

Benchmarking Hallucination Mitigation Techniques for Open-Source License Compatibility Reasoning: A Comparative Study of Prompting, RAG, and Fine-Tuning

Himanshi Yenugupalli ¹, Aarti Sanjay Gawai ²

¹ Student, Department of Information Technology and Computer Science, D.G. Ruparel College, Mumbai, Maharashtra, India

² Assistant Professor, Department of Information Technology and Computer Science, D.G. Ruparel College, Mumbai, Maharashtra, India

himanshiyenugupalli@gmail.com¹ aartigawai1611@gmail.com²

Abstract - Large language models are being increasingly utilized for their capacity to reason across software development, but license compatibility reasoning can fail when a model treats a conditional permission as unconditional or provides a conclusion without sufficient evidence. This study evaluates three mitigation strategies: structuring prompting, retrieval augmented generation, and parameter efficient fine-tuning in the context of three-way open-source license compatibility classification. The experiments use a frozen 50-case Ground Truth V2 benchmark for the Gemini Flash 3.6 prompting and RAG conditions, while the Qwen2.5-1.5B Instruct fine-tuning experiment is evaluated on a held-out 10 case set after learning on 40 examples. Prompting was the strongest directly matched configuration at 88.0% accuracy and 0.909 Macro-F1, outperforming the tested RAG configuration at 82.0% accuracy and 0.862 Macro-F1. Both Gemini conditions were weakest on CONDITIONAL cases. LoRA boosted Qwen accuracy from 20.0% to 30.0%, but Macro-F1 was 0.167 and the model developed strong output bias. This study finds that analysis of error structures and evaluation controls is more informative than a simple accuracy metric as a basis for legal reliability.

Key Words: open-source licenses, license compatibility, large language models, retrieval-augmented generation, hallucination mitigation, LoRA

1. INTRODUCTION

Large language models (LLMs) are increasingly being employed in software development tasks that require the

interpretation of technical documentation, the generation of code and the explanation of the implementation of specific features within a software system. The open-source licensing of many software systems presents a challenging scenario for LLMs, as there are various licenses that may be applied to a software system, each with different versions, distributions, secondary licenses and conditions that must be met in order to use the software system. Therefore, an LLM can generate a fluent explanation of a software system's license, yet still incorrectly determine the license that is applied to the software system. Research into the application of LLMs to license conflict analysis and compliance with software licenses has revealed both the potential of these models and their weaknesses in interpreting license terms and providing reliable information regarding license compliance [1] - [3].

The retrieval-augmented generation (RAG) model tackles the issue of hallucination in natural language models by adding explicit information source to the model generation process. The original RAG model combined parametric model knowledge with non-parametric retrieved memory and was motivated in part by the need for more factual, knowledge-grounded generation [12]. Software licensing models could use retrieval to locate the specific clause within a license agreement that governs a particular compatibility mechanism. Recent research has explored the use of RAG models in conjunction with knowledge graphs to improve the explainability and citation-based reasoning of license-checking models [4]. However, the use of retrieval in RAG models does not guarantee that the model will arrive at the correct conclusion; irrelevant, incomplete or poorly connected information can still lead to incorrect conclusions. Also, research on hallucinations has shown that methods for mitigating hallucinations can have negative impacts on other aspects of the model's performance [5] - [8].

The second class of model modifications consists of fine-tuning the pre-trained model. Fine-tuning modifies the model's reaction to certain inputs, rather than modifying the input itself. Methods such as Low-Rank Adaptation (LoRA) allow fine-tuning even with smaller open models. So these models should be analyzed based on the error type structures found during testing, rather than simply the overall accuracy of the model.

