App Inventor Foundation

ICLR

Factual Context Validation and Simplification: A Scalable Method to Enhance GPT Trustworthiness and Efficiency

Tianyi Huang

ICLR 2025 Blogposts Track

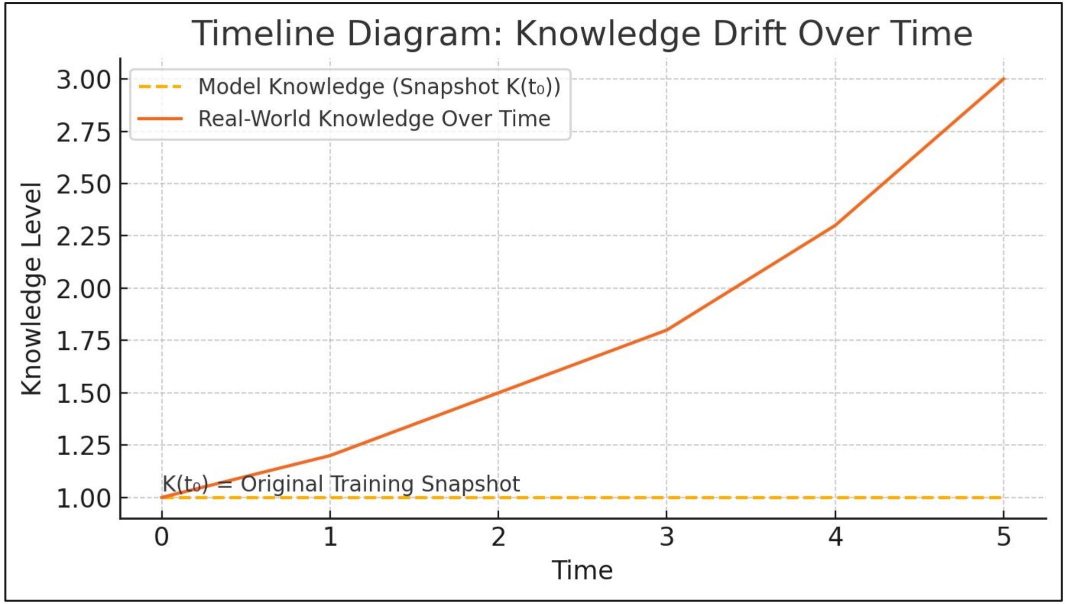
Motivation

GPT models are powerful, but prone to **hallucinations**, where a key challenge involves **mismatched**/**outdated** knowledge.

Knowledge divergence example:

$K(t) = K(t_0) + \Delta K(t)$,

where in practice $\Delta K(t) = 0$ for most deployed LLMs ⟶ knowledge remains "frozen" and results in **increasingly inaccurate** outputs.



App Inventor Foundation

Research Aims

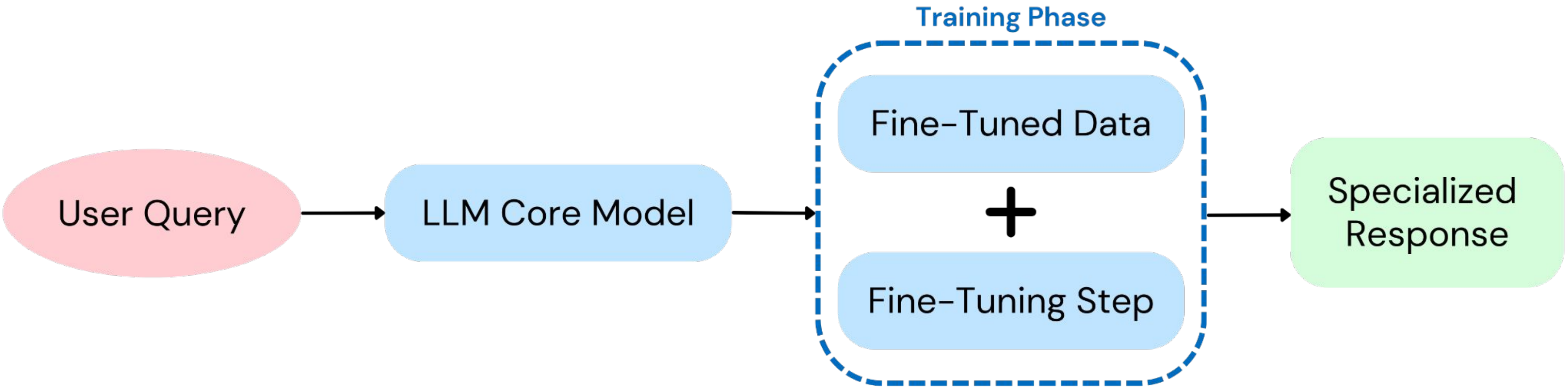
Since each step P(x□) < 1 drives overall correctness down **exponentially**, our approach aims to address three main aspects:

