ISSN: 3071-186X Volume 01, Issue 05, June, 2026

Scienta Press Journals

American Innovator
Journal of Emerging Technologies and Research

A Markov-Decision Formalization of Context Selection and a Hybrid Text–Graph Knowledge Model for Retrieval-Augmented Legal Question Answering

Odilbek Askaraliyev[1], Abdurakhmon Sharifbaev[2],
[1]Sarbon university, Tashkent, Uzbekistan
[2]Tashkent University of Information Technologies named after Muhammad al-Khwarizmi
oasqaraliyev77@gmail.com, mr.sharifbayev@mail.ru

Abstract

Retrieval-Augmented Generation (RAG) couples a large language model with an external knowledge store, yet the quality of its answers is dominated by an upstream, under-formalized step: *which* context to retrieve. This paper addresses two gaps. First, we give a precise formalization of relevant-context selection as a constrained information-maximization problem and reduce it to a Markov Decision Process (MDP), passing through a non-observable context-utility functional, a computable surrogate that combines relevance, diversity and graph coherence, and an analysis of its diminishing-returns structure that justifies both greedy and learned policies. Second, we propose a hybrid text–graph model of documented information that unifies a lexical, a vector and a graph level over a heterogeneous text-attributed graph, together with the database-structuring principles that make the three levels jointly searchable. We instantiate the model on the national legislation database of the Republic of Uzbekistan, building a typed three-level graph (documents, regulated entities, defined concepts) of 207 nodes and 1,718 typed edges directly from act cross-references and definition articles. On four multi-hop question-answering benchmarks and a 1,197-document legislative corpus, the resulting method

(GraphRAG-RL) reaches an F1 of up to 0.832 and a domain accuracy of 91.3%, improving over naive, graph and reinforcement-learning RAG baselines at a moderate retrieval cost.

Keywords: *Retrieval-Augmented Generation, Markov Decision Process, Knowledge Graphs, Graph Neural Networks, Reinforcement Learning, Legal Information Retrieval*

Introduction

The volume of legal and parliamentary information grows faster than it can be read. Large Language Models (LLMs) can phrase fluent answers to complex legal queries, yet without a disciplined retrieval stage they hallucinate and cannot trace claims to their sources. Retrieval-Augmented Generation (RAG) [1] mitigates this by conditioning generation on passages pulled from an external store. In practice, however, the retrieval stage is treated heuristically: candidates are ranked by similarity, the top-k are taken and concatenated. For legislative text — dense with inter-article references, hierarchies of normative acts, temporal amendments and strict traceability requirements — locally "most similar" passages rarely form a globally coherent and sufficient context.

This paper contributes two elements that, together, turn context selection from a heuristic into a learnable, structurally informed decision process. First, *a Markov-decision formalization of context selection*: we define a context-utility functional, derive a computable surrogate $\widehat{U}(q, C)$ that augments per-passage relevance with a diversity and a graph-coherence term, analyze its monotonicity and diminishing-returns properties, and recast sequential retrieval as an MDP $\mathcal{M}_{\text{ret}} = \langle S, A, P, R \rangle$ whose policy can be trained by reinforcement learning. Second, *a hybrid text–graph knowledge model*: we represent the knowledge base as a heterogeneous text-attributed graph integrating a lexical, a vector and a graph level, give its node/edge schema for the legal domain, and describe the database-structuring principles that make a single query traverse all three levels.

We ground the model empirically by constructing a real three-level graph from a national legislative corpus [2], and we report end-to-end results against strong baselines. The remainder of the paper reviews

related work, develops the methodology, describes the experimental setup, discusses results, and concludes with future directions.

Related Work

RAG and adaptive retrieval. Dense retrieval [3] and lexical scoring [4] provide candidate passages; ReAct [5] and Self-RAG [6] interleave retrieval with reasoning and self-critique, partially accounting for the downstream effect of a retrieval decision. A persistent gap remains between the objective a retriever is trained on and the metric by which the final answer is judged.

Graph-based RAG. GraphRAG [7], LightRAG [8] and HippoRAG [9] show that organizing the corpus as a graph and retrieving connected subgraphs improves recall and reduces noise; a recent survey [10] frames the pipeline as graph indexing, graph retrieval and graph-enhanced generation. Graph Neural Networks — GCN [11], GraphSAGE [12], GAT [13] and ChebNet [14] — supply the node and subgraph encoders.

