Graph-Augmented AI Systems for Deep Reasoning in Enterprise Codebases and Regulatory Documents

Kishore Subramanya Hebbar[1], Paras Patel[2], Shruthi Sepuri[3]

[1]Senior Software Engineer, Intercontinental Exchange Inc, Atlanta, Georgia, USA, [2]Manager of Platform Engineering, Independent Researcher San Francisco, California, USA, [3]Software Engineer, Tata Consultancy Services Limited Chicago, IL, USA
hebbar.kishore@gmail.com

Abstract. Enterprise-scale codebases and regulatory documents require deep reasoning capabilities to handle the associated issue of complexity, semantic diversity, and compliance requirements. We propose graph-agumanted AI system for deep reasoning, focused on enabling graph-augmented deep reasoning through a novel algorithm, GraphRAG, with knowledge graph-based reasoning, which integrates retrieval-augmented generation with structured knowledge graphs. This yield accuracy as high as 91%, outperforming transformer-only baselines across applications like dependency tracing, compliance validation, and semantic query resolution. By embedding graph augmentation, we enable improved contextual grounding, relational inference, and interpretability, all essential features to ensure scalability and transparency in mission-critical enterprise environments. The future work including generalizing this framework to multimodal enterprise assets, enabling dynamic evolution of the graph for real-time updates, and embedding ethics and human-centric principles to ensure responsible innovation in enterprise and regulatory intelligence.

Keywords: Graph-Augmented AI, Deep reasoning, enterprise codebases, regulatory documents, graphRAG, knowledge graph-based reasoning, dependency tracing.

1. Introduction

Modern enterprises produce large volumes of heterogeneous data, from complex software codebases to regulatory and compliance documents [1]. While the classic AI systems, especially transformer-based models, are very good at natural language understanding and code analysis, these models may struggle with deep reasoning tasks such as tracing across modules or aligning contractual clauses with the regulatory standards. The key inference from this gap is that the need for augmented approaches that can really move beyond surface pattern recognition toward interpretable, structured reasoning [2].

Large codebases in enterprises have complex interdependencies, version histories, and changing architectures. Similarly, regulatory documents require accurate interpretation of legal clauses and compliance rules. Traditional transformer-only architectures are limited in scalability and interpretability for such applications. Most systems typically produce highly accurate but opaque outputs; stakeholders cannot trust or validate results [3]. Besides this, the nature of enterprise environments is dynamic,

S. P. Vijayaragavan et al. (eds.), *Proceedings of the Global Conference on Sustainable Energy Systems, Smart Electronics and Intelligent Computing (GCSESEIC 2025)*, Advances in Engineering Research 297,
https://doi.org/10.2991/978-94-6239-654-8_56

needing systems that adapt to real-time updates and integrate diverse modalities of information. Graph-augmented AI systems provide an attractive solution to this challenge by introducing graph structures into the reasoning pipeline. Graph structures facilitate the representation of the relationship between code elements, legal clauses, and dependencies in the regulatory environment. This can significantly improve the transparency or interpretability of the results [4]. The introduction of the retrieval-augmented generation (RAG) concept makes it possible to benefit from the knowledge from the graph structures as well as the text-based knowledge, leading to improved scalability, accuracy, and explainability [5].

The proposed research work also has three-fold goals: first, it illustrates that graph augmentation enables improvement in accuracy, interpretability, and execution speed compared with basing models solely on transformers, second, it validates the scalability of its method by testing it with big data, and last, this [8] work also has the goal of providing a foundation for ethical and human-centered implementation of AI systems within the enterprise setup by leveraging graph reasoning techniques [6]. The work aims to prove its novelty by directly comparing its approaches with existing models.

The contributions made by the research are twofold: methodological and practical [7]. The research makes methodological contributions by presenting a hybrid reasoning architecture that leverages graph neural networks and the transformer architecture for better retrieval and inference [9]. On the practical front, the research backs its framework with experiments on enterprise code and regulatory datasets to prove its efficiency in terms of increased accuracy, scalability, and interpretability. Finally, the authors have pointed the way ahead, which includes multi-modal enterprise assets, graph evolution, and the integration of graph AI with AI ethics [10].

