A Citation-Grounded PDPA Ontology and QA Benchmark for Thai Legal Question Answering, with an Exploratory Graph-First RAG Study

Kyi Thin Nu[1,*], Chutiporn Anutariya[1] and Pattama Krataithong[2]

[1]Faculty of Advanced Science and Technology, Asian Institute of Technology, Khlong Luang, Pathum Thani 12120, Thailand
[2]National Electronics and Computer Technology Center (NECTEC), Khlong Luang, Pathum Thani 12120, Thailand

Abstract

Legal question answering demands answers grounded in the statute and traceable to the provision that supports them, a requirement general-purpose language models meet unreliably. We present three resources for Thailand's Personal Data Protection Act (PDPA): a hand-authored, citation-first OWL ontology of Sections 1-29 (85 classes, 35 object properties, 80 named individuals, 1,194 triples) in which every concept is annotated with its governing Section; a Neo4j knowledge graph materialised from that ontology and the statutory text, in which citation edges make every concept-to-provision link traversable; and a 35-question gold benchmark with reference answers and gold section citations. On these resources we build a graph-first hybrid RAG system (KG-RAG) that translates questions to Cypher, falls back to dense retrieval when the graph returns no usable rows, and abstains when neither path yields grounded context, and compare it with a section-aware vector baseline (V-RAG) sharing its embedding model and generator. KG-RAG is more accurate (0.417 vs. 0.329) and cites provisions far more precisely (0.661 vs. 0.333) at roughly twice the median latency, while the two recall the same share of governing sections (0.510). A retrieval-path analysis added in this version qualifies that headline: the graph path answered 29 of 35 questions, but the six questions routed to the dense fallback contribute 41% of KG-RAG's total accuracy gain, and KG-RAG cited no correct section on any of them. The advantage is therefore partly attributable to the hybrid's dense component rather than to graph structure alone. Our primary contribution is the resources and an explicit account of what this class of comparison can and cannot establish. All artefacts are released.

Keywords

knowledge graph engineering, ontology, retrieval-augmented generation, legal question answering, PDPA, evaluation methodology

1. Introduction

Thailand's PDPA [1] imposes concrete duties on data controllers and processors, but its provisions are dense, cross-referential and unfamiliar to non-specialists. Automated question answering is an attractive compliance aid, provided its answers can be trusted. In the legal setting trust has a precise meaning: an answer must be grounded in the statute and traceable to the exact provision supporting it. A fluent but unattributable answer is useless to a practitioner who must justify a decision, and an incorrect one carries regulatory risk.

Retrieval-augmented generation [2] conditions generation on retrieved evidence. Its dominant form, vector RAG, retrieves nearest text chunks; this is fast and effective for definitional questions, but it retrieves opaque prose spans, so the section number anchoring an answer surfaces only when it appears verbatim, and dense similarity does not represent the relational structure of law. An ontology-driven knowledge graph offers a complementary substrate in which the provision grounding an answer is an explicit, traversable object. GDPRtEXT [3] and the W3C Data Privacy Vocabulary [4] have shown the value of modelling data-protection regulation as linked data; Thailand's PDPA has no such resource.

This paper is primarily a resource contribution with an exploratory evaluation attached. We release a citation-first PDPA ontology, a retrieval-shaped Neo4j knowledge graph, and a 35-question gold

IJCKG 2026 Posters, Demos and Workshops, November 19–21, 2026, Bangkok, Thailand
*Corresponding author.
kyithinnu@ait.asia (K. T. Nu); chutiporn@ait.asia (C. Anutariya); pattama.krataithong@nectec.or.th (P. Krataithong)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-06

CEUR-WS.org/Vol-4316/kge-short-1.pdf

benchmark, and we report an exploratory comparison of a graph-first hybrid RAG system against a section-aware vector baseline.

We are explicit about what the comparison does not establish. It does not isolate the effect of graph retrieval: the two systems differ in retrieval substrate, in query formulation and in fallback behaviour. Section 5.1 quantifies this directly, showing that a disproportionate share of the measured advantage originates on the dense fallback path rather than in the graph. We do not independently assess the correctness of the generated Cypher queries, and therefore do not claim deployment readiness. Given the observed answer accuracies of 0.33–0.42, the results suggest that neither system should be relied upon for compliance-related guidance without appropriate expert review.

2. Related Work

Legal NLP and Thai statutory QA. Retrieval is central to modern legal analytics [5], and neural methods support tasks such as judgment prediction [6]. NitiBench [7] provides two Thai legal QA datasets and finds that section-based chunking substantially improves retrieval, which independently corroborates our baseline design (Section 4). Our contribution is not the first Thai legal QA resource; it is the first ontology-grounded, provenance-traversable resource for the Thai PDPA specifically, a statute NitiBench does not cover.