Reinforcement learning for retrieval. R1-Searcher [15] and related work optimize a search policy with reinforcement learning, typically with GRPO-style group-relative advantages [16]. The retrieval/generation split is naturally modelled as sequential decision making, for which Markov decision processes are the standard formalism [17]. Our formalization clarifies *what* such a policy optimizes and provides a structurally informed surrogate utility and a reward defined at the level of selected passages, decoupled from the frozen generator.

Legal information retrieval. Legislative corpora differ from open-web text in ways that bear directly on retrieval: a strict hierarchy of acts, dense inter-article references, amendments and repeals that change a norm over time, and a requirement that every answer be traceable to a specific norm. These properties make a typed knowledge graph a natural backbone and motivate the coherence- and coverage-aware selection studied here.

Methodology

We develop a formalization of context selection and a hybrid text–graph knowledge model over which an adaptive policy operates. Figure 1 shows the overall architecture.

Problem setting and constrained selection. Let Q be the set of user queries and $D = \{d_1, \dots, d_N\}$ the set of elementary knowledge units (documents, paragraphs, fragments, triples). A RAG system is the

composition of a retriever $R_\phi: (q, \mathcal{K}) \mapsto C(q) \subseteq D$ and a generator $G_\theta: (q, C(q)) \mapsto \hat{y}$, with answer quality measured by a functional $M(\hat{y}, y^\star) \in \mathbb{R}$ such as token-overlap F1, semantic similarity, or an LLM-as-judge score. With the generator fixed (a pretrained API model), the retrieval objective is

$$\max_{\phi} \mathbb{E}_{(q,y^\star)\sim\mathcal{D}}\big[\, M\big(G_\theta(q, R_\phi(q,\mathcal{K})), y^\star\big)\,\big].$$

Retrieval is bounded: the generator admits at most $L_{\max}$ context tokens and $K_{\max}$ fragments. With $\ell(\cdot)$ the fragment length, the admissible contexts are $\mathcal{C}_{\text{adm}}(q) = \{C \subseteq D : \sum_{c\in C} \ell\,(c) \le L_{\max},\ |C| \le K_{\max}\}$, and the selection problem is the constrained maximization

$$C^\star(q) = \arg\max_{C\in\mathcal{C}_{\text{adm}}(q)} \mathbb{E}[M(G_\theta(q,C), y^\star) \mid q].$$

Context utility and a computable surrogate. Define the non-observable context utility $U(q,C) = \mathbb{E}[M(G_\theta(q,C), y^\star) \mid q]$. Because U requires invoking G_θ, direct optimization is intractable: the number of admissible subsets grows exponentially in N. We replace U by a surrogate that depends only on available signals — per-fragment relevance, embedding statistics and graph structure:

$$\widehat{U}(q,C) = \sum_{c\in C} \rho\,(q,c) + \lambda_{\text{div}}\,\mathrm{Div}(C) + \lambda_{\text{coh}}\,\mathrm{Coh}(C,G),$$

where $\rho(q,c) \in [0,1]$ is a relevance score (cosine similarity of dense representations) and $\lambda_{\text{div}}, \lambda_{\text{coh}} \ge 0$ balance the terms. Diversity is the mean pairwise embedding distance, and coherence is the share of fragments in the largest connected component of the induced subgraph:

$$\mathrm{Div}(C) = \frac{2}{|C|(|C|-1)} \sum_{c_i,c_j\in C,\, i<j} \Big(1 - \cos(e_{c_i}, e_{c_j})\Big), \qquad \mathrm{Coh}(C,G) = \frac{|\mathrm{LCC}(C,G)|}{|C|}.$$

Marginal gain and greedy selection. The marginal gain of adding d to C decomposes as $\Delta(d \mid C) = \rho(q,d) + \lambda_{\text{div}}\Delta_{\text{div}}(d \mid C) + \lambda_{\text{coh}}\Delta_{\text{coh}}(d \mid C)$. For the surrogate above, (i) the relevance term is non-decreasing under additions when $\rho \ge 0$; (ii) for fixed $|C|$, $\Delta_{\text{div}}(d \mid C)$ shrinks as d grows more similar to the selected set (diminishing marginal informativeness); and (iii) $\mathrm{Coh}(C,G)$ does not decrease when d is linked to C in G. The relevance and diversity components together behave approximately like a monotone submodular set function, for which a cardinality-constrained greedy

algorithm attains a $(1-1/e)$ approximation [18]. This justifies the following greedy selector.