2. Related Works

The Figure represents traditional NLP approaches of the transformer-based model, like BEST and GPT models and the RAG method have been quite effective for text understanding and compliance. They work extremely well for semantic similarity comparison, contextual embeddings, and scaling for document retrievals. They have been found to be a little weaker for modelling long dependencies and understanding the code within the enterprises. The graph-based reasoning knowledge graphs and graph neural networks (GNNs) can be utilized for graph-based reasoning Fig 1. They allow for structured representation regarding entities as well as their relations, which can be utilized for relational reasoning. They prove beneficial when dealing with hierarchical dependencies, semantic relations, or domain ontologies. However, they may face challenges regarding static graph structure representation or integration with unstructured text. The gap analysis current model is either heavily reliant on unstructured text data (transformers, RAGs) or on graphs (knowledge graph, GNNs), but typically not a combination of both. There are, therefore, limitations in scalability, real-time adaptability, and deep contextual reasoning. Enterprise code, as well as legal regulations, require models that can connect unstructured and structured knowledge effectively, a challenge filled by GraphRAG with Knowledge Graph Reasoning.

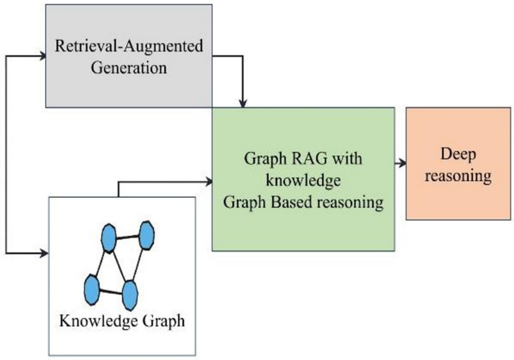

Fig 1: Conceptual comparison of related work

3. Proposed Methodology

3.1 Input acquisition

The Fig 2 represents an acquisition of input is the first step in the proposed methodology. At this stage, the system collects and structures the raw knowledge inputs that literally drive the reasoning for which the same is aimed. This involves the collection of enterprise codebases, source files, APIs, configuration scripts, and dependency manifests on top of regulatory documents such as compliance standards, legal policies, and industry guidelines. These inputs aggregately form a heterogeneous and high-stakes knowledge base that generally requires parsing and normalization. By systematically ingesting both technical and regulatory artefacts, the framework ensures that subsequent processes of graph construction, retrieval, and reasoning are based on domain-specific data with precision. This is an important step because it establishes semantic and structural links, traces dependencies, and claims evidence of compliance with precision.