Ontologies for data-protection law. Data-protection modelling is overwhelmingly GDPR-centred: GDPRtEXT renders the regulation as linked data [3], DPV offers a standardised vocabulary for personal-data handling [4], and conceptual models map GDPR provisions onto ontology concepts [8]. We follow GDPRtEXT's organisation rather than DPV's processing-centric vocabulary because our task is provision retrieval, not processing-activity description.

Graph-based RAG. A growing family of systems retrieves over graphs rather than flat text. It is worth distinguishing them carefully, since they are often grouped together. Microsoft's GraphRAG [10] does not retrieve from a curated knowledge graph at all: it extracts an entity graph from a document corpus, detects communities in that graph, and uses hierarchical community summaries to answer corpus-level, query-focused summarisation questions. LightRAG [11] likewise builds its index from text, and HippoRAG [12] runs Personalised PageRank over an automatically constructed open KG for multi-hop retrieval. All three derive their graph from text automatically. Our setting is different in both the object retrieved and the question asked: we query a hand-authored, schema-constrained graph of a single statute with a structured query language, to return a specific provision rather than a synthesised summary. We therefore position this work as adjacent to, rather than an instance of, that line of research.

Natural language to Cypher. Because our system reaches the graph through an LLM-generated Cypher query, the closest technical literature is semantic parsing for graph query languages. Text2Cypher [13] releases a large instruction dataset and fine-tuned models for the task, and Cypher-Bench [14] benchmarks retrieval over full-scale property graphs, reporting that even strong LLMs fall well short of reliable query generation. Synthetic-data methods aim to close that gap [15]. This literature matters for interpreting our results: our graph path inherits the error profile of text-to-Cypher translation, and we report its end-to-end success rate in Section 5.1 while noting in Section 6 that we do not evaluate query correctness independently.

Evaluating RAG. Reference-free frameworks such as RAGAS [16] and ARES [17] use LLM judges to score answer and context quality, and LegalBench-RAG evaluates retrieval for legal questions [18]. These frameworks assume a single retrieval representation, so how their context-side metrics behave across structurally different retrievers has received little attention; Section 5 takes this up.

3. Study Design

3.1. Citation-first ontology

The ontology was authored manually in Protégé [9] in OWL 2 DL, scoped by competency questions to the Introduction and Chapters I-II (Sections 1-29), the parts a user-interest analysis identified as covering the most frequent practical concerns. It contains 1,194 triples, 85 classes, 35 object properties and 80 named individuals, and is reasoner-checked for consistency.

Its organising decision is a citation-first principle: every substantive concept carries a `:source` annotation naming the Section, Part or Chapter that governs it, so any answer derived from the ontology can be traced to a provision. Three further design choices follow from the legal setting. First, statutory structure and substantive law are modelled in separate branches (Figure 1), so that renumbering the Act would not disturb the concept hierarchy. Second, the enumerable devices of the Act are modelled as closed individual sets rather than free-text values: 6 lawful bases, 6 exemptions and 13 consent exceptions, which lets a query ask which exceptions apply without parsing prose. Third, sections referenced from within scope but lying outside it are retained as flagged, text-free stubs, so that a cross-reference leaving the modelled range is visible to a query rather than silently absent. The 35 object properties combine structural relations (`has_Chapter`, `has_Part`, `has_Section`, with inverses) with substantive ones such as `hasLegalBasis`, `requiresConsent`, `hasExemption` and `transferToForeignCountry`.