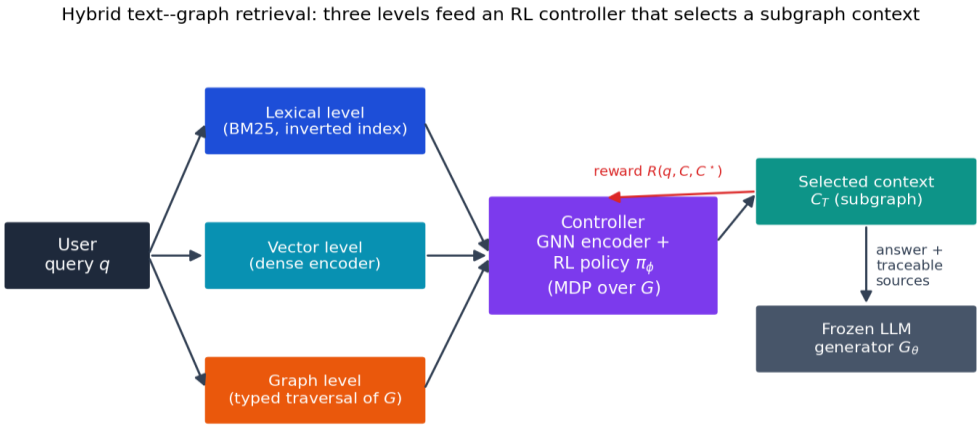

Fig. 1 The hybrid text–graph retrieval architecture. A query is dispatched to the lexical, vector and graph levels; a GNN + RL controller selects a subgraph context that a frozen LLM consumes. During training the controller is optimized by a fragment-level reward $R(q, C, C^{\star})$.

Algorithm 1: Greedy surrogate context selection.

Input: query q, candidates D_{cand}, graph G, budgets $L_{\max}, K_{\max}$, thresholds $\tau_{\min}, \tau_{\text{stop}}$.

1. Initialize $C \leftarrow \varnothing$ and prune candidates to $D_{\text{cand}} = \{d : \rho(q,d) \geq \tau_{\min}\}$.
2. While $|C| < K_{\max}$ and $D_{\text{cand}} \neq \varnothing$: select $d^{\star} = \operatorname{argmax}_d \Delta(d \mid C)$; stop if adding it would exceed $L_{\max}$ or if $\Delta(d^{\star} \mid C) < \tau_{\text{stop}}$; otherwise add $d^{\star}$ to C and remove it from D_{cand}.
3. Return C.

The greedy rule optimizes the surrogate $\widehat{U}$ rather than the true U, uses only local features, and fixes $\lambda_{\text{div}}, \lambda_{\text{coh}}, \tau_{\text{stop}}$ in advance, which motivates a learned, adaptive policy.

The retrieval MDP. We model sequential retrieval as $\mathcal{M}_{\text{ret}} = \langle S, A, P, R \rangle$. A state $s_t = (q, C_t, G)$ holds the query, the fragments selected so far and the knowledge graph. An action a_t selects a fragment $d \in D \backslash C_t$, traverses an edge of G to expand the candidate frontier, or terminates (Stop). The transition is the deterministic update $C_{t+1} = C_t \cup \{d_{t+1}\}$. The reward is defined on the final set C_T relative to the gold set $C^{\star}$ (below). An episode terminates when $|C_t| = K_{\max}$, $\sum_{c \in C_t} \ell(c) \geq L_{\max}$, or $a_t = \text{Stop}$. The objective becomes optimization of the policy $\pi_{\phi}(a_t \mid s_t)$:

$$\max_{\phi} \mathbb{E}_{(q,y^\star)\sim\mathcal{D}}\left[M(G_\theta(q, C_T^{\pi_\phi}(q)), y^\star)\right].$$