1. **Granular Fact Validation**: Decomposing outputs into small “atomic” claims.
2. **Efficient Context Management**: Summarization & Clustering can reduce storage by up to 57.7%.
3. **Robust RAG Integration**: Minimizing error propagation in multi-step reasoning.

Our research demonstrates that this granular approach, combined with efficient context management, has the potential to enhance both **accuracy** and **computational efficiency**.

Existing Methods & Their Shortcomings

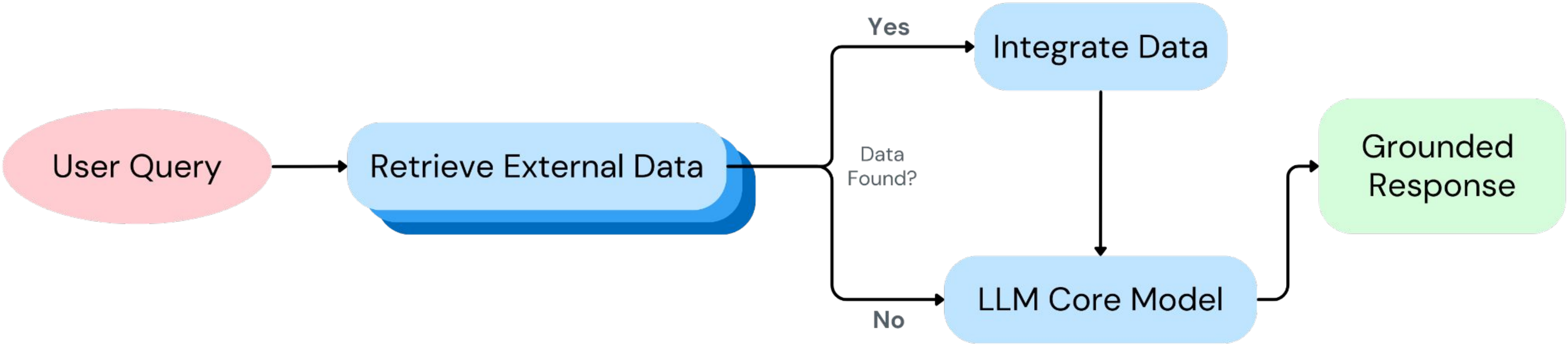
- Fine-Tuning: Helps to improve domain-specific knowledge.
  - But: Is **expensive** and quickly **outdated**.



App Inventor Foundation

Existing Methods & Their Shortcomings

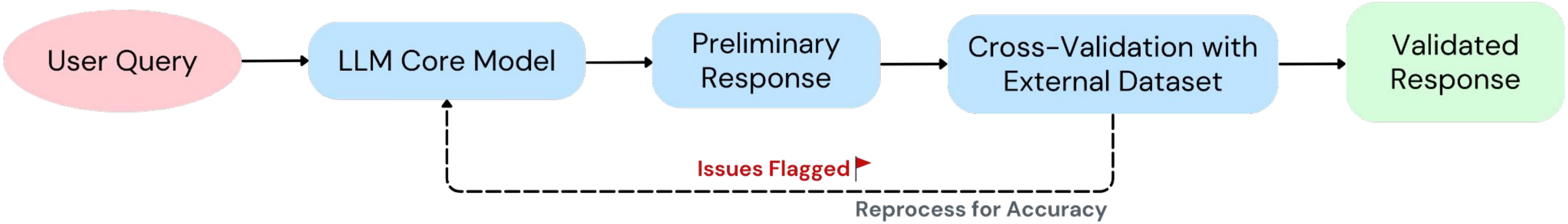
- RAG: Helps in retrieving context and information.
    - But: Contains **no inherent** truth validation, only retrieval.



