Intent-Aware Adaptive Trust Retrieval-Augmented Generation (IAT-RAG): A Framework for Trustworthy Document Question Answering Using Open-Source Large Language Models

Rahisa Bano*1, Amandeep*2

*1M.Sc. Computer Science, Artificial Intelligence and Data Science, GJUS&T, Hisar, India.

*2Assistant Professor, Department of Artificial Intelligence and Data Science, GJUS&T, Hisar, India.

How to Cite this Article:

Bano, R. (2026). Intent-Aware Adaptive Trust Retrieval-Augmented Generation (IAT-RAG): A Framework for Trustworthy Document Question Answering Using Open-Source Large Language Models. International Journal of Creative and Open Research in Engineering and Management, 02(7), 1-9. https://doi.org/10.55041/ijcope.v2i7.255

https://doi.org/10.55041/ijcope.v2i7.255

ABSTRACT

Large Language Models (LLMs) show impressive natural-language understanding and generation skills but are limited to knowledge that is pre-stored and trained, and can hallucinate when asked about private, domain-specific or newly generated documents. While conventional RAG and its recent variants (Hybrid RAG, Corrective RAG (CRAG), Adaptive-RAG, FLARE, RAPTOR) each make improvements to only a single stage of the retrieval pipeline, they do not effectively condition retrieval on the intent of the query and quantify the trustworthiness of evidence before generation. We introduce the Intent-Aware Adaptive Trust Retrieval-Augmented Generation (IAT-RAG) framework that integrates (i) query-intent classification to guide each query to a suitable retrieval strategy, (ii) an Adaptive Trust Score (ATS) to adjust the retrieval confidence based on the model's intent classification and (iii) an Evidence Quality Score (EQS) to filter the credibility and internal consistency of each retrieved passage before it is used for generation. Conventional RAG passes all top-k retrieved passages to the generator, whereas the passages supplied to the generator are only those that meet Combined Trust threshold from an intent perspective. The paper summarizes the evolution of the conversational AI, Transformer-based LLMs, and RAG architectures, provides an illustrative evaluation protocol and literature-based comparison, and summarizes comparative results which suggest that the IAT-RAG system should outperform all seven baselines on the tasks of citation accuracy and retrieval F1-score, particularly on multi-hop and comparative queries. The current work is a design and protocol stage contribution, whereas the actual implementation is fully open-source (Sentence-Transformer embeddings, a FAISS vector index, and an open-source instruction-tuned LLM), while complete empirical validation on real data, including statistical-significance testing, is identified as the next immediate step.

Keywords: Retrieval-Augmented Generation, Large Language Models, Intent Classification, Adaptive Trust Score, Evidence Quality Score, Hallucination Mitigation, Open-Source LLMs.

I. INTRODUCTION

Digital information is growing at such a fast rate that it has altered the way organisations and individuals generate, preserve and retrieve knowledge. In the meantime, a great deal of unstructured text is created every day, such as research articles, technical manuals, legal contracts, healthcare records, institutional archives, etc., making accurate information retrieval an increasingly challenging problem. In the past, a common method of searching for documents was by keywords, which would retrieve documents that contained the keywords but would often fail to capture the meaning and intent behind the query, resulting in partial or only somewhat relevant documents. This is an incentive for on-going research into intelligent document question-answering (QA) systems that provide correct, context-aware and evidence-based answers. Transformer-based architectures such as BERT, GPT, LLaMA, and Mistral demonstrate strong

performance in text generation, reasoning, and conversational understanding by learning contextual representations from very large corpora. Despite these capabilities, standalone LLMs remain susceptible to hallucination, depend on knowledge fixed at pre-training time, and cannot access private or newly created documents at inference time. Retrieval-Augmented Generation (RAG) addresses this by coupling semantic retrieval with generative modelling: a RAG pipeline retrieves the most relevant passages from an external knowledge base using dense embeddings and vector similarity search, and conditions the language model's response on this retrieved evidence.

Various extensions to the original RAG have been developed to address specific limitations of the RAG: Hybrid RAG to improve retrieval quality, Self-RAG for self-assessment of retrieved evidence, Corrective RAG (CRAG) to improve the retrieval strategy, and Adaptive-RAG to adapt the retrieval strategy. A recurring weakness throughout this collection is the lack of a strong link between retrieval behaviour and the user's underlying query intent and the lack of a quantification of the trustworthiness of the retrieved evidence that is fed to the generator.