Hybrid text–graph knowledge model. We represent the knowledge base as a directed, typed, text-attributed graph $G = (V, E, X)$ with $E \subseteq V \times R \times V$, where V holds documents, structural elements, domain entities and interaction records; E carries typed relations $r \in R$ (*refers_to*, *amends*, *repeals*, *regulates*, *defines*, *contains*); and X holds the text and numeric attributes of nodes and edges. A single query traverses three jointly indexed levels: a **lexical level** (inverted index, BM25) for exact matches on article numbers and fixed legal phrasings; a **vector level** (multilingual transformer encoder) for semantic proximity; and a **graph level** (typed traversal of G) that returns a structured subgraph neighborhood rather than a bag of fragments. The store is organized so the three levels share node identities: a fragment node carries its postings, its dense embedding and its position in G. Table 1 gives the node and edge schema for the legal domain.

Table 1 Node and edge schema of the heterogeneous knowledge graph.

Node type	**Description**
Document	law, code, by-law, regulation
Structural element	chapter, article, part, item, sub-item
Entity	authority, office, procedure, legal category
Concept	legal term, definition, legal construct
Edge type	**Semantics**
contains	a document includes a structural element
refers_to	inter-article / inter-document reference
amends	a norm specifies or extends another norm
repeals	an act terminates another act or norm
regulates	a norm sets rules for an entity
defines	an article introduces a legal definition

Policy navigation over the three levels. At step t the policy is in state $s_t = (q, C_t, G)$ and chooses an action from $A = A_{\text{lex}} \cup A_{\text{sem}} \cup A_{\text{graph}} \cup \{\text{Stop}\}$. A lexical action issues a sub-query and returns its matches; a vector action returns the nearest fragments; a graph action expands the current subgraph. Let $V_t = \bigcup_{c\in C_t} \mu(c)$ be the nodes covered by the current context, where $\mu(c)$ maps a fragment to its nodes. The reachable frontier is

$$F_t = \{v \in V \backslash V_t \mid \exists\, u \in V_t,\ r \in R:\ (u, r, v) \in E\}.$$

A graph action picks a relation r and traversal parameters; the executor returns the part of the frontier reachable by edges of type r, and the context grows by the associated fragments, $C_{t+1} = C_t \cup \{c \in D : \mu(c) \cap F_t^{(r)} \neq \varnothing\}$, whereas lexical and vector actions add their results directly. The policy thus selects, at each step, the level best suited to the information need.

Reward function and training. The reward is defined on the selected fragment set, not on the generated answer, which removes the credit-assignment ambiguity of end-to-end RAG-RL:

$$R(q, C, C^\star) = \alpha\, R_{\text{chunks}}(C, C^\star) + \beta\, R_{\text{coverage}}(C, G, q) + \gamma\, R_{\text{diversity}}(C, G) + \delta\, R_{\text{coherence}}(C, G),$$

where R_{chunks} is the set-level F1 against the gold fragments (primary, $\alpha = 0.5$), R_{coverage} the covered share of query-relevant graph nodes ($\beta = 0.2$), $R_{\text{diversity}}$ the spread of hop-distances of selected nodes ($\gamma = 0.2$) and $R_{\text{coherence}}$ the largest-connected-component share ($\delta = 0.1$). The latter three need no gold annotation and are computable from G alone, so the reward is dense even where labels are scarce. A GNN-based controller encodes node and subgraph state; an LLM policy (a fine-tuned Gemma model) emits the abstract actions above, which an executor compiles into index and graph queries. The policy is optimized with GRPO [16]: for a query we sample a group of trajectories, score each by R, and form a group-relative advantage that needs no learned value baseline,

$$\hat{A}^{(i)} = \frac{R^{(i)} - \text{mean}_j R^{(j)}}{\text{std}_j R^{(j)} + \varepsilon},$$

maximized with a KL penalty to a reference policy. Because R is defined purely on the selected fragments and the generator is never invoked during training, optimization is stable and independent of the downstream LLM.