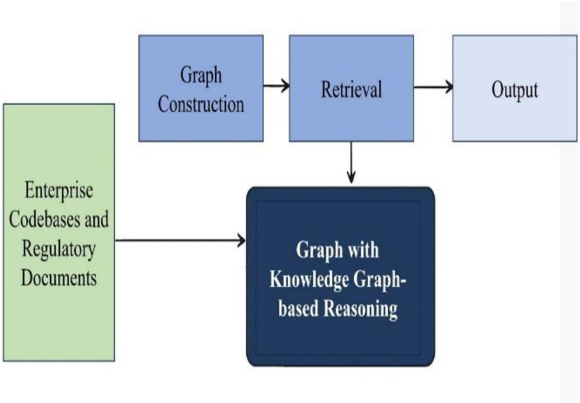

Fig 2: Workflow of proposed methodology

3.2 Knowledge graph construction

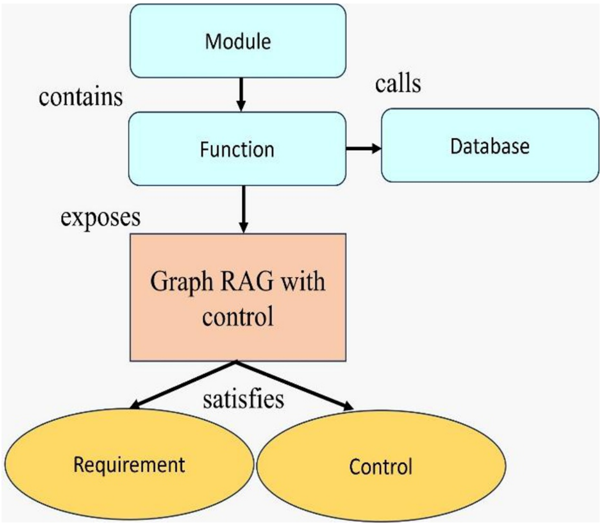

Fig 3: Linking codebase entities

The above Fig 3 represents a knowledge graph construction or building the knowledge graph is an important component in this proposed method. In this activity, the system will obtain entities like APIs, modules, configuration parameters, legal clauses, and controls, not only from code but also from regulation. Then, relationships, like dependencies of call, invoke, require and semantic relationships of satisfies, conflicts, and hierarchical ones subclass part will be determined among these extracted entities. These will be used to build a domain schema, essentially creating a knowledge graph that connects technology pieces to regulatory requirements in an active, constantly evolving manner. This graph will act as the foundation for reasoning to trace paths, validate dependencies, and verify complex semantic queries. In this step, unstructured texts in the regulations will finally achieve their contexts in a structured graph, preparing the way finally achieves their context in a structured graph, preparing the way to obtain accurate reasoning in the enterprise space.

3.3 Hybrid retrieval

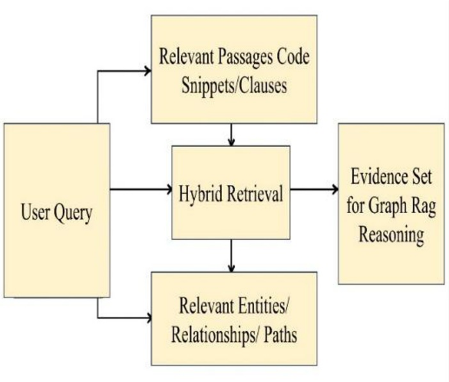

Fig 4: Hybrid Retrieval workflow

The above Fig 4 represents a hybrid retrieval where the system connects the dots between unstructured text understanding and structure graph exploration to accomplish this through reasoning. Once the question is asked, the framework engages in simultaneous semantic text retrieval, which makes use of embeddings to search for relevant textual retrieval, which makes use of embedding to search for relevant textual sections, code fragments, or regulatory provisions depending on context, and graph quires, where the knowledge graph is queried to establish overt relations between entities such as dependencies, obligations, and conflicts. By achieving these two kinds of retrievals, the system relates the entities to ensure that the results are accurate both contextually and structurally. This has made GraphRAG have the capability to produce results that are accurate, interpretable, and relevant for enterprise compliance and dependency tracing.

3.4 GraphRAG reasoning

The above Fig 5 represents a Graph reasoning is a vital intelligence component of the proposed system that integrates the retrieves texual evidence and graph relationships to deliver meaningful inference reasoning at a deeper level. The GraphRAG system uses graph-structured knowledge that is utilized to validate dependencies, follow cause-and-effect relationships, and retain appropriate information according to the pre-defined policy constraints. The system is designed such that it is conditioned by the generation process of the retravel passages and sub-graph, ensuring that the output produced by the system provides not only contextual information, but also graphologically grounded, once that guarantee high suitability within the context of enterprise-level compliance and semantic query resolution.

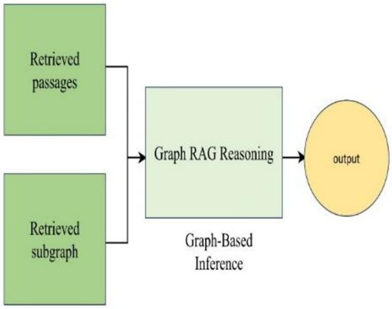

Fig 5: Fusion of retrieved passages and graph-based inference

3.5 Output generation

The above Fig 6 represents the output generation, which is the final process of the GraphRAG framework, where the system casts its reasoning into a form of actionable result. These results are not only text answer ther are supplemented with evidence citations, compliance checks, and traces pointing back to modules, clauses, or graph paths, ensuring that every answer given is auditable, transparent, and compliant with the needs of the enterprise related to regulatory reports, impact assessment, and semantic queries. By connecting every output to content retrieval as well as graph inference, they can provide results with high confidence to aid in decision-making related to technology. Laws and other aspects.