The present study investigates three different approaches to open-source license compatibility: prompting, retrieval-augmented generation and LoRA fine-tuning. Each of these methods was tested against a set of three categories: compatible, conditional and incompatible. Two of the models (Gemini and RAG) were evaluated against the same benchmark of 50 license combinations. The third model (Qwen2.5-1.5B-Instruct) was tested against a separate benchmark of 40 license combinations that were used for training the model, as well as 10 license combinations that were held out of the training data. These two benchmarks were evaluated separately from one another.

Also, the study adheres to the terminology limit that the available baseline records show classification errors but not hallucinations. Therefore, the term classification error rate is used in the study to describe the outcomes of the models and hallucination is discussed separately as a motivation for those classifications and as a potential interpretation of those classifications that requires more annotation of the model's outputs.

The research questions would be:

(RQ1) How does the performance of the proposed model compare to the RAG model on the same 50-case benchmark?

(RQ2) What types of failures remain in the proposed model when retrieval is introduced?

(RQ3) Does small-data LoRA fine-tuning improve three-class compatibility reasoning for Qwen2.5-1.5B-Instruct?

(RQ4) What do the observed error patterns suggest for future license-reasoning system design?

2. RELATED WORK

Analysis of open-source licenses has been going on for long time, and in general it is based on license identification, checking of license compatibility using sets of rules, and license compliance tools. Recent work by Cui et al. investigates the use of LLMs for open-source license conflict analysis and notes that general-purpose models struggle with domain-specific license-term reasoning [1]. Related work, OSS-LCAF, views open-source license analysis as a combination of license terms automated extraction and LLM-based conflict reasoning, highlighting the importance of having structured output of the system and adequate task-specific evaluation [2]. From a different angle, LiCoEval assesses the ability of LLMs to provide adequate license information for generated code, which resembles existing open-source code and implementations [3]. The studies reviewed above point out that license reasoning is not a generic question-answering task and requires connection of license textual terms to concrete software and distribution situations.

In order to move beyond the mere use of a model's parameters for generating text, RAG provides a framework for using retrieved non-parametric information in conjunction with a parametric generator to address knowledge-intensive tasks. This area has been further explored in order to mitigate

hallucinations, with retrieval, in-context learning, and parameter-efficient fine-tuning having been tested as separate interventions [5]. The finetune-RAG model treats imperfect retrieval as a training condition rather than treating the retrieved context as if it were perfect [6]. The RAG-HAT model, on the other hand, recognizes that retrieval itself can be vulnerable to hallucinations and follows up the retrieval stage with a tuning pipeline that is able to not only detect but also correct hallucinated content [7]. In its most advanced form, retrieval is treated as a matching or judgment problem and heterogeneous LLM judgments are modeled in order to reduce the incidence of hallucination in a variety of different compliance-oriented services [8].

License-specific RAG work has also highlighted its explainability. Ben Khalifa et al. describe a system combining knowledge graphs, LLMs and RAG to provide automated software-license checking with citation-based explanations [4]. This line of work is particularly crucial to licensing, where a correct label without a supporting clause is less useful than a conclusion whose evidentiary basis can be inspected.

The present study differs from these systems in scope and experimental focus. It does not propose a new retrieval architecture or a new fine-tuning algorithm. Instead, the author freezes a 50-case compatibility benchmark and compares two Gemini inference conditions under the same evaluation, while reporting a separate small-model fine-tuning feasibility experiment. The contribution is therefore comparative and error-focused: the study asks whether the tested interventions actually change the recurring failure mode of the benchmark. This design also makes a methodological distinction between classification correctness and evidence faithfulness. The RAG prompt required the model to record evidence used and evidence gaps, but the preserved RAG response does not retain those fields, so evidence grounding quality cannot be independently audited for the reported run.

Authoritative licensing resources are critical in this context. SPDX provides standardized identifiers, license names, texts and permanent references [9], while the Open Source Initiative maintains a catalog of approved licenses [10]. Fosso logy provides an established software-license compliance scanning toolkit [11]. These resources would show the type of authoritative information that the license reasoning system should favor over web summaries.