Experimental Setup

Datasets. We evaluate on four multi-hop question-answering datasets — HotpotQA [19], MuSiQue [20] and 2WikiMultiHopQA [21] (in-domain) and PopQA [22] (out-of-domain) — and on a target corpus of 1,197 Uzbek legislative documents with 700 annotated query–answer pairs split into train/validation/test. We report token-level F1, an LLM-as-judge accuracy ACC_L, and SBERT [23] similarity, averaged over the test questions of each dataset.

Three-level legislative graph. To ground the model on real data we build the graph from the national legislation database of the Republic of Uzbekistan: a corpus of normative acts (the Constitution, the codes and laws) with the inter-article and inter-document references that legislative texts carry. Starting from the Constitution and the principal codes and following references, we parse 140 acts; each act contributes its outgoing references as typed edges, where the relation type (*refers_to*, *amends*, *repeals*, article reference) is inferred from the wording around the reference. On top of this document level we mine the other two levels of the schema directly from the act texts: *entities* — state bodies recognized by a gazetteer (parliament, the Cabinet of Ministers, the President, the chambers of the Oliy Majlis, the courts, the Accounts Chamber) — linked by *regulates* edges; and *concepts* — legal terms taken from the "basic concepts" definition articles (e.g., *information*, *information resources*, *subsoil*, *consumer*) — linked by *defines* edges. Keeping only nodes whose type is known yields a directed graph of 207 nodes and 1,718 typed edges, whose largest weakly connected component covers a fraction 1.00 of the nodes, confirming that legislative acts form one densely cross-referenced body. Table 2 summarizes the graph; Figure 2 shows the three-level neighbourhood of one law and Figure 3 the whole graph.

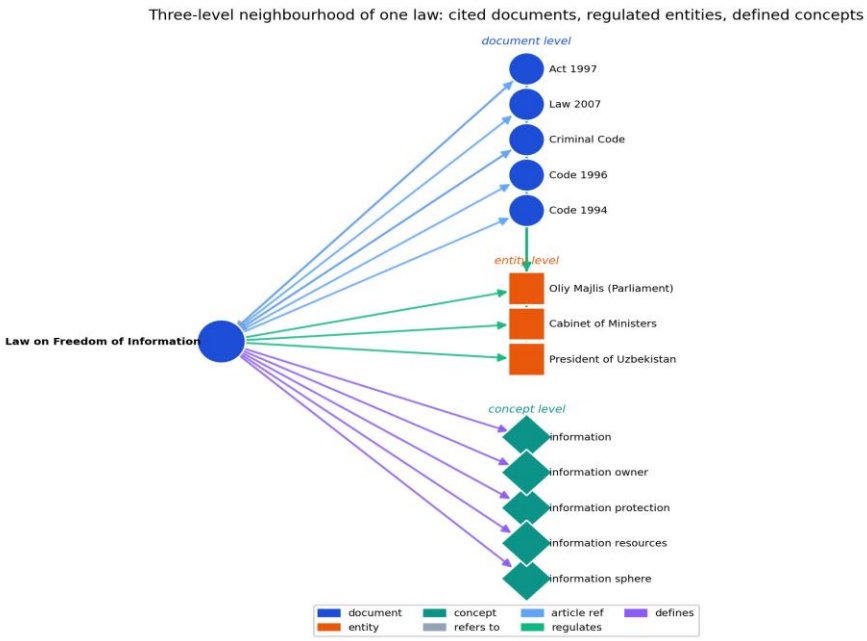

Fig. 2 Simplified three-level view of the model, built from real legislative data: the neighbourhood of one law across the document level (cited

acts), the entity level (regulated state bodies, via regulates edges) and the concept level (defined legal terms, via defines edges).

Table 2 Composition of the empirical three-level legislative graph.