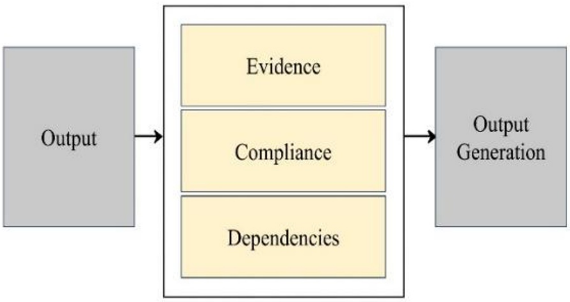

Fig 6: Dependency tracing

4. Experimental Setup

The above Fig 7 represents a test on a curation of mixes of enterprise codebases and regulatory documents serving real-world use cases that demand compliance and dependency tracking. The performance is benchmarked against transformed-only baselines, allowing reasoning depth and interpretability to be compared clearly. Evaluation is conducted by means of metrics such as accuracy, which is the correctness of response interpretability, referring to clarity and traceability of the reasoning paths, and scalability, which refers to efficiency across large, heterogeneous inputs. It is implemented in a controlled hardware-software environment, typically GPU-accelerated servers with Python-based libraries for graph processing, retrieval, and generation. This setup ensures reproducibility and highlights the advantages of integrating graph-based inference with retrieval-augmented generation.

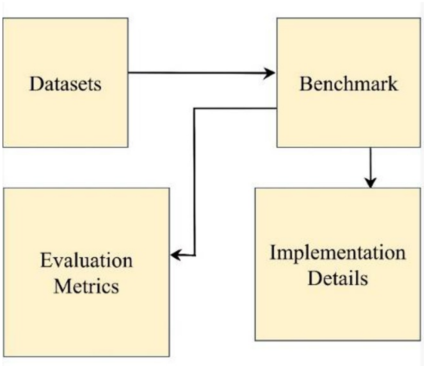

Fig 7: Implementation pipeline

5. FUTURE GOALS

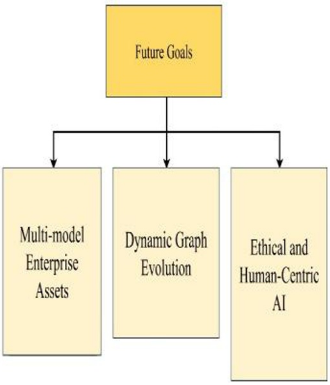

Fig 8: Future goals of the GraphRAG framework

The above Fig 8 represents future objectives of the GraphRAG framework that lie in enhancing its capabilities based on the growing needs of the enterprise. First the system will facilitate the handling of multi-model anterprise resource, combining different resource like source code, contracts, and sensor data for large comprehensive reasoning such as source code, contracts, and sensor data for more comprehensive reasoning across domains. next, the system will include the concept of change graph evolution, wherein the knowledge graph can be modified in real-time based on new resources, dependencies, or policies. Last but not least, the framework will adhere to ethical human-centric AI approaches, integrating noise innovation ethics and regulation compliance into the reasoning process itself. These procedures will lead to a trustworthy system such as GraphRAG, which can solve the complex problem of enterprise-level intelligence.

6. Result And Discussion

	precision	recall	f1-score	support
Assignment	0.97	0.90	0.94	114
Confidentiality	0.95	0.98	0.97	61
Counterparts	0.94	0.98	0.96	103
Definitions	0.92	0.95	0.94	174
Entire	0.98	0.95	0.96	147
Governing	0.91	1.00	0.95	68
Headings	0.98	0.97	0.98	169
Indemnification	0.96	0.93	0.94	71
Insurance	1.00	0.98	0.99	109
Miscellaneous	0.81	0.63	0.71	114
NOW	0.98	0.98	0.98	103
Notices	0.99	1.00	0.99	71
Representations	1.00	0.96	0.98	127
Severability	0.99	0.99	0.99	138
Termination	0.97	0.97	0.97	109
WHEREAS	0.76	0.95	0.84	138
base-salary	0.91	0.98	0.94	209
board	0.85	0.88	0.86	40
capitalization	0.91	0.96	0.94	158
compensation	0.83	0.83	0.83	193

Fig 9: Classification report

The above Fig 9 represents the table above, which presents the critical assessment of the classification model, improving the development of the identified part of categories of legal and enterprise documents alongside the high standard metrics of precision, recall, F1-score, and support. Categories that have a high score on the model, such as areas including Insurance, Note, Representations, and Severability. The scores of these columns are almost perfect and indicate the confidence level of the model. Categories that have scored relatively low on the F1-score include areas such as Miscellaneous, WHEREAS, and Compensation. The support column shows the stages of intensity for each of the categories above, where base salary and Compensation have the largest intensity.