App Inventor Foundation

Existing Methods & Their Shortcomings

- Post-hoc Correction: Helps to correct factual inaccuracies.
  - But: Introduces **latency**, and offers **no improvement** of base generation.



App Inventor Foundation

Existing Methods & Their Shortcomings

While these approaches offer **valuable improvements** to ensuring factual accuracy in LLM, they often fail to **inherently** validate responses or address the **root causes** of hallucinations. Our goal is to unify the best of retrieval with a **lightweight, statement-level validation mechanism**.

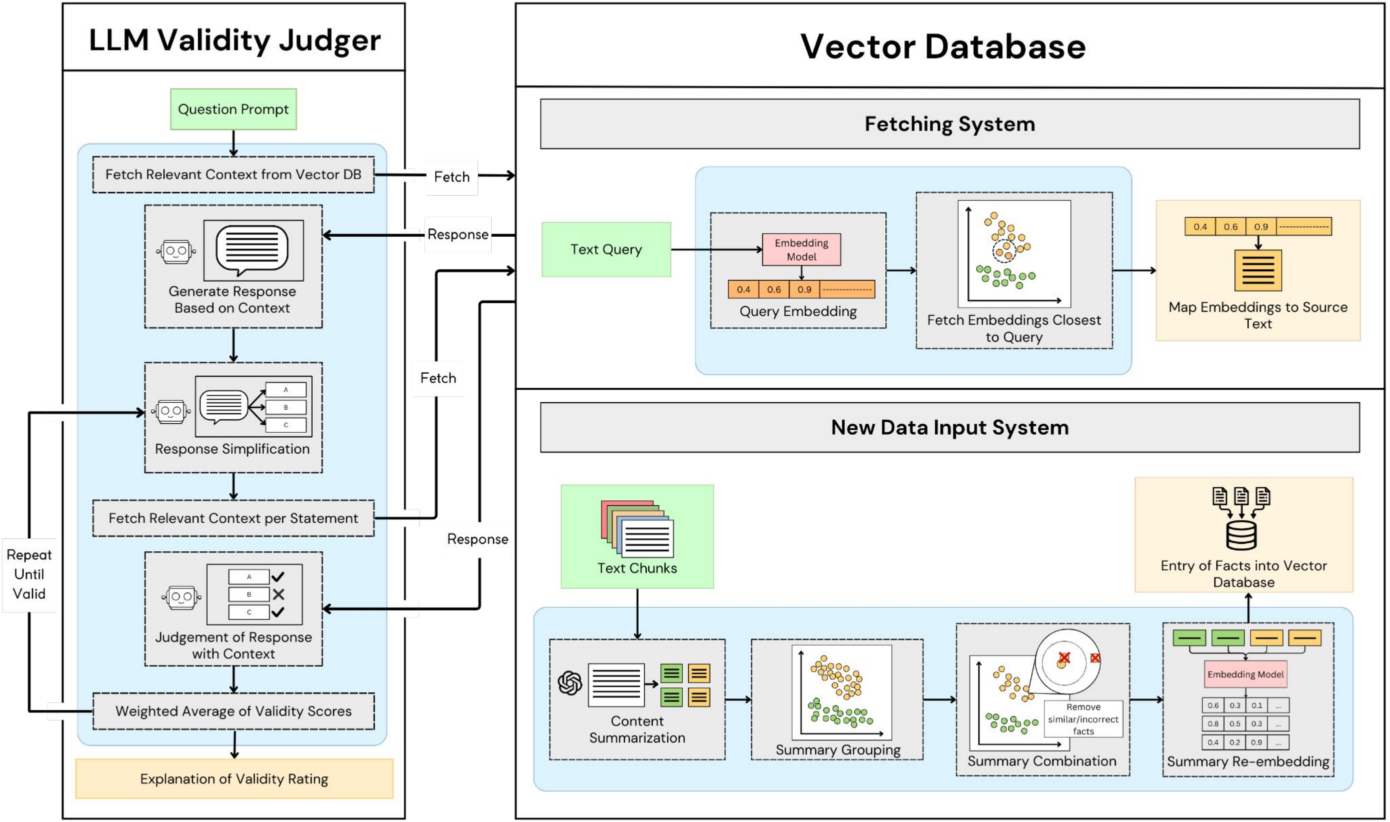
We need:

Granularity + Verification + Scalability

Error Propagation & Hessian Insights

- **Cascading errors:** A small inaccuracy at step 1 can balloon in later steps, resulting in **exponential** decay of correctness
  - $P(\text{total}) = \prod(n=1 \text{ to } N)\ P(x□)$
  - Exponential decay if each $P(x_n) < 1$.
- Second-order effects captured by a Hessian can **amplify** errors:
  - Hessian $H_i□ = \partial^2 E / (\partial x_i\, \partial x□)$
  - Small local errors can magnify each other.

Proposed Framework



LLM Validity Judger

Question Prompt

Fetch Relevant Context from Vector DB

Generate Response Based on Context

Response Simplification

Fetch Relevant Context per Statement

Judgement of Response with Context

Weighted Average of Validity Scores

Explanation of Validity Rating

Repeat Until Valid

Fetch

Response

Fetch

Response

Vector Database

Fetching System

Text Query

Query Embedding

Fetch Embeddings Closest to Query

Map Embeddings to Source Text

New Data Input System

Text Chunks

Content Summarization

Summary Grouping

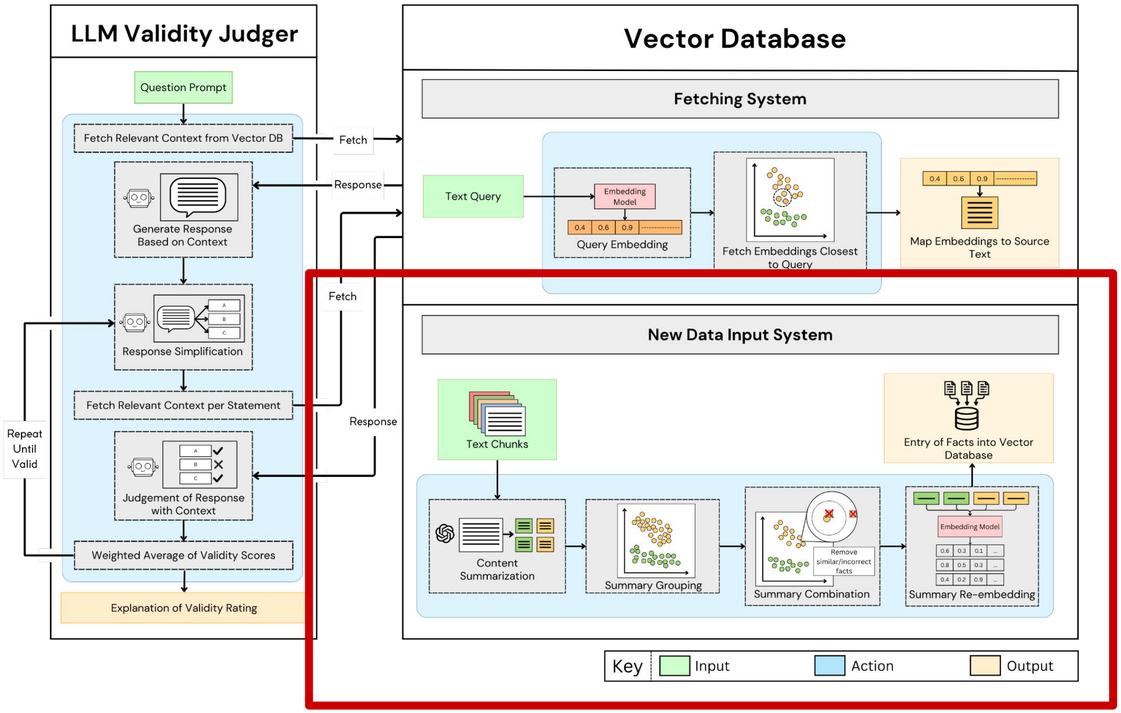
Summary Combination

Summary Re-embedding

Entry of Facts into Vector Database

Key 🟩 Input 🟦 Action 🟨 Output

Proposed Framework



Data Preprocessing:

- **Summarize** each text chunk using GPT.
- Embed using text-embedding-3-large → $v \in \mathbb{R}^d$.
- **Cluster** with DBSCAN:
  - Condition: $d(v_□, v_\varphi) \le \varepsilon$, local density ≥ minPts → same cluster.
- **Re-summarize** each cluster → store final embeddings.
  - Achieves significant memory savings.