Nodes by level	**count**
Documents (acts)	140
Entities (state bodies)	51
Concepts (defined terms)	16
total	207
Edges by relation	**count**
refers_to / article reference	994
amends / *repeals*	58
regulates (doc → entity)	648
defines (doc → concept)	18
total	1,718
Largest connected component	1.00

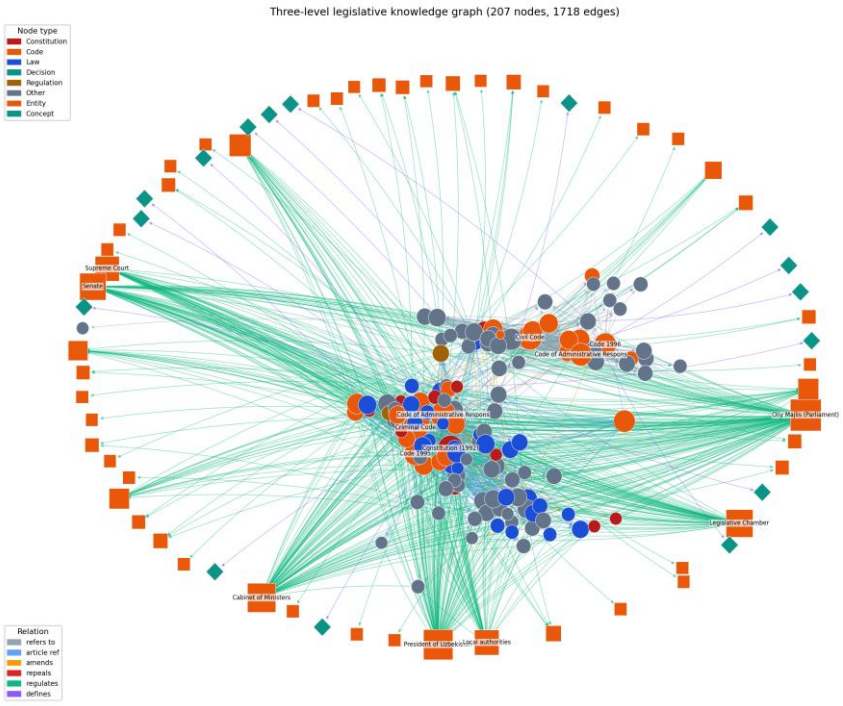

Fig. 3 The full three-level legislative knowledge graph (207 nodes, 1,718 edges). Documents are coloured by act type; entity nodes (squares) and concept nodes (diamonds) sit on the periphery, linked to documents by regulates and defines edges. Node size grows with citation count.

Implementation. The system is realized as a layered information system. A data layer keeps the three indices over shared node identities

(an inverted index, a dense-vector store and a property graph). An intelligent-processing layer hosts the executor and the controller: the executor compiles the policy's abstract actions into concrete index and graph queries and returns observations, while the controller maintains the state, encodes the current subgraph with a GNN, and emits the next action. An interaction layer returns each answer together with the selected fragments and their source acts, so every statement is traceable to a specific norm.

Results and Discussion

Table 3 compares GraphRAG-RL with a parametric-only baseline, naive RAG, two graph-RAG methods and a reinforcement-learning retriever.

GraphRAG-RL attains the best F1 and judge accuracy on the in-domain multi-hop sets and the strongest out-of-domain F1 on PopQA, indicating that the learned policy generalizes rather than overfitting the training domains. Against the parametric-only and naive-RAG baselines the gains are large; against the graph and RL baselines the advantage is consistent and is largest on the hardest multi-hop sets, where integrating heterogeneous fragments matters most. Because the policy learns *when* to stop (the Stop action), the number of retriever calls and the total token budget stay moderate: the policy does not over-search, but uses a minimally sufficient context.

Table 3 Multi-hop QA results, reported as F1 / ACC_L. Best per column in bold.

Method	HotpotQA	MuSiQue	2Wiki	PopQA (OOD)
Vanilla LLM	0.421/0.685	0.358/0.612	0.428/0.691	0.476/0.721
Naive RAG	0.632/0.837	0.468/0.727	0.509/0.742	0.615/0.869
LightRAG	0.458/0.721	0.389/0.648	0.454/0.708	0.524/0.779
HippoRAG	0.653/0.861	0.467/0.769	0.501/0.782	0.618/0.867
R1-Searcher	0.746/0.788	0.602/0.791	0.737/0.865	**0.780**/0.878

GraphRAG-RL	**0.776/0.932**	**0.765/0.918**	0.721/**0.926**	**0.832/0.935**

Legislative corpus. On the target corpus of 1,197 Uzbek legislative documents the system reaches a domain accuracy of 91.3%, with every answer traceable to specific norms — a requirement for deployment in legislative and parliamentary-oversight settings.