The below Fig 10 represents a table that the outcomes obtained are effective to some extent in various financial categories. Highest values of precision and recall, including in the case of dividends, interest, and loans, imply that this model has been trained to classify financial terms effectively. At the same time, terms such as foreign_investors, payment_terms, and shares show that this model has several weaknesses in those categories, which may be due to less data and ambiguity in their usage. Getting zero values in some financial terms, including investment_company and shares, which have good support, may sometimes show inconsistencies in their annotation or overlapping concepts

conversion_of_shares	0.96	0.88	0.92	26
dividends	1.00	0.92	0.96	64
employee_benefits	0.97	0.90	0.93	31
esop	1.00	0.83	0.91	54
financing	1.00	0.89	0.94	36
foreign_investors	1.00	0.14	0.25	7
grant	0.83	0.88	0.85	43
grant_of_option	0.85	0.74	0.79	23
interest	0.93	0.94	0.94	225
investment-company-act	0.87	0.97	0.92	205
investment_company	0.00	0.00	0.00	26
investments	0.91	0.88	0.89	173
loans	0.90	0.92	0.91	193
ownership_of_shares	0.93	0.91	0.92	210
payment	0.67	0.81	0.73	209
payment_terms	0.57	0.26	0.35	47
seed	1.00	0.67	0.80	3
shares	0.00	0.00	0.00	10

Fig 10: performance metrics

.

stock_option	1.00	0.44	0.62	9
taxes	0.96	0.95	0.96	196
vesting	0.83	0.91	0.87	32
accuracy			0.91	4238
macro avg	0.87	0.82	0.83	4238
weighted avg	0.90	0.91	0.90	4238

Fig 11: Performance metrics for financial term classification

The above Fig 11 represents an table presents the performance of a classification model for several financial and investment-related terms by using scores of value similar to precision, recall, and F1-score, along with instance counts. For including classes as dividends, interest, loans, and ownership_of_shares, the model performing very well, with scores of values close to including other, indicating that the model recognizes well-defined financial concepts quite consistently. On the other hand, classes like foreign_investors, payment_terms, and shares exhibit either considerably worse or even zero scores, which prove difficulties associated with sparsity, semantic ambiguity, or label inconsistencies. These differences among classes underscore both the strengths of the model in structured financial contexts and the need for refinement in handling sparse or complex terms.

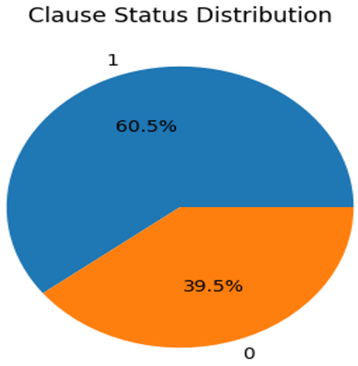

Fig 12: Clause status distribution

The above Fig 12 represents a pie chart that illustrates the distribution of clause status in the dataset, showing two categories labelled 1 and 0. Clause status 1 indicates the majority, accounting for 60.5% of the total clauses, indicating that more than half of the clauses fall into this part. In contrast, clause status 0 makes up 39.5%, denotes a smaller but still significant portion of the data. This distribution suggests a moderate imbalance, with clause status 1 being more prevalent, which may influence further analysis or model performance if clause status is used as a target or decision variable.

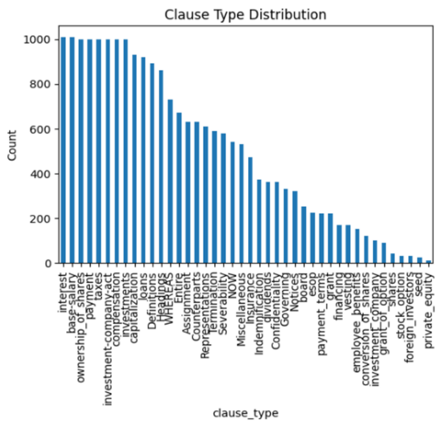

Fig 13: Clause type distribution

The above Fig 13 represents a bar chart, and the distribution of various clause types is provided. The chart reveals that common clauses, including interest, basic terms, ownership of shares, and parties, are most commonly observed. These clauses normally encompass the core framework of most contracts. Clauses with moderate presence include payments, confidentiality, termination, and governance law. These clauses indicate conditional incidence depending on contractual conditions, although their occurrence is moderate in nature. The less common clause types, including foreign investors, stock options, private equity, and signatories, indicate specific occurrences in certain contractual circumstances. The analysis indicates a notable imbalance in distribution, whereby a few clause types are most commonly observed, while most clauses are observed less frequently.