Proposed Framework

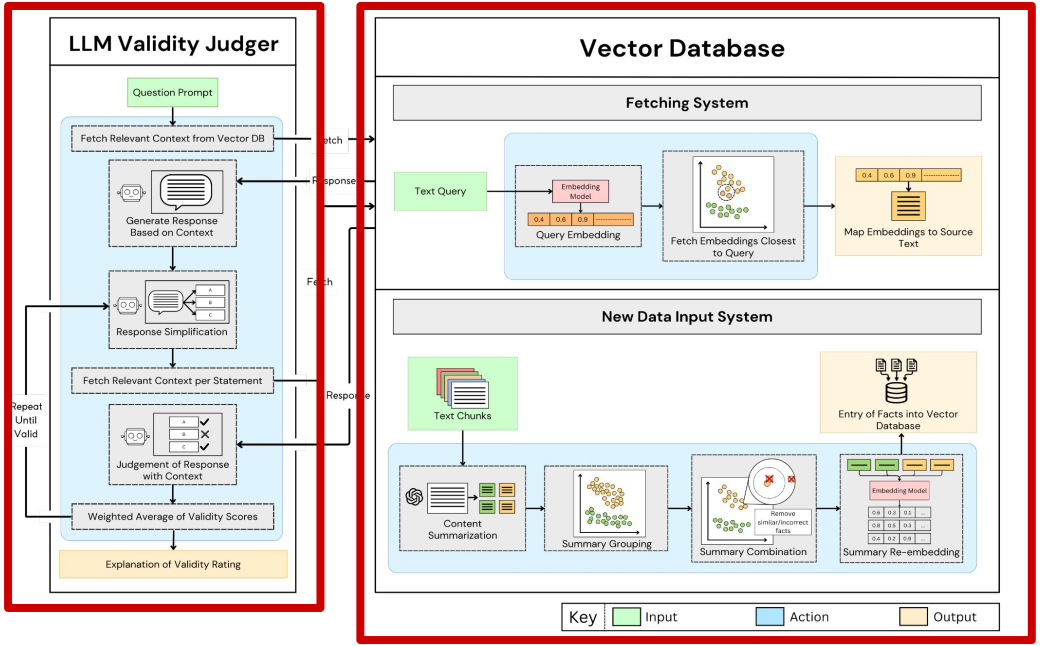
LLM Validity Judger

Provides **granular fact-checking** at the statement level:

1. **Splits** response

2. Statement-level **validation**

3. Final **rating**.



Vector Database

Enables **efficient** similarity searches:

- Store embeddings $\{v_i\}$ in $\mathbb{R}$^d.
- Similarity: Sim(v_i, k□) = (v_i · k□) / (‖v_i‖ ‖k□‖).