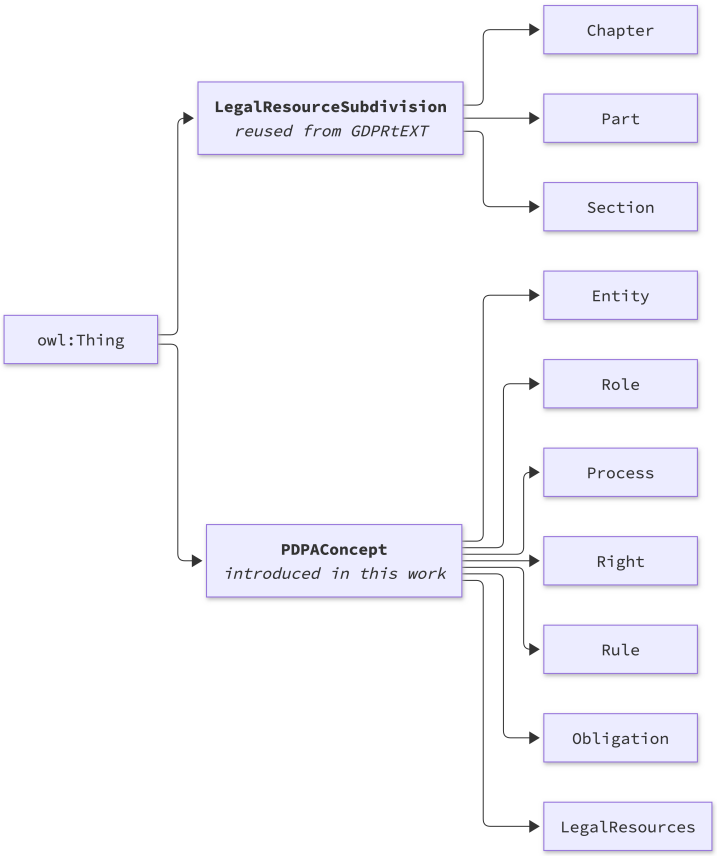


Figure 1: Top-level classes of the PDPA ontology.

3.2. Retrieval-shaped knowledge graph

An initial import of the OWL file through the NeoSemantics plugin mirrored the raw RDF structure and produced a graph awkward to traverse at query time (82 nodes, 147 relationships). The released graph is instead built by a script that parses the ontology with `rdflib`, extracts the text of each in-scope section, and writes nodes, relationships and vector indexes into Neo4j with idempotent `MERGE` statements. Statutory text is stored on the nodes, so a query that reaches a `Section` returns the wording that grounds the answer.

The graph holds 150 nodes and 265 relationships under 8 node labels and 9 relationship types. One `LegalAct` node connects through 3 `Chapter` and 3 `Part` nodes to 37 `Section` nodes, 29 of them in scope; 81 `Concept` nodes carry the substantive law alongside the enumerated `LawfulBasis`, `Exemption` and `ConsentException` individuals, and 103 `DEFINED_IN` and `CITED_IN` edges link concepts to their grounding sections.

3.3. Benchmark and annotation protocol

We collected 54 frequently asked questions from three authoritative Thai PDPA sources and categorised each by theme and by governing chapter and section; this analysis fixed the study scope. From that pool we constructed 35 (question, ground-truth) pairs, each with a reference answer, gold section citations, a thematic category and a question type. Of the 35, 29 fall fully within scope, 5 extend beyond Chapters I-II and 1 falls outside the PDPA entirely; the last six probe behaviour at and beyond the scope boundary. Citation scoring covers the 34 questions carrying gold sections.

The protocol has a limitation we state plainly. All reference answers and gold citations were authored by the first author, working from the official English translation rather than from the ontology. No qualified legal practitioner reviewed the benchmark, and because annotation was performed by a single annotator, inter-annotator agreement could not be measured. The 19 excluded questions were near-duplicates, questions with no mappable provision, or items whose phrasing admitted more than one defensible answer. The benchmark is also not independent of the ontology: both derive from the same 54-question pool, so results characterise in-scope behaviour on user-salient questions rather than generalisation to arbitrary PDPA questions.

4. RAG Systems

Both systems share an embedding model (`nomic-embed-text`, 768-d, served through Ollama) and a generator (GPT-4o-mini, temperature 0), so the comparison varies retrieval while holding generation constant. GPT-4o-mini was selected in a prior generator comparison that used this same 35-question benchmark, so the generator is not independent of the evaluation set; because both systems use it, this inflates both absolute scores rather than favouring either.

KG-RAG is graph-first and hybrid (Figure 2). GPT-4o-mini translates the question into a Cypher query, which is executed against the graph. A result is treated as usable when the query executed without error, the result set is non-empty after removing embedding vectors and duplicate rows, and at least one row carries statutory text or a resolvable in-scope section. Otherwise the system falls back to dense retrieval over the concept nodes, using the same embedding model as the baseline; if neither path yields grounded context it returns a fixed refusal. Retrieved rows are serialised to text with section tags preserved before generation.

V-RAG is a retrieve-then-generate baseline. The PDPA text is split by section-aware chunking at each "Section N" heading, so each of the 49 chunks corresponds to one section; chunks are embedded and stored in a FAISS index, and the four nearest ($k = 4$) are retrieved, each prefixed with its section label so the citation instruction is actionable. This is not a naive baseline: section-aware chunking is precisely the design NitiBench found to improve Thai legal retrieval [7], and both systems receive

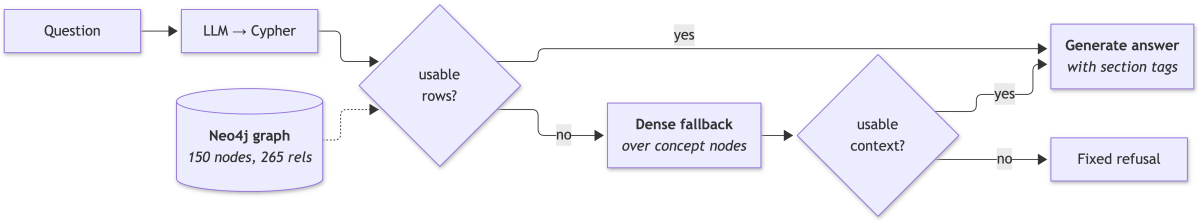


Figure 2: The KG-RAG graph-first hybrid pipeline.