3. METHODOLOGY

3.1 Benchmark and classification task

The Benchmark ground truth used for the principal comparison is Ground Truth V2 (i.e. GT-V2). This consists of 50 cases GC-001 through GC-050. The cases are within the scope of software-license compatibility scenarios and have been labeled as one of the three types of cases, such as COMPATIBLE, CONDITIONAL and INCOMPATIBLE.

The class supports in the benchmark for the baseline were 25 COMPATIBLE, 21 CONDITIONAL, and 4 INCOMPATIBLE cases. The total number of correct predictions by the frozen

baseline of Gemini Flash 3.6 was 44 out of 50 cases in the Ground Truth V2 benchmark for the principal comparison.

3.2 Prompting and retrieval-augmented generation

The purpose of the control condition for the baseline is to set a behaviour of the model when it is working with a scenario and it is required to come to a conclusion within the strict guidelines of a standard prompt. We do not aim to write a prompt that tries to get as much out of the model as possible. Instead we write a prompt to test the behaviour of the model in the case described in the scenario above. This behaviour will serve as a good starting point to test interventions on the prompt afterwards.

The RAG condition in the benchmark uses the same 50 test cases and the same decision space of three labels (COMPATIBLE, CONDITIONAL, INCOMPATIBLE). However, the prompt has been modified in a controlled fashion to retrieve evidence, specifically to retrieve the specific authoritative clause or clauses for each of the license mechanisms presented in a scenario. It is preferred that these are retrieved from the official license text itself or from the official organization that issued the license. The model is instructed to report exactly the source (e.g. a specific Section) from which the evidence was retrieved, and to state an evidence gap if the retrieved material does not resolve a material issue in the scenario.

In implementing the RAG (Retrieval-Augmented Generation) approach for the study, we are retrieving compatibility rules in the form of the clause that sets out said compatibility and also the secondary-license(s) that are applicable. More critically, we are also retrieving the conditions that need to be met in the given scenario and which establish whether said conditions have been established in the scenario under consideration. Thus, retrieving a statement that two licenses are compatible in principle is not sufficient; we must also establish whether the scenario in question establishes the conditions for said compatibility.

3.3 Fine-tuning feasibility experiment

This fine-tuning experiment employs the Qwen2.5-1.5B-Instruct as a local open-model feasibility condition. The model is adapted using LoRA on 40 training examples, and evaluated on 10 held-out cases. The held-out set is separated from the training examples, and the final report treats this experiment as a feasibility study rather than as a matched comparison to Gemini.

The Qwen experiment is particularly useful as an error-analysis contrast: the base model gets only 2 out of 10 correct predictions, and the LoRA model gets 3 out of 10, increasing accuracy by 10.0% (20.0% to 30.0%) but retaining Macro-F1 at 0.167. The qualitative evaluation also notes two outputs using the COMPLIANT label, which is not in the required three-label vocabulary. As such, the experiment is interpreted as evidence of behavioural change under small-data adaptation, rather than as evidence of reliable generalization.

3.4 Evaluation metrics

Accuracy is calculated as the proportion of exact label matches. In favor of Macro-F1 is used for overall performance, as it provides equal weighting across the three classes even in the face of uneven support. Per-class precision, recall, and F1 are reported to identify whether an apparent improvement is localized: e.g., whether the increased classification accuracy is driven by recall gains in one label. Confusion matrices and case-level error categories are used for qualitative diagnosis.

For the Gemini prompting condition, accuracy is $44/50 = 88.0\%$, producing a classification error rate of 12.0%. For RAG, accuracy is $41/50 = 82.0\%$, giving an error rate of 18.0%. These error rates are reported as classification measures, and not hallucination rates. No independent hallucination annotation was available in the supplied results. Similarly, in favor of the preserved RAG output that did not retain the intended EVIDENCE USED and EVIDENCE GAP fields, evidence faithfulness is treated as a limitation, rather than as a measured success criterion.