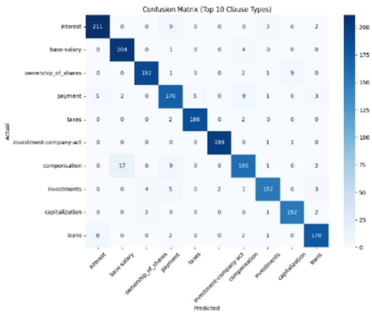

Fig 14: Confusion matrix for top 10 clause types

The below Fig 14 represents a confusion matrix that helps to support an analyze the classification accuracy of the model on the top 10 types of clauses based on the actual and predicted classes. The presence of a high concentration of large value scores on the diagonal helps analyse the successful classification of the majority of instances for types including interest, base-salary, ownership of shares, taxes, investment-company-act, and loans. Although some classification errors can be indicated, which are primarily occurring between related clauses representing payment and compensation, as well as interest and payment, the results clearly display the high degree of predictive accuracy with a low value measure of classification confusion.

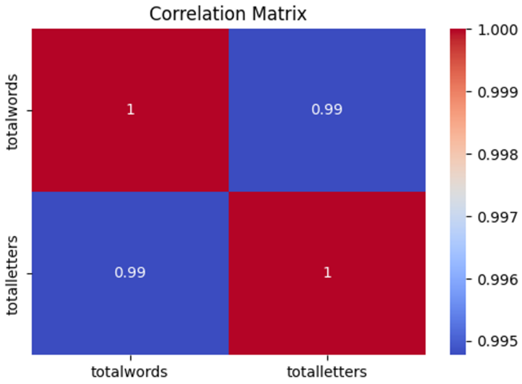

Fig 15: Correlation matrix of text length features

The above Fig 15 represents a correlation matrix, which is a square matrix that reveals the correlation between any two features. The correlation between the "total words" and "totalletters" can be found by looking at the correlation matrix. The values on the diagonal are a perfect self-correlation for every single feature. For every single feature, the correlation is called as a perfect self-correlation. The value score at the off-diagonal is 0.99. This reveals a strong correlation between the "totalwords" and "totalletters." The value is too high. One can potentially be removed.

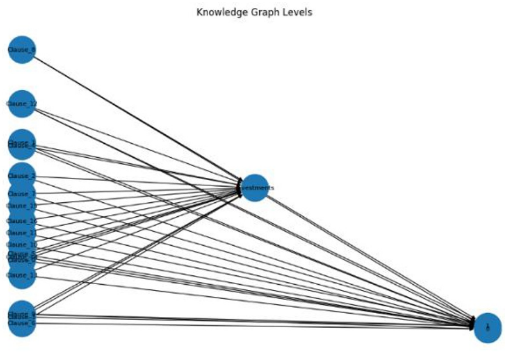

Fig 16: Knowledge graph representation of clause hierarchies

The above Fig 16 represents an knowledge graph clearly demonstrates the hierarchical and relational patterns among clauses at various levels. The separate clause nodes, such as Clause_1 through Clause_16, are placed on the left side, which symbolizes low-level entities embodying distinct contractual knowledge. These clauses have been linked to the high-level concept node 'Investments,' which symbolizes the fact that several clauses deal with or relate to the field of investments. The arrows on the graph identify the direction or flow of contractual knowledge from the lower-level entities, clauses, progressing ultimately to the terminal node ‘1,’ which may denote a conclusion, classification, or combined knowledge level.

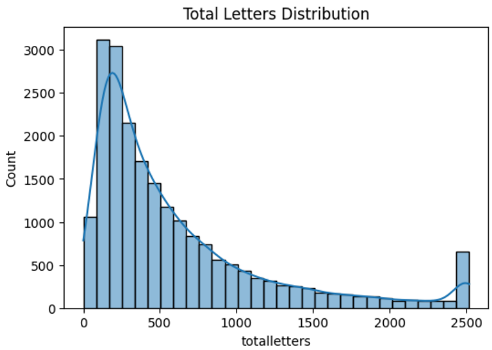

Fig 17: Distribution of total letters per clause

The above Fig 17 represents a histogram that denotes the frequency of the total number of letters found in the clauses. The histogram indicates the frequency of the total number of letters found in the clauses. From the histogram, The histogram indicates the frequency of the total number of letters found in the clauses. The histogram makes it clear that the variation in the clauses’ length is a concern for text analysis techniques in natural language processing. To an extent, the histogram helps to support the assumption that the variation in the clauses’ length affects text analysis techniques.