App Inventor Foundation

Alternative: Statement-Level Granularity

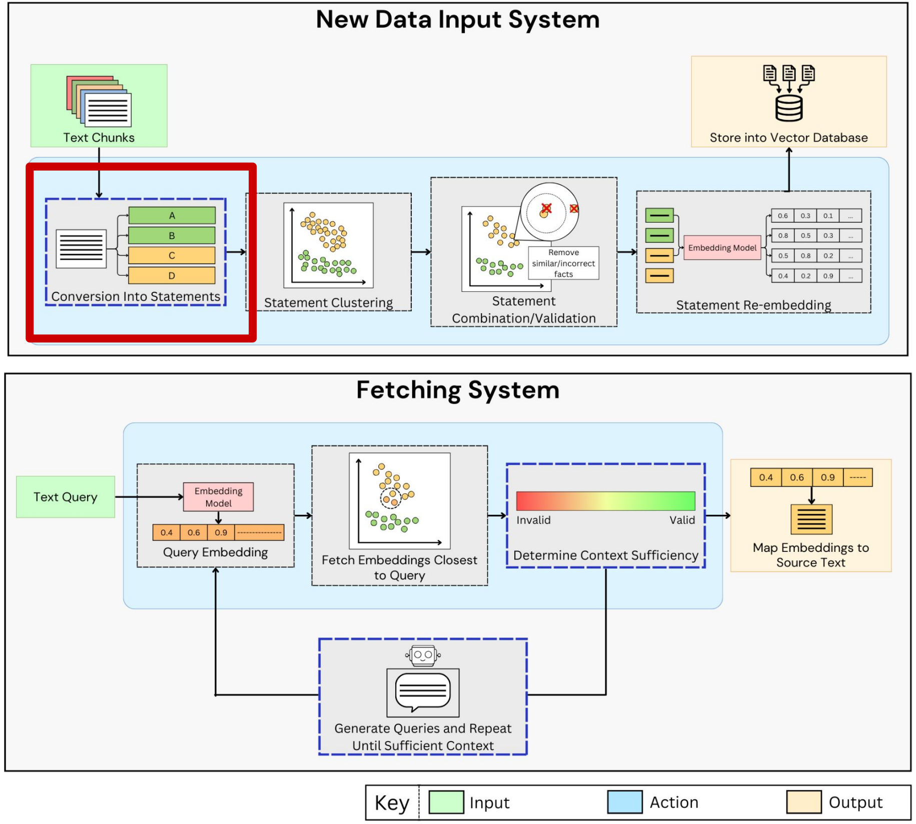
For **high-stakes** contexts where no detail can be lost (fields like medicine/law), summarizing can potentially **cut out** disclaimers or edge-case information. We propose an alternative pipeline:

- **Extracts** each factual statement from text
- Clusters duplicates for **minimal** compression
- Stores **each** statement as an embedding
- Trade-Off: Less **compression**, but higher **fidelity** and **reliability**
  - Ensures that **critical** information is retained →ideal for applications where **precision** is paramount

Alternative: Statement-Level Granularity

Condenses input data into **standalone**, **verifiable** statements.
→ ensures no crucial content is omitted

Statement Extraction Pipeline



New Data Input System

Text Chunks

A

B

C

D

Conversion Into Statements

Statement Clustering

Remove similar/incorrect facts

Statement Combination/Validation

Embedding Model

0.6 0.3 0.1 ...

0.8 0.5 0.3 ...

0.5 0.8 0.2 ...

0.4 0.2 0.9 ...

Statement Re-embedding

Store into Vector Database

Fetching System

Text Query

Embedding Model

0.4 0.6 0.9 ---------------

Query Embedding

Fetch Embeddings Closest to Query

Invalid

Valid

Determine Context Sufficiency

0.4 0.6 0.9 -----

Map Embeddings to Source Text

Generate Queries and Repeat Until Sufficient Context

Key 🟩 Input 🟦 Action 🟨 Output

App Inventor Foundation

Statement-Level Validation

This framework takes a **granular approach** that allows us to identify **specific** inaccuracies in a response rather than making a binary judgement about the entire output. Therefore, we can provide more **nuanced** feedback and **improve** overall factual accuracy by validating at the **statement level**, specifically where:

- LLM output is **decomposed** → {F_1, F_2, …, F□}.
    - e.g.) “F_1: The patient is 24 years old,” “F_2: She has a history of X”
- For each F_i, retrieve **top-k** matches from DB.
- Score each $F_i \in [0,1]$ via **alignment** with matching facts.
- **Aggregate** scores (weighted average or other aggregator).

Benchmark: PubMedQA

- Dataset: 1,000 QA pairs (Yes/No/Maybe)
- Baseline: Traditional RAG storing entire paragraphs.
- Our Pipeline: Summarization and Clustering.
- Metrics:
  - Factual Accuracy, RAG Effectiveness, Storage Efficiency.

