the pre-trained GTR-T5-XL model for semantic search using cosine similarity to retrieve the top-K tools. Extending the tool retrieval process to incorporate ranking should be a straightforward endeavor. We evaluate tool retrieval performance with and without context retrieval using Recall@K.

Planner The plannerâs objective is to select the most appro- priate tool from the retrieved tool list and generate a well-formed plan. A plan comprises an API call constructed using the chosen tool and parame- ters extracted from the query and retrieved context. We fine-tune OpenLLaMA-v2-7B (Touvron et al., 2023) for plan generation. To assess the plannerâs performance, we employ the Abstract Syntax Tree (AST) matching strategy to compute plan accuracy. A hallucination is defined as a plan generated using an imaginary tool.

Comparison of various Context Retrieval methods

Retrieval Method	Recall@K			NDCG@K		
	K=3	K=5	K=10	K=3	K=5	K=10
BM25	11.35	13.47	14.92	56.45	52.33	50.91
Semantic Search	23.74	25.38	26.99	65.44	64.31	64.02
CoT Augmentation	71.77	85.61	94.41	93.67	91.78	88.40
Finetuned Semantic Search	73.48	88.52	95.13	93.81	94.07	94.23
Finetuned w/ CoT Augmentation	73.55	88.53	95.17	93.92	94.11	94.22
LambdaMART-RRF	**81.27**	**92.65**	**98.77**	**96.39**	**97.11**	**98.24**

Figure: A comparison of various Context Retrieval methods using Recall@K and NDCG@K metrics. The context-seeking query is used as input to perform a federated search across different context stores, after which semantic search or ranking is applied.

Tool Retrieval Results

Incorporating relevant context into Tool Retrieval consistently yields substantial gains across various K-values. Recall@K for Tool Retrieval improved from a range of 52%-64% to 75%-97% for K = 1 to 10.

Planner Results

To establish the plannerâs lower bound, we remove the retrieval step, while the upper bound is set by directly utilizing context and/or tool labels, effectively employing oracle retrievers. Table 4 encapsulates the end-to-end evaluation of the fine-tuned planner, demonstrating that the context-tuned planner significantly outperforms the planner based on traditional RAG using semantic search. Notably, even when the correct tool is retrieved, incorporating relevant context in plan generation, as evidenced by the upper bound, helps in reducing hallucination.

Setting	AST-based Plan Acc ↑	Exact Match ↑	Hallucination ↓
Lower Bound	43.77	39.45	2.59
RAG-based Planner	76.39	58.12	1.76
Context-tuned RAG Planner	**85.24**	**67.33**	**0.93**
Upper Bound	91.47	72.65	0.85
Context-tuned Upper Bound	91.62	72.84	**0.53**

Figure: End-to-end planner evaluation both with and without context tuning. "Lower Bound" excludes retrieval and performs direct plan generation while "Upper Bound" assumes perfect context and tool retrieval.