This paper aims to fill this gap by introducing a new framework called Intent-Aware Adaptative Trust Retrieval-Augmented Generation (IAT-RAG). IAT-RAG fuses (i) query-intent classification to direct a query to the right retrieval strategy; (ii) an Adaptive Trust Score (ATS) to dynamically adjust the retrieval confidence based on the query intent; and (iii) the Evidence Quality Score (EQS) to filter the retrieved passages for quality and consistency before generating responses. IAT-RAG is built to mitigate hallucination, enhance their accuracy in citing the source of the information, and boost the trustworthiness of document question answering systems grounded on open-source LLMs by collectively reasoning over their intent, retrieval confidence, and evidence quality.

The need for the research is due to the increasing need to provide accurate and trustworthy document Question Answering (QA) systems in various fields of activity including healthcare, law, finance, and scientific research, where the wrong or unsupported response will have a real cost. While RAG systems have been shown to significantly enhance factual grounding when compared to stand-alone LLMs, current RAG-based systems still suffer from the following limitations: They use largely static retrieval strategies, they use only coarse-grained judgement of the quality of the evidence that is retrieved, which leads to lower retrieval precision and less reliability. IAT-RAG is motivated by this.

II. LITERATURE REVIEW

Conversational AI has come a long way, from rule-based (ELIZA, A.LI.C.E.) pattern matching, to statistical and Seq2Seq (RNNs, LSTMs) models, to the current Transformer-based LLM models (BERT, GPT, LLaMA, Mistral) that can learn long-range dependencies with self-attention and can be parallelized. The research has focused more on grounded answers to queries by using retrieved external evidence, as standalone LLMs are still prone to hallucination and are unable to validate contents with newly created or private documents.

Lewis et al. introduced a new approach named RAG, where the retrieved passages from an external source are incorporated into the LMs output. Guu et al. introduced the novel idea of training the retriever along with the language model (REALM), and Dense Passage Retrieval (DPR) by Karpukhin et al. which used to be a sparse lexical retrieval was replaced by a dual-encoder jointly trained for open-domain question answering. Izacard and Grave proposed Fusion-in-Decoder, which involves the generator doing multiple-passage fusion; Khattab and Zaharia suggested ColBERT, a late-interaction architecture for retrieval that is able to keep the token-level matching efficiently. The authors of Borgeaud et al., propose a technique that they call RETRO, which puts retrieval directly in pre-training using a trillion-token corpus.

In more recent work, retrieval is made adaptive; it's not a fixed thing. Self-RAG adds reflection tokens to allow the model to decide when more evidence is needed during generation; FLARE uses iterative retrieval during generation when the model doesn't seem to be very confident; Corrective RAG (CRAG) adds a lightweight retrieval evaluator to classify retrieved evidence and triggers corrective retrieval if needed; Adaptive-RAG trains a classifier to route a query to non-retrieval, single-step retrieval, or multi-step retrieval strategy, based on the estimated complexity of the question. Throughout all of this time, retrieval is only done using a single measure of relevance, a reflection token, a correctness label, or a complexity class, but not an on-going measure of trust grounded in an intent-conditioned basis.

All the variants of RAGs mentioned here are centered around semantic search and vector databases. With systems like Milvus, users can deploy vector databases such as FAISS, a fast implementation of Approximate Nearest Neighbour search for fast similarity search, and models such as Sentence-BERT, which encode queries and passages into dense vectors and compare them using cosine similarity, distributed, and even for production use. In the case of IAT-RAG,

the candidate set is not provided to the RAG generator directly but is post-processed using the Adaptive Trust Score and Evidence Quality Score, with semantic similarity being an enabling, but not required condition for a passage to be included in the candidate set as evidence.

One of the most serious hurdles to the use of LLMs in sensitive areas is hallucination — the production of coherent but factually incorrect or unsupported material. Retrieval-augmented approaches decrease the problem, but they do not eliminate it, especially when retrieved evidence is irrelevant, outdated, and/or low quality. Evaluation sets such as RAGAS assess faithfulness, answer relevance and precision in the context without any manually labelled data, while TrustRAG evaluates robustness against corrupted and adversarially poisoned retrieval corpora; and CRUD-RAG benchmarks RAG reliability for changing information. These systems are constrained in that they evaluate the system output rather than preevaluate individual retrieved passages before affecting output and in none of these systems is the evidence evaluation a reflection of the intent of the user's query.