Summarization Pipeline vs. Traditional RAG

Metric	Traditional Pipeline	Proposed Pipeline	Difference
Factual Accuracy	71.7%	71.2%	-0.5%
RAG Effectiveness	99.2%	98.9%	-0.3%
Storage Efficiency	1,351 KB	571 KB	-57.7% (Reduction)

- ~57.7% **reduction** in storage and **near-parity** on factual accuracy (within 0.5%) + RAG effectiveness (within 0.3%)
    - summarizing context does not hinder the LLM's ability to generate correct answers.
    - Significant for large scale deployments: less data to store and query with minimal performance loss
- Maintains performance while significantly reducing computational and storage requirements

Statement Extraction Pipeline

SQuAD

Metric	Traditional Pipeline	Statement Extraction	Difference
Factual Accuracy	87.3%	89.7%	+2.4%
Storage Size	1.4 MB	1.1 MB	-21.43%

HotpotQA

Metric	Traditional Pipeline	Statement Extraction	Difference
Factual Accuracy	92.0%	93.3%	+1.3%
Storage Size	763 KB	701 KB	-8.12%

- Gains in **multi-hop reasoning** from statement-level detail.
- **Improvement in accuracy** for both benchmarks → demonstrates that statement-level granularity can enhance performance on complex reasoning tasks.
- Valuable approach for applications where **precision** is more important than storage efficiency.

Error Minimization & Scalability

- Local validation **mitigates** exponential decay in multi-step correctness.
- Summaries or statements → O(N) vectors stored.
- DBSCAN runs in O(N × log N) or similar (depending on implementation).
- **Implementation**:
  - Vector DB with approximate nearest neighbor (e.g., Pinecone, FAISS).
  - GPT-4o-mini for summarization.
- **Modest** overhead, **reasonable** computational complexity, uses **existing** tools and libraries
  - **Practical** for real-world applications

Open Challenges & Next Steps

- Source Bias: Original documents may contain **biases** → pipeline inherits them.
- Context Gaps: Summaries or statements can **lose** broader discourse context.
- Real-Time Updates: Knowledge updates currently handled in **manual** embedding steps.
- Future Work → Concept-Based Representation:
    - Store knowledge as **relationships** (CONCEPT$_1$, RELATION, CONCEPT$_2$).
    - Potentially more **robust** for advanced reasoning

Key Takeaways

We propose two **flexible** pipelines for **factual context validation**:

- Summarization and Clustering → around 57.7% increase in **memory savings** with **minimal** performance penalty.
- Statement Extraction → preserves **full detail,** can improve **multi-hop accuracy**

Both pipelines use statement-level checks to combat **error cascades** and reduce **hallucinations**. In addition, they can be **easily integrated** into any standard RAG approach.

Our work contributes to the ongoing effort to make large language models more **trustworthy** and **efficient**, particularly in high-stakes domains where **factual accuracy** is critical.

Thank You!

Contact:

- tianyi@appinventorfoundation.org
- tianyi@appinclub.org

Check Out Code Repository:

github.com/Tonyhrule/Factual-Validation-Simplification

LinkedIn

OpenReview

App Inventor Foundation

References

Language Models Are Few-Shot Learners
Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., 2020. arXiv preprint arXiv:2005.14165.

Survey of Hallucination in Natural Language Generation
Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A. and Fung, P., 2022. arXiv preprint arXiv:2202.03629.

On Faithfulness and Factuality in Abstractive Summarization
Maynez, J., Narayan, S., Bohnet, B. and McDonald, R., 2020. arXiv preprint arXiv:2005.00661.

The Curious Case of Neural Text Degeneration
Holtzman, A., Buys, J., Du, L., Forbes, M. and Choi, Y., 2020. arXiv preprint arXiv:1904.09751.

A Guide to Deep Learning in Healthcare
Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S. and Dean, J., 2019. Nature Medicine, Vol 25, pp. 24--29.

Challenges and Barriers of Using Large Language Models (LLM) Such as ChatGPT for Diagnostic Medicine with a Focus on Digital Pathology – A Recent Scoping Review
Ullah, E., Parwani, A., Baig, M.M. and Singh, R., 2024. Diagnostic Pathology, Vol 19.
REALM: Retrieval-Augmented Language Model Pre-Training
Guu, K., Lee, K., Tung, Z., Pasupat, P. and Chang, M., 2020. arXiv preprint arXiv:2002.08909.

A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise
Ester, M., Kriegel, H., Sander, J. and Xu, X., 1996.

PubMedQA: A Dataset for Biomedical Research Question Answering
Jin, Q., Dhingra, B., Liu, Z., Cohen, W.W. and Lu, X., 2019. arXiv preprint arXiv:1909.06146.

Language Models are Unsupervised Multitask Learners
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019.