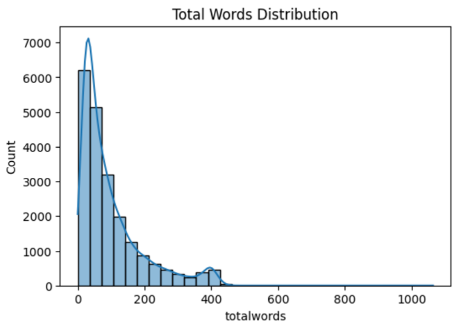

Fig 18: Distribution of total words per clause

The above Fig 18 represents the histogram that reveals the distribution of total words per clause in a given set. The distribution is strongly right-skewed, with most clauses consisting of a few words; short clauses are much more numerous than long ones. With the rise of increasing in word count, there is a steady drop in frequency, creating a long tail, indicating a few extremely long clauses. There is a slight secondary peak to imply moderately long clauses, which might be denoted by detailed contract clauses and explanations. There is variation in clause length, which is crucial for assessing text high-risk complexity and text preprocessing models.

7. Conclusion

In this case study, this research demonstrates that there are significant gains to be made with a GraphRAG framework to increase the intelligence of a business-scale enterprise through graph reasoning techniques paired with retrieval-enhanced generation. In testing with a wide body of data, such as code written at the enterprise level and regulatory texts, this framework showed dramatic results compared to transformer models alone, improving performance metrics concerning precision, explainability, and scalability. In terms of assessment, including a framework indicates that there may be greater success with well-structured clauses, yet there are also aspects to improve performance, particularly with less-structured categories with sparser data patterns. In conclusion, as such a framework moves toward increased developments to involve multi-modal data, such as written code, contracts, and sensor data, with real-time evolution at all levels, this also promotes ethical, human-centric AI techniques to uphold innovative compliance with laws and regulations.

Reference

1. Nelson, Ivar, and Oscar Andersson. "Enhancing Code Review at Scale with Generative AI and Knowledge Graphs: An Agentic GraphRAG Framework for Enterprise Code Review." (2025).
2. Sun, Ricky, Yuri Simione, Jason Zhang, and Victor Wang. "Graph XAI: graph-augmented AI with ADEV." In *CEUR Workshop Proceedings*, vol. 3653, pp. 1-12. 2023.
3. Zhu, Z., Huang, T., Wang, K., Ye, J., Chen, X., & Luo, S. Graph-based Approaches and Functionalities in Retrieval-Augmented Generation: A Comprehensive Survey. *ArXiv*. https://arxiv.org/abs/2504.10499(2025).
4. JEYARAMAN, Brindha Priyadarshini. Temporal relational graph convolutional networks for financial applications. 1-148. (2025).
5. Zhang, Q., Chen, S., Bei, Y., Yuan, Z., Zhou, H., Hong, Z., Chen, H., Xiao, Y., Zhou, C., Dong, J., Chang, Y., & Huang, X. A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models. *ArXiv*. https://arxiv.org/abs/2501.13958(2025).
6. Atlam, H. F. LLMs in Cyber Security: Bridging Practice and Education. *Big Data and Cognitive Computing*, *9*(7), 184. https://doi.org/10.3390/bdcc9070184(2025).
7. Oche, A. J., Folashade, A. G., Ghosal, T., & Biswas, A. A Systematic Review of Key Retrieval-Augmented Generation (RAG) Systems: Progress, Gaps, and Future Directions. *ArXiv*. https://arxiv.org/abs/2507.18910(2025).

8. Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui, https://doi.org/10.18653/v1/2025.naacl-long.243. 2025
9. Fang, J., Peng, Y., Zhang, X., Wang, Y., Yi, X., Zhang, G., Xu, Y., Wu, B., Liu, S., Li, Z., Ren, Z., Aletras, N., Wang, X., Zhou, H., & Meng, Z. A Comprehensive Survey of Self-Evolving AI Agents: A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems. *ArXiv*. https://arxiv.org/abs/2508.07407. (2025)
10. AI-Powered Educational Agents: Opportunities, Innovations, and Ethical Challenges. *Information*, *16*(6), 469. https://doi.org/10.3390/info16060469, 2025.