3.5 Experimental controls and comparability

The Gemini comparison is the single strongest controlled comparison in the study, as it is the only one that employs the same 50 cases in both conditions. The Qwen results are not pooled into a single ranking with Gemini, as they are using a separate 10-case held-out set. No confidence intervals or significance tests are reported in the supplied results, and the paper emphasizes descriptive differences, confusion patterns, and qualitative error behaviour: particularly in the Qwen condition, where one additional correct prediction changes the accuracy by 10 percentage points.

4. RESULTS AND DISCUSSION

4.1 Overall performance

Table 1 summarizes the results for the four evaluated configurations. Gemini prompting demonstrates the best direct comparison baseline with 88.0% accuracy and 0.909 Macro-F1. The RAG configuration achieves 82.0% accuracy and 0.862 Macro-F1, which is 6.0 PPL lower in accuracy and 0.047 lower in Macro-F1 than prompting. On the separate Qwen held-out set, the base model achieves 20.0% accuracy and 0.167 Macro-F1, while LoRA reaches 30.0% accuracy but the same Macro-F1.

The class-wise results provide context for the aggregate comparison. Prompting achieves F1 scores of 0.893, 0.833, and 1.000 for COMPATIBLE, CONDITIONAL, and INCOMPATIBLE, respectively. RAG scores are 0.842, 0.743, and 1.000. Thus, both configurations achieve perfect F1 on the smallest INCOMPATIBLE class in the supplied evaluation, but have the greatest room for improvement on the CONDITIONAL class.

Fig. 1. Accuracy values are directly matched only for Gemini prompting and RAG; Qwen values represent a separate fine-tuning feasibility experiment.

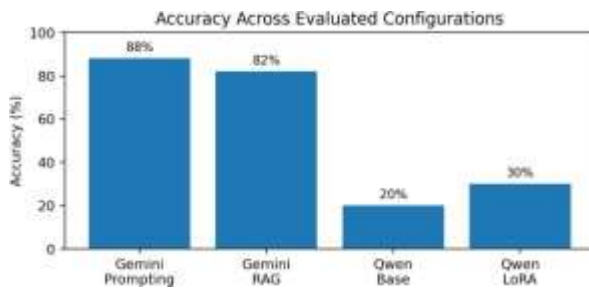


Fig. 1: Accuracy across evaluated configurations

Note: INCOMPATIBLE F1 is 1.000 for both Gemini conditions; the Qwen experiment uses a separate 10-case held-out set and is not a matched comparison with Gemini

Method	Errors	Dominant pattern	Interpretation
Gemini Prompting	6/50	CONDITIONAL → COMPATIBLE	Conditionality resolution
Gemini RAG	9/50	8 C→C; 1 C→CONDITIONAL	Conditionality remains
Qwen Base	8/10	Strong INCOMPATIBLE bias	No CONDITIONAL predictions
Qwen LoRA	7/10	Strong CONDITIONAL bias; 2 invalid labels	Format + class bias

Table 1: Overall classification performance across the evaluated configurations

4.2 Error distribution

The prompting baseline has six errors, all of which are CONDITIONAL → COMPATIBLE. These are cases LC-033, LC-041, LC-044, LC-048, LC-049, and LC-050. An analysis of the baseline errors finds that they are CONDITIONAL cases where the model fails to preserve the distinction between compatibility in principle and compatibility under the stated facts. The cases involve secondary-license and compatibility mechanisms with MPL, EPL, LGPL/AGPL, and EUPL conditions. The model does not recognize that a compatibility mechanism exists; it determines that such a mechanism would apply universally in this scenario regardless of the additional facts provided.