Table 1
Main results. Context-side metrics are within-system diagnostics (Section 5); cross-system rows carry paired Wilcoxon tests.

Metric	Comparability	V-RAG	KG-RAG	Test
Context Precision	within-system	0.810 ± 0.016	0.769 ± 0.018	—
Context Recall	within-system	0.342 ± 0.001	0.564 ± 0.040	—
Faithfulness	within-system	0.740 ± 0.009	0.764 ± 0.014	—
Noise Sensitivity	within-system	0.268 ± 0.003	0.323 ± 0.018	—
Answer Accuracy	cross-system	0.329	**0.417**	$W = 26.0, p = 0.028$
Response Relevancy	cross-system	0.751	**0.821**	not tested
Citation recall	cross-system	0.510	0.510	$W = 38.5, p = 0.97$
Citation precision	cross-system	0.333	**0.661**	$W = 14.0, p = 0.005$
Median latency (s)	cross-system	**2.12**	4.35	—

the same citation instruction. What it lacks, and what we did not run, is a closed-book condition establishing how much retrieval contributes at all.

5. Evaluation and Results

RAGAS [16] context metrics score the passages a system retrieves. They are informative within a retriever but are not comparable across architectures returning structurally different objects: V-RAG retrieves prose chunks, KG-RAG serialised graph records, and an LLM judge scoring both is not grading like against like. We therefore treat Context Precision, Context Recall, Faithfulness and Noise Sensitivity as within-system diagnostics only, and base cross-system comparison on answer-level measures: Answer Accuracy and Response Relevancy, both LLM-judged, and citation precision and recall computed by extracting PDPA section numbers from the generated answer and comparing them with the gold set after normalisation. The citation measure is architecture-neutral but end-to-end: it does not separate a retrieval miss from a citation the generator failed to emit.

Each judge is run three times at temperature 0 and we report mean $\pm$ SD. Cross-system comparators are tested with two-sided paired Wilcoxon signed-rank tests. Citation precision is undefined when a system cites nothing, which happened for KG-RAG on 6 of the 34 scored questions and for V-RAG on none, so that test uses the 28 questions where both are defined. Three caveats bound every judged number: GPT-4o-mini both generates and judges, so self-preference cannot be excluded; low run-to-run variance establishes stability, not agreement with expert legal judgment; and the absolute values should therefore be read as a relative comparator, not as a rate of legal correctness.

Table 1 collects the headline figures. KG-RAG is more accurate and far more precisely cited; the two systems tie on citation recall, and the paired test confirms the tie is genuine rather than an averaging artefact. V-RAG is roughly twice as fast at the median.

Table 2
Performance by KG-RAG retrieval path. Citation precision is undefined on the fallback path because KG-RAG cited no section on five of those six questions.

		Answer Accuracy		Citation prec.		Citation recall	
Path	n	V-RAG	KG-RAG	V-RAG	KG-RAG	V-RAG	KG-RAG
Graph-routed	29	0.353	0.417	0.384	0.661	0.563	0.598
Fallback-routed	6	0.208	0.417	0.100	n/a	0.200	0.000

Table 3
Citation precision and recall by question type; latency is the median in seconds. Types sum to 35; citation scoring covers 34.

		Citation prec.		Citation recall		
Question type	n	V-RAG	KG-RAG	V-RAG	KG-RAG	Lat. ratio
Definition	7	**0.786**	0.667	**0.857**	0.500	1.9×
Yes-no	11	0.235	**0.722**	**0.636**	0.545	2.3×
Explanatory	5	0.250	**0.667**	0.375	**0.500**	2.2×
List	5	0.367	**0.800**	0.367	**0.567**	1.9×
Scenario	4	0.125	**0.500**	0.250	**0.500**	2.1×
Procedural	3	0.000	**0.250**	0.000	**0.333**	1.7×

5.1. Retrieval path: where the advantage actually comes from

Because KG-RAG is hybrid, the aggregate figures conflate two different retrieval mechanisms. The graph path produced usable rows on 29 of 35 questions (82.9%; 85.3% of the 34 in-scope questions), and 6 fell through to the dense fallback: five