Qualitative navigation example. Figure 2 illustrates a typical trajectory. For a query about freedom of information, a vector action first retrieves the governing law; a graph action then follows *defines* edges to the concepts it introduces (*information*, *information owner*, *information resources*) and *regulates* edges to the bodies it binds (the Oliy Majlis, the Cabinet of Ministers, the President), and a further graph action follows *refers_to* edges to the cited acts. A purely lexical or vector retriever would surface passages textually similar to the query but miss the defined concepts and regulated bodies, which share little surface form with the question yet are essential for a complete, traceable answer. The structural reward terms reward exactly this kind of connected, multi-aspect selection.

Limitations. The empirical graph realizes the document, entity and concept levels; the interaction level is specified but populated only at training time. Edge and entity/concept typing rely on gazetteers and lexical rules — high precision but recall-limited — which should be replaced by a learned classifier and extended to Uzbek-language pages. Finally, amendment and repeal edges carry temporal semantics that the current policy treats structurally rather than as a time-aware reasoning constraint.

Conclusion

We formalized relevant-context selection in RAG as a constrained information-maximization problem and reduced it to a Markov decision process, via a computable surrogate utility whose diversity and graph-coherence terms give it a diminishing-returns structure that justifies both greedy and learned policies. We paired this with a hybrid text–graph knowledge model that integrates lexical, vector and graph levels over a heterogeneous text-attributed graph, and instantiated its document, entity and concept levels on a real 207-node graph mined from a national legislative corpus. The resulting method improves answer quality over strong baselines at moderate retrieval cost and produces source-

traceable answers, making it a practical basis for an intelligent information system over large legal corpora. Future work targets joint optimization over the interaction level and richer amendment-aware temporal reasoning.

References

1. P. Lewis *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in neural information processing systems (NeurIPS)*, 2020, pp. 9459–9474.
2. “National database of legislation of the republic of uzbekistan (lex.uz).” https://lex.uz.
3. V. Karpukhin *et al.*, “Dense passage retrieval for open-domain question answering,” in *Proc. EMNLP*, 2020, pp. 6769–6781.
4. S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
5. S. Yao *et al.*, “ReAct: Synergizing reasoning and acting in language models,” in *International conference on learning representations (ICLR)*, 2023.
6. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” in *International conference on learning representations (ICLR)*, 2024.
7. D. Edge *et al.*, “From local to global: A graph RAG approach to query-focused summarization,” *arXiv preprint arXiv:2404.16130*, 2024.
8. Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, “LightRAG: Simple and fast retrieval-augmented generation,” *arXiv preprint arXiv:2410.05779*, 2024.
9. B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, “HippoRAG: Neurobiologically inspired long-term memory for large language models,” *arXiv preprint arXiv:2405.14831*, 2024.
10. B. Peng *et al.*, “Graph retrieval-augmented generation: A survey,” *arXiv preprint arXiv:2408.08921*, 2024.
11. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International conference on learning representations (ICLR)*, 2017.
12. W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in neural information processing systems (NeurIPS)*, 2017.
13. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *International conference on learning representations (ICLR)*, 2018.

14. M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in neural information processing systems (NeurIPS)*, 2016.
15. H. Song *et al.*, “R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning,” *arXiv preprint arXiv:2503.05592*, 2025.
16. Z. Shao *et al.*, “DeepSeekMath: Pushing the limits of mathematical reasoning in open language models,” *arXiv preprint arXiv:2402.03300*, 2024.
17. R. S. Sutton and A. G. Barto, “Reinforcement learning: An introduction,” *MIT Press, 2nd ed.*, 2018.
18. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—I,” *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
19. Z. Yang *et al.*, “HotpotQA: A dataset for diverse, explainable multi-hop question answering,” in *Proc. EMNLP*, 2018, pp. 2369–2380.
20. H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, “MuSiQue: Multihop questions via single-hop question composition,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 539–554, 2022.
21. X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, “Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps,” in *Proc. COLING*, 2020, pp. 6609–6625.
22. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, “When not to trust language models: Investigating the effectiveness of parametric and non-parametric memories,” in *Proc. ACL*, 2023, pp. 9802–9822.
23. N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” *Proc. EMNLP*, 2019.