The RAG configuration has nine errors, eight of which are CONDITIONAL → COMPATIBLE and one COMPATIBLE → CONDITIONAL. Thus, RAG changes nine predictions from the baseline, correcting three errors but introducing six new ones. This result suggests that adding retrieved context was insufficient to resolve the fundamental issue with the prompt

design. The model still performed poorly on the most frequent class, although it achieved perfect recall for INCOMPATIBLE. The RAG errors primarily follow the same pattern as the baseline errors. Eight of the nine errors are CONDITIONAL → COMPATIBLE, and one is COMPATIBLE → CONDITIONAL. The COMPATIBLE → CONDITIONAL change was predictable given the model's strong initial bias toward COMPATIBLE. This result suggests that the evidence retrieval process added value in a few cases but was ineffective at addressing the baseline issue. The Qwen errors show a different pattern, as the base model strongly favors INCOMPATIBLE and does not predict CONDITIONAL at all. LoRA causes the model to prefer CONDITIONAL, but it introduces two invalid COMPLIANT predictions. Both fine-tuning approaches cause class imbalance issues that reduce the Macro-F1 score.

Method	N	Correct	Accuracy	Macro-F1	COMP F1	COND F1
Gemini Prompting	50	44	88.0%	0.909	0.893	0.833
Gemini RAG	50	41	82.0%	0.862	0.842	0.743
Qwen Base	10	2	20.0%	0.167	0.250	0.000
Qwen LoRA	10	3	30.0%	0.167	0.000	0.500

Table 2: Dominant observed error patterns

4.3 Prompting versus RAG

The 6-point drop in accuracy from the prompting baseline to the RAG configuration should not be interpreted as evidence that retrieval-augmented prompts are categorically worse than prompting. Rather, this result reflects the specific retrieval setup used in this experiment: the baseline changes by 9 predictions, of which 3 are correct and 6 are incorrect.

One potential explanation for this result is that compatibility evidence is easier to retrieve than conditional applicability evidence. A license may explicitly name a compatible licensing framework, but the final choice between frameworks depends on the scenario-specific facts, a distribution condition, a notice, a version, or an availability factor. As suggested by the baseline error analysis, evidence retrieval should be designed to distinguish between the compatibility mechanism and the applicable scenario-specific condition.

The evidence retrieval also raises questions about the design of the evaluation. The prompt instructed the model to provide EVIDENCE USED and EVIDENCE GAP fields, but they are not present in the preserved response. As a result, this experiment cannot distinguish between a correct prediction that cites an irrelevant passage and an incorrect prediction that cites a relevant passage. This outcome highlights the finding in the

related work that retrieval presence and evidence correctness are distinct evaluation dimensions [4], [6], [7].

4.4 Fine-tuning feasibility

The Qwen results demonstrate why accuracy is not an adequate metric for a small held-out set. An increase from 20.0% to 30.0% accuracy may be reason for optimism, but Macro-F1 actually remained the same at 0.167. The per-class F1 scores show that the base model achieved 0.250, 0.000, and 0.250 for COMPATIBLE, CONDITIONAL, and INCOMPATIBLE, respectively. LoRA increased the CONDITIONAL F1 to 0.500 but reduced the other two to 0.000, making the model more likely to predict any class but achieving poor overall performance. Again, no class imbalance improvement was found. That is also the reason why LoRA reaches a slightly higher accuracy than the baseline model in aggregate.

Changing the model's preference from COMPATIBLE to CONDITIONAL did not result in a consistent pattern across the classes, and instead LoRA's outputs for the most part were in a COMPLIANT format, two of which used an invalid label. A system intended for production use would flag such invalid outputs and send them for human verification. In addition to Classifying Validity, a Parameter-efficient fine-tuning approach for sequence generation must also be able to generate Valid Outputs, in a valid Format, and generalize to Inputs that were not used for fine-tuning.