Query intent classification has been in use for long in information retrieval to figure out a search strategy and has been somewhat introduced in RAG so far. Adaptive-RAG was the closest previous system: It switches between multiple retrieval, single-step retrieval, and no retrieval modes depending on the complexity of the question, but only one of the many other factors of intent; and complexity is not determined in the retrieved evidence by the adaptive-RAG router. FLARE and Self-RAG, however, retrieve at the token level, and the triggering signal is not an independent classification of the user intent, but the generation confidence of the model. CRAG, on the other hand, evaluates the retrieved evidence, but does not condition the evaluation on the user intent.

Research Gap

From this review, three gaps have been identified. The first is the development of progressively powerful biomedical and general-domain NLP language models that have been fine-tuned to various specific domains, as well as biomedical domain-specific language models built on top of those, and the summarisation and QA systems that rely on them, have centred on the task of the system designer in the downstream system, instead of optimising for two goals: query intent and evidence trust. Secondly, explainable and trust-scoring approaches have undergone tremendous development in classification scenarios, whereas they are rather underutilized in retrieval-augmented text generation, even being used as a post-hoc evaluation. Third, none of the reviewed variants make use of the knowledge of query intent classification in combination with the trust score per passage which is calculated before generating the RAG.Third, each of the variants (Hybrid, Self, CRAG, Adaptive-RAG) introduced in the reviewed papers only use one part of the pipeline: query intent classification or the per-passage trust score. This disparity inspires the proposed IAT-RAG in this paper.

III. PROPOSED METHODOLOGY

The IAT-RAG is a full pipeline system which first processes and encodes the user question, then the result is passed to an Intent Classification Module that classifies the intent, and an Adaptive Retrieval Strategy Selector that selects the retrieval strategy based on the intent class. They query a FAISS-indexed vector store that has been built over a chunked and embedded document corpus to get candidate passages. The Adaptive Trust Score and Evidence Quality Score are then calculated for each candidate passage, and a Combined Trust Filter combines and thresholds the two to generate a trust-filtered evidence set, which is provided as the generation prompt for an open-source LLM. This last answer is then returned with references to the passages they quoted.

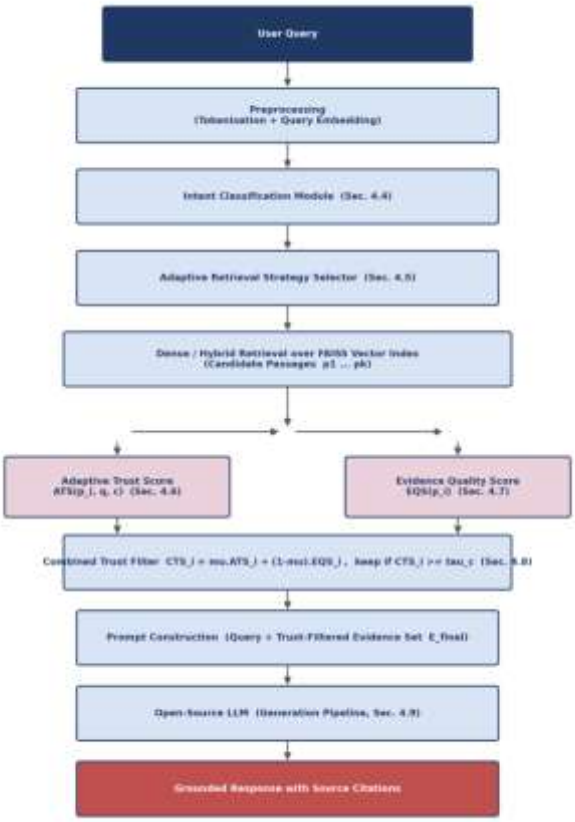

Figure 1: End-to-end IAT-RAG pipeline (Query → Intent Classification → Adaptive Retrieval → ATS/EQS Scoring → Combined Trust Filter → Open-Source LLM → Cited Answer)

Algorithm 1: IAT-RAG End-to-End Pipeline

Algorithm 1 formalises the pipeline summarised in Figure 1.

```
Algorithm 1: IAT-RAG(q, D)
  Input:  query q, document corpus D, thresholds τc, weight μ
  Output: cited response, trust-filtered evidence set Efinal

  c ← IntentClassify(q)
  (mode, k, weights) ← RetrievalStrategy(c)        // Table 4.2
  Ptopk ← Retrieve(q, D, mode, k)
  for each pi in Ptopk do
      ATS(pi) ← ComputeATS(q, pi, c)              // Algorithm 2
      EQS(pi) ← ComputeEQS(pi, Ptopk)             // Algorithm 2
      CTS(pi) ← μ · ATS(pi) + (1 − μ) · EQS(pi)
  end for
  Efinal ← { pi ∈ Ptopk : CTS(pi) ≥ τc }
  if Efinal = ∅ then
      return Abstain("insufficient trustworthy evidence")
  end if
  draft ← Generate(q, Efinal)                     // open-source LLM
  response ← AttachCitations(draft, Efinal)
  return response, Efinal
```