Is Your LLM Outdated? Benchmarking LLMs \& Alignment Algorithms for Time-Sensitive Knowledge
Mousavi, S.M., Alghisi, S. and Riccardi, G., 2024. arXiv preprint arXiv:2404.08700.

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?
Bender, E., McMillan-Major, A., Shmitchell, S. and Gebru, T., 2021. FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 610--623.

Chain of Thought Prompting Elicits Reasoning in Large Language Models
Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. and Zhou, D., 2022. arXiv preprint arXiv:2201.11903.

Exact Calculation of the Hessian Matrix for the Multilayer Perceptron
Bishop, C.M., 1992. Neural Computation, Vol 4(4), pp. 494--501. MIT Press.

Universal Language Model Fine-tuning for Text Classification
Howard, J. and Ruder, S., 2018. arXiv preprint arXiv:1801.06146.

Fine-Tuning Pre-Trained Language Models Effectively by Optimizing Subnetworks Adaptively
Zhang, H., Li, G., Li, J., Zhang, Z., Zhu, Y. and Jin, Z., 2022. arXiv preprint arXiv:2211.01642.

On the Opportunities and Risks of Foundation Models
Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K. and Liang, P., 2021. arXiv preprint arXiv:2108.07258.

References

Transfer Learning in Natural Language Processing
Ruder, S., Peters, M.E., Swayamdipta, S. and Wolf, T., 2019. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15--18.

What is Retrieval-Augmented Generation?
Martineau, K., 2021. IBM Research Blog.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S. and Kiela, D., 2021. arXiv preprint arXiv:2005.11401.

Dense Passage Retrieval for Open-Domain Question Answering
Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W., 2020. arXiv preprint arXiv:2004.04906.

Summary of a Haystack: A Challenge to Long-Context LLMs and RAG Systems
Laban, P., Fabbri, A.R., Xiong, C. and Wu, C., 2024. arXiv preprint arXiv:2407.01370.

Harnessing Large Language Models as Post-hoc Correctors
Zhong, Z., Zhou, K. and Mottin, D., 2024. arXiv preprint arXiv:2402.13414.

Towards a Unified Language Model for Knowledge-Intensive Tasks Utilizing External Corpus
Li, X., Dou, Z., Zhou, Y. and Liu, F., 2024. arXiv preprint arXiv:2402.01176.

Billion-scale Similarity Search with GPUs
Johnson, J., Douze, M. and Jégou, H., 2017. arXiv preprint arXiv:1702.08734.

Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks
Reimers, N. and Gurevych, I., 2019. arXiv preprint arXiv:1908.10084.

GPT-4o Mini: Advancing Cost-Efficient Intelligence
OpenAI,, 2024. OpenAI Research.

OpenAI Platform
OpenAI,, 2024. OpenAI Documentation.

Pinecone Documentation: Getting Started
Pinecone,, 2024. Pinecone Documentation.

Evaluating the Factual Consistency of Abstractive Text Summarization
Kryściński, W., McCann, B., Xiong, C. and Socher, R., 2019. arXiv preprint arXiv:1910.12840.

A Brief Review of Nearest Neighbor Algorithm for Learning and Classification
Taunk, K., De, S., Verma, S. and Swetapadma, A., 2019. 2019 International Conference on Intelligent Computing and Control Systems (ICCS), pp. 1255--1260.
k-Nearest Neighbour Classifiers: 2nd Edition (with Python Examples)
Cunningham, P. and Delany, S.J., 2020. arXiv preprint arXiv:2004.04523.

SQuAD: 100,000+ Questions for Machine Comprehension of Text
Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., 2016. arXiv preprint arXiv:1606.05250.

HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering
Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R. and Manning, C.D., 2018. arXiv preprint arXiv:1809.09600.

Introducing Gemini 2.0: our new AI model for the agentic era
Pichai, S., Hassabis, D. and Kavukcuoglu, K., 2024.

Google Reveals Gemini 2, AI Agents, and a Prototype Personal Assistant
Knight, W., 2024.

VELO: A Vector Database-Assisted Cloud-Edge Collaborative LLM QoS Optimization Framework

Yao, Z., Tang, Z., Lou, J., Shen, P. and Jia, W., 2024. arXiv preprint arXiv:2406.13399.

Efficient Estimation of Word Representations in Vector Space

Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. arXiv preprint arXiv:1301.3781.