4.5 Implications for hallucination mitigation

What the results of our experiments into hallucination mitigation show, is that such mitigation has to be seen as a series of reasoning controls rather than a property of a single model. While the prompting baseline did not manage to resolve the fundamental ambiguity of the three-way classification even after having been carefully designed, the amount of additional validation provided by the retrieval of external evidence for the most straightforward cases is substantial. The fine-tuning of the model in the end changed the decision making of the model and became less consistent. Therefore, future research into hallucination mitigation has to explicitly address conditional compatibility and design a prompt or even a training procedure that distinguishes between compatibility in principle and compatibility with the given facts.

A possible prompt structure for such a system would be to first identify the relevant license compatibility mechanism and then extract the conditions that must be met for that mechanism to apply. The model would then analyse the scenario to identify which of these conditions are present and determine whether the mechanism fully applies, is conditional, or is incompatible. This approach aligns with the three-way classification scheme and the benchmark's own reasoning for selecting CONDITIONAL as the default option when the material facts are unknown.

The results also suggest that three distinct evaluation questions should be asked when analysing hallucination mitigation methods: (1) whether the predicted license compatibility status is correct, (2) whether the relevant license clause was interpreted correctly, and (3) whether the evidence actually

supports the predicted status. This study primarily analysed the first question and partially addressed the second by categorizing the types of errors. However, the findings indicate that researchers should separately evaluate the third question by analyzing whether the retrieved evidence actually entails the predicted status. The current study could not address this concern due to the design of the RAG prompt, which did not preserve the retrieved evidence for analysis.

4.6 Practical interpretation

For the considered conditions, structured prompting is the best option if the primary goal is to classify on the frozen benchmark with minimal computational resources. RAG should not be ruled out, as its evidence retrieval mechanism may be useful in subsequent iterations of this work, or when combined with stronger conditional control mechanisms. Finally, parameter-efficient fine-tuning has potential but requires further analysis to ensure that it does not exacerbate the class imbalance issue. The larger 40/10 experiment serves as an initial validation of fine-tuning as a method for adapting open language models to the compliance domain.

The key practical contribution is that the failure in the prompting baseline should not be summarized as "the model hallucinated." A more precise description of the issue is that the model frequently predicted compatibility in principle but failed to preserve the distinction between compatibility in principle and compatibility under the stated facts. This diagnostic insight can be used to improve future prompt designs or train the model to recognize conditional compatibility scenarios.

5. LIMITATIONS

This study has several limitations. First, the principal benchmark contains only 50 cases. While the cases are controlled and directly matched across the Gemini conditions, this small sample size cannot support establishing broad statistical superiority. Second, the Qwen experiment uses only 10 held-out cases and therefore should be treated as a feasibility result rather than a general performance estimate. Third, the supplied RAG evaluation does not preserve the required evidence fields, preventing independent measurement of evidence faithfulness and retrieval causality. Fourth, the supplied results do not contain a separate human hallucination annotation, so classification error rate cannot be equated with hallucination rate. Fifth, the study does not report confidence intervals, significance tests, latency, monetary cost, or a systematic robustness evaluation over unseen license families and versions.

These limitations also dictate the next experimental steps. A larger benchmark should incorporate license-family and version splits, explicit evidence annotations, independent adjudication of compatibility labels, and controlled distractor retrieval. Fine-tuning should be evaluated on a larger held-out set with balanced classes and strict output validation. A future combined RAG-plus-fine-tuning condition could test whether retrieval supplies the required legal evidence while fine-tuning teaches the model how to reason over conditional requirements.

6. CONCLUSIONS

This paper presented a comparative evaluation of prompting, retrieval-augmented generation, and LoRA fine-tuning for open-source license compatibility reasoning. On the matched 50-case Ground Truth V2 benchmark, Gemini Flash 3.6 with standardized prompting achieved 88.0% accuracy and 0.909 Macro-F1, while the evaluated RAG configuration achieved 82.0% accuracy and 0.862 Macro-F1. The principal error pattern in both Gemini conditions was `CONDITIONAL` → `COMPATIBLE`, showing that retrieval did not by itself resolve the distinction between a compatibility mechanism and the conditions required to use it.