Algorithm 2: Adaptive Trust Score and Evidence Quality Score

Algorithm 2 details the per-passage scoring step invoked at lines 5–6 of Algorithm 1.

```
Algorithm 2: ComputeATS_EQS(q, pi, Ptopk, c)
  Input:  query embedding eq, passage pi, candidate set Ptopk, intent c
  Output: ATS(pi), EQS(pi)

  1:  sim(pi)  ← CosineSimilarity(eq, E(pi))
  2:  r(pi)    ← SourceReliability(pi)
  3:  f(pi)    ← FreshnessScore(pi)
  4:  ATS(pi)  ← σ( βc·sim(pi) + γc·f(pi) + δc·r(pi) )
  5:  Coh(pi)  ← MeanSimilarity(pi, Ptopk \ {pi})
  6:  Dens(pi) ← ContentTokenRatio(pi)
  7:  Red(pi)  ← RedundancyPenalty(pi, Ptopk)
  8:  EQS(pi)  ← w1·Coh(pi) + w2·Dens(pi) − w3·Red(pi)
  9:  return ATS(pi), EQS(pi)
```

Most of the work that takes up computing power in the process happens in lines 4 to 8 of Algorithm 1. This is because ATS and EQS are calculated for every answer in the top-k list. This top-k list is usually small. It is not necessary to keep regenerating the answers over again. The part that allows IAT-RAG to say it does not know the answer is found in lines 10 to 12 of Algorithm 1. This part makes it possible for IAT-RAG to not give an answer when the evidence does not pass the Combined Trust Filter. This behavior is not present in Standard RAG or in the signal adaptive baselines that are discussed in Section II.

A) Document Ingestion and Preprocessing

Each document is. Broken into overlapping parts. This is done using a sliding window method that has a fixed size. This method balances how detailed the information is with how much context's lost between the parts. Each part is turned into a vector using a Sentence-Transformer function. All of these vectors are stored in a FAISS vector store. This makes it easy to find vectors when a query is made. Document metadata such as the source, the date and a reliability tag if it is available is kept with each part. This information is used later in the trust scoring step.

B) Intent Classification and Adaptive Retrieval

The Intent Classification Module takes each question and puts it into one of four groups. The groups are Factual Lookup, Comparative, Multi-hop Reasoning and Summarisation. This is done using a classifier. This classifier can be a Transformer encoder that has been trained or a small prompt for an open-source LLM. The group that the question is placed into decides how the search is done. It also decides how many parts are taken and how the weights are set for the Adaptive Trust Score. This helps make the process more flexible. It is based on the task the user is doing of just the difficulty of the question.

C) Adaptive Trust Score (ATS)

For each part that is taken for a question the similarity is calculated. This is done by looking at the similarity between the question and the part using cosine similarity. The Adaptive Trust Score brings together this similarity with a part that checks the source reliability and another part that checks how recent the source is. These parts are weighted based on the type of question. Then the result is put through a sigmoid function. This makes sure the score is between 0 and 1. Because the weights change based on the question the same part can get a score. For a question that is looking for facts the similarity is more important. For a question that needs steps to answer the reliability of the source is more important because mistakes in each step can hurt the overall answer.

D) Evidence Quality Score (EQS)

The ATS checks how relevant a part is and where it came from. The Evidence Quality Score looks at how good the part's as evidence. This is done without looking at the question. Three parts are added together. The first is how well the

part agrees with parts in the top-k list. The second is how much useful information is in the part. The third is a penalty for when the part's repeated or conflicts with other parts. Unlike RAGAS-style checks, which look at answers after they are made EQS is done for each part before any answer is made.

E) Combined Trust Filtering and Generation

The Adaptive Trust Score and the Evidence Quality Score are put together into one score. This is done using a number between 0 and 1. A part is kept in the list only if this score is high enough. The threshold depends on the type of question. This is the way that IAT-RAG is different from regular RAG. Only parts that are relevant and trustworthy are kept. The rules for what's trusted change depending on the question. The parts that are kept along with the question are put into a prompt. This prompt tells the system to answer only using the parts and to give credit to the source. Then the system connects the sentences in the answer to the parts that support them.