The Qwen2.5-1.5B-Instruct feasibility experiment produced a smaller and different result: LoRA increased accuracy from 20.0% to 30.0% on 10 held-out cases, but Macro-F1 remained 0.167 and output behaviour became strongly conditional-biased, including two invalid `COMPLIANT` labels. These findings argue against evaluating mitigation methods solely by aggregate accuracy.

The central contribution of the study is therefore diagnostic. Reliable license reasoning requires a model to reach the correct compatibility decision, interpret the governing clause correctly, and connect that clause to the actual scenario facts. Future work should evaluate these dimensions separately, preserve evidence provenance, and use larger controlled benchmarks. The present results support structured prompting as the strongest tested baseline while identifying conditionality resolution as the clearest target for improved retrieval and fine-tuning.

ACKNOWLEDGEMENT

The author acknowledges the use of the supplied benchmark, experimental reports, evaluation packages, and manuscript template that formed the basis of this study. The reported findings are limited to the experimental conditions documented in those materials.

REFERENCES

1. X. Cui, J. Wu, X. Ling, T. Luo, M. Yang, and W. Ou, "Exploring Large Language Models for Analyzing Open Source License Conflicts: How Far Are We?," Proc. IEEE/ACM 47th International Conference on Software Engineering: Companion Proceedings, 2025, pp. 291–302, doi: 10.1109/ICSE-Companion66252.2025.00083.
2. A. Kahol, A. C. Tummepalli, and P. R. Anish, "OSS-LCAF: Open-Source Software License Conflict Analysis Framework," Proc. IEEE/ACM 47th International Conference on Software Engineering: Companion Proceedings, 2025, pp. 303–314, doi: 10.1109/ICSE-Companion66252.2025.00084.
3. W. Xu, K. Gao, H. He, and M. Zhou, "LiCoEval: Evaluating LLMs on License Compliance in Code Generation," Proc. IEEE/ACM 47th International Conference on Software

Engineering, 2025, pp. 1665–1677, doi: 10.1109/ICSE55347.2025.00052.

4. I. Ben Khalifa, M. Ben Messaoud, and M. Wiem Mkaouer, "Toward Explainable and Automated Software License Checking via RAG and Knowledge Graphs," IEEE Software, 2026, doi: 10.1109/MS.2026.3677583.
5. M. Liang et al., "THaMES: An End-to-End Tool for Hallucination Mitigation and Evaluation in Large Language Models," Advances in Neural Information Processing Systems, 2024.
6. Z. P. Lee, A. Lin, and C. Tan, "Finetune-RAG: Fine-Tuning Language Models to Resist Hallucination in Retrieval-Augmented Generation," arXiv preprint arXiv:2505.10792, 2025.
7. J. Song, X. Wang, J. Zhu, Y. Wu, X. Cheng, R. Zhong, and C. Niu, "RAG-HAT: A Hallucination-Aware Tuning Pipeline for LLM in Retrieval-Augmented Generation," Proc. 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2024, pp. 1548–1558, doi: 10.18653/v1/2024.emnlp-industry.113.
8. M. Hong et al., "RAL2M: Retrieval Augmented Learning-To-Match Against Hallucination in Compliance-Guaranteed Service Systems," arXiv preprint arXiv:2601.02917, 2026.
9. Software Package Data Exchange (SPDX), "SPDX License List," SPDX Project, 2026.
10. Open Source Initiative, "Open Source Licenses," Open Source Initiative, 2026.
11. FOSSology Project, "FOSSology: Open Source License Compliance Software," Linux Foundation/FOSSology Project, 2026.
12. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems 33, 2020.

BIOGRAPHIES



Ms. Himanshi Yenugupalli
M.Sc. Information Technology



Ms. Aarti Sanjay Gawai
M.Sc. Information Technology
Asst. Prof., IT/CS Dept. D G Ruparel
Collage