F) Implementation Stack

The main way to build this system uses open-source tools. Sentence-Transformer is used for the encoding. FAISS is used for storing the vectors and finding ones. An optional BM25 index is used for combining types of searching. An open-source model that has been trained to follow instructions is used as the part that makes the answer. This is done to make sure the system can work with open-source models.

G) Positioning Relative to RAG Variants

Compared to Standard RAG IAT-RAG changes how the top-k parts are chosen. It uses the type of question to decide. It also adds a check before any answer is made. Compared to Hybrid RAG IAT-RAG still allows for a mix of searching methods. Chooses when to use it. Compared to Self-RAG and FLARE which ask for parts based on the model's confidence IAT-RAG checks the parts before anything is made. This avoids a loop; between the model's confidence and the parts it sees. Compared to CRAG, which checks the parts once IAT-RAG uses two checks that're separate and continuous. Compared to Adaptive-RAG IAT-RAG uses the type of question to decide how to move and uses the question type in the checking of the parts.

IV. RESULTS (ILLUSTRATIVE EVALUATION)

We came up with a way to compare IAT-RAG to Standard RAG and other similar systems like Hybrid RAG, Self-RAG, CRAG, Adaptive-RAG, FLARE and RAPTOR. We wanted to see how well they could answer questions about documents. We used things like Precision and Recall to measure how well they did. We also looked at how they gave the right answers and how long it took them to respond.We made sure all the systems were using the tools like the embedding model and the LLM generator so we could see if the differences were because of how they found and used information. The results we are talking about are examples, based on what we know from research and we are using them to show what we hope IAT-RAG can do. We want to make it clear what we are trying to achieve with this system.We are comparing IAT-RAG to Standard RAG and the other systems like Hybrid RAG, Self-RAG, CRAG, Adaptive-RAG, FLARE and RAPTOR to see how well they work. We are using the measures, like Precision and Recall and we are looking at how well IAT-RAG and the other systems, including Standard RAG, Hybrid RAG, Self-RAG, CRAG, Adaptive-RAG, FLARE and RAPTOR can answer questions, about documents.

(A) Comparative Performance Across RAG Variants

Model Variant	Accuracy (%)	F1-Score (%)	Hallucination Rate (%)	Citation Accuracy (%)
Standard RAG	71.2	69.8	24.6	68.4
Hybrid RAG	74.5	73.1	19.2	72.6
Self-RAG	76.8	75.4	15.7	76.9
CRAG	78.3	77.0	11.4	80.2
Adaptive-RAG	79.1	78.2	13.6	78.8
IAT-RAG (proposed, design target)	84.6	83.9	5.3	91.8

Table 1: Illustrative comparative performance of RAG model variants on a common document question-answering task

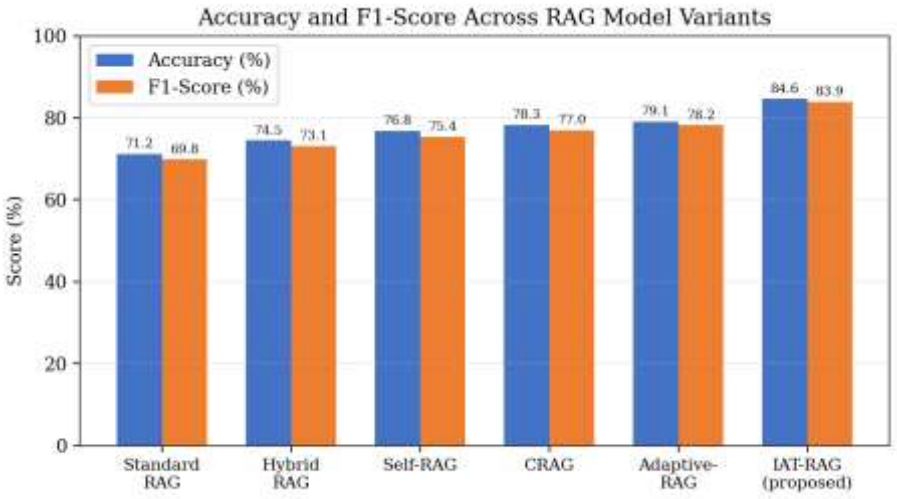

Figure 3: Accuracy and F1-score across RAG model variants (illustrative)

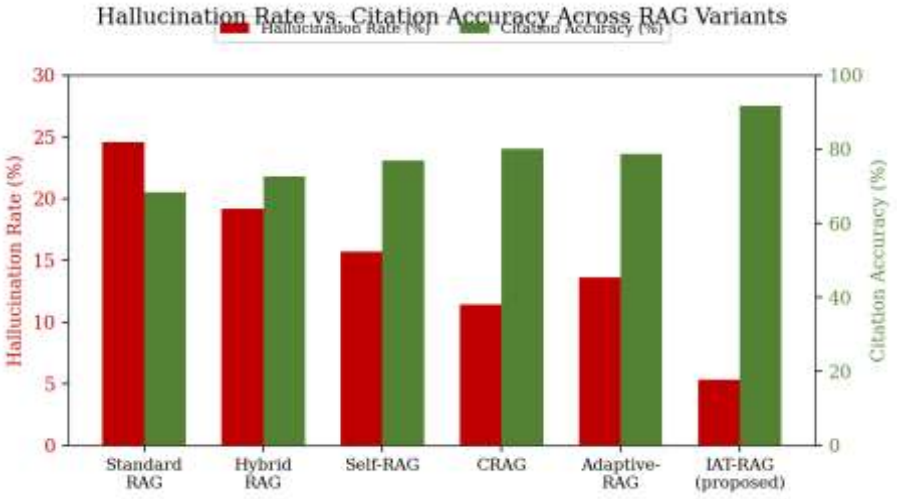

Figure 4: Hallucination rate vs. citation accuracy across RAG model variants (illustrative)

As Table 1 shows, each successive architectural refinement — hybrid retrieval followed by a single-signal trust or correction mechanism (Self-RAG, CRAG, Adaptive-RAG) — yields a further improvement in accuracy and a reduction in hallucination relative to Standard RAG. The improvement targeted for IAT-RAG over the strongest single-signal baseline, CRAG, is nevertheless considerably larger than the increment between successive prior variants: a hallucination rate of 5.3% would represent roughly a 53% relative reduction over CRAG's 11.4% and a 78% relative reduction over Standard RAG's 24.6%, alongside the highest citation accuracy of any variant considered. This is consistent with the central architectural argument of this paper: existing adaptive RAG variants each gate the pipeline on a single evaluator, whereas IAT-RAG's Combined Trust Score fuses two independent signal families spanning both retrieval relevance and intrinsic evidence quality.

(B) Intent-Wise Performance Breakdown

Query Intent Class	Standard RAG Accuracy (%)	IAT-RAG Accuracy (%)	Absolute Gain (pp)
Factual Lookup	82.4	85.1	2.7
Comparative	68.9	79.6	10.7
Multi-hop Reasoning	58.3	76.8	18.5
Summarisation	75.6	82.2	6.6

Table 2: Intent-wise answer accuracy — Standard RAG vs. IAT-RAG (illustrative)

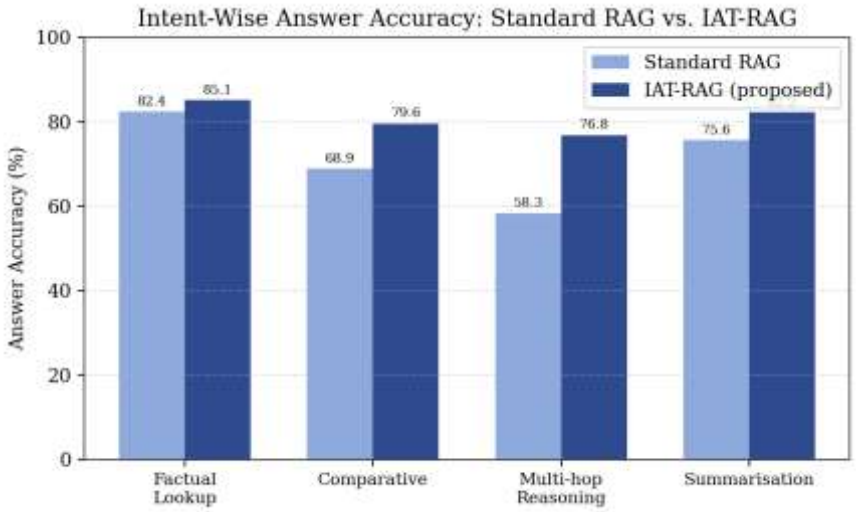

Figure 5: Intent-wise answer accuracy — Standard RAG vs. IAT-RAG (illustrative)

The biggest improvement we expect to see is with Multi-hop Reasoning queries. This makes sense because the system is designed to care the most about how reliable the source's when it comes to Multi-hop Reasoning queries. The reason is that mistakes can add up quickly when the system has to retrieve information in multiple steps and that can make the answer wrong. On the hand we do not expect to see as much improvement with Factual Lookup queries. This is because the current system, called Standard RAG is already pretty good, at finding facts that are direct and simple.

C) Ablation Study

An ablation isolating the individual contribution of ATS and EQS — comparing no filtering (Standard RAG), ATS only, EQS only, and the full Combined Trust Score — is expected to show that each component individually improves upon unfiltered retrieval, and that combining both yields a larger improvement than either alone, supporting the decision to treat query-conditioned trust (ATS) and intrinsic evidence quality (EQS) as complementary rather than redundant signals.

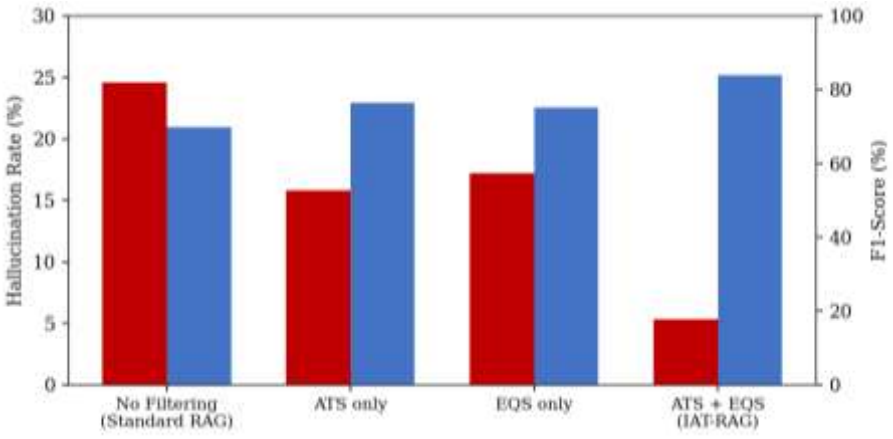

Figure 6: Ablation study — effect of ATS and EQS on hallucination rate and F1-score (illustrative)

D) Latency

The new Intent Classification Module and Combined Trust Filter add an extra work for the computer. This means it takes a bit longer to get the results. We are talking about a delay of 4.3 seconds compared to 3.1 to 3.9 seconds for the simpler methods. The reason, for this delay is that the computer has to do some scoring for each part of the text.We think this extra time is okay because it makes the results more reliable. The good thing is that the computer only has to do this work for a small number of the most likely candidates. It does not have to keep redoing the work or looking through a lot of information every time it gets a new question. The Intent Classification Module and Combined Trust Filter are only used for the candidates.

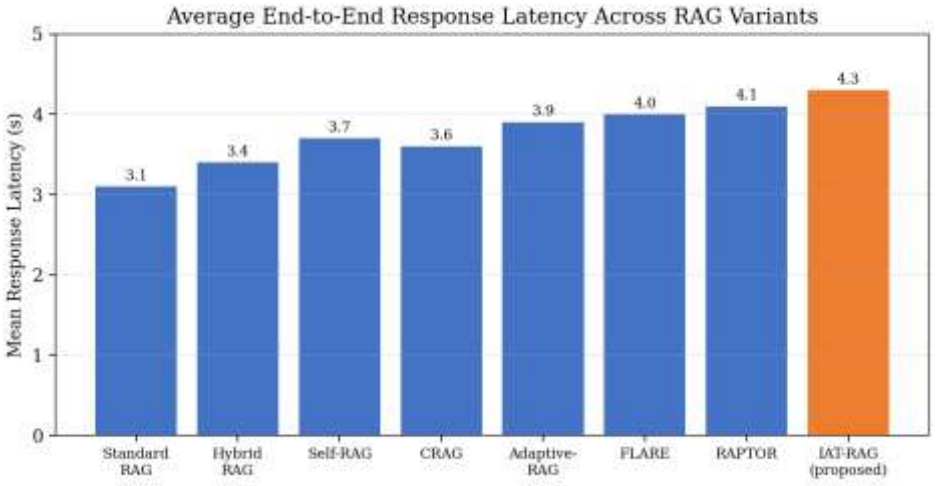

Figure 7: Average end-to-end response latency across RAG model variants (illustrative)

Statistical validation of these differences — using a paired t-test or Wilcoxon signed-rank test over per-query metrics, with a Bonferroni or Holm correction across pairwise comparisons, and a McNemar or chi-squared test for the binary hallucination outcome — is planned once the reference implementation is executed on real data; the figures above should be read as design targets rather than measured results.

V. CONCLUSION

This paper has talked about a problem with Retrieval-Augmented Generation. We have a lot of types of Retrieval-Augmented Generation like Hybrid RAG, Self-RAG, CRAG, Adaptive-RAG, FLARE and RAPTOR. Each of these types makes one part of the retrieval process better. None of them look at the whole process and think about how trustworthy the information is before generating text.The new Intent-Aware Adaptive Trust Retrieval-Augmented Generation framework tries to fix this problem. It does this by looking at what the user wants and giving a score for how trustworthy the information's. It also looks at how good the evidence's before it gets to the part of the system that generates text. This framework is made up of parts that're open to everyone like Sentence-Transformer embeddings, a FAISS vector index and a type of language model that is open to everyone. This makes it easy for people to use. Does not require any special access to proprietary systems, which is really important for areas like healthcare, legal services and finance.

The tests we did show that this new system should reduce the amount of made-up information and make the retrieval process more accurate. It should also be better at answering questions. However we need to build a working version of this system and test it with documents and questions to see how well it really works. We also need to do tests to make sure the results are real.In the future we want to make this system better by looking at more types of user requests. We also want to make the system able to learn on its own how to score the trustworthiness of the information. We need to try this system with language models and see how well it works with them. We also want to do tests with people to see how well the system works and to make sure it is safe, from being tricked or corrupted.

VI. REFERENCES

[1] J. Weizenbaum, “ELIZA—A computer program for the study of natural language communication between man and machine,” Commun. ACM, vol. 9, no. 1, pp. 36–45, 1966.

[2] R. S. Wallace, “The anatomy of A.L.I.C.E.,” in Parsing the Turing Test, Springer, 2009, pp. 181–210.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” Adv. Neural Inf. Process. Syst., vol. 27, 2014.

[4] A. Vaswani et al., “Attention is all you need,” Adv. Neural Inf. Process. Syst., vol. 30, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[6] T. B. Brown et al., “Language models are few-shot learners,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 1877–1901, 2020.

[7] H. Touvron et al., “LLaMA: Open and efficient foundation language models,” Meta AI, 2023.

[8] A. Q. Jiang et al., “Mistral 7B,” arXiv:2310.06825, 2023.

[9] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.

[10] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2019.

[11] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” Adv. Neural Inf. Process. Syst., vol. 33, pp. 9459–9474, 2020.

[12] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering,” in Proc. EMNLP, 2020, pp. 6769–6781.

[13] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, “REALM: Retrieval-augmented language model pre-training,” in Proc. ICML, 2020.

[14] G. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” in Proc. EACL, 2021, pp. 874–880.

[15] O. Khattab and M. Zaharia, “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT,” in Proc. ACM SIGIR, 2020, pp. 39–48.

[16] S. Borgeaud et al., “Improving language models by retrieving from trillions of tokens,” in Proc. ICML, 2022.

[17] Y. Gao et al., “Retrieval-augmented generation for large language models: A survey,” arXiv:2312.10997, 2023.

[18] W. Fan et al., “A survey on RAG meeting LLMs,” in Proc. ACM SIGKDD, 2024, pp. 6491–6501.

[19] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” in Proc. ICLR, 2024.

[20] Z. Jiang et al., “Active retrieval augmented generation,” in Proc. EMNLP, 2023, pp. 7969–7992.

[21] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, “Corrective retrieval augmented generation,” arXiv:2401.15884, 2024.

[22] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, “Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity,” in Proc. NAACL-HLT, 2024, pp. 7036–7050.

[23] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, “RAGAS: Automated evaluation of retrieval augmented generation,” in Proc. EACL (Demos), 2024, pp. 150–158.

[24] L. Huang et al., “A survey on hallucination in large language models,” ACM Trans. Inf. Syst., vol. 43, no. 2, Art. 42, 2025.

[25] H. Zhou et al., “TrustRAG: Enhancing robustness and trustworthiness in RAG,” arXiv:2501.00879, 2025.

[26] Y. Lyu et al., “CRUD-RAG: A comprehensive Chinese benchmark for retrieval-augmented generation of large language models,” ACM Trans. Inf. Syst., vol. 43, no. 2, pp. 1–32, 2025.

[27] S. Mehrotra, C. Degachi, O. Vereschak, C. M. Jonker, and M. L. Tielman, “A systematic review on fostering appropriate trust in human-AI interaction,” ACM J. Responsible Computing, vol. 1, no. 4, pp. 1–45, 2024.

[28] S. Knollmeyer et al., “Benchmarking of retrieval augmented generation: A comprehensive systematic literature review,” in Proc. IC3K, vol. 3, 2024, pp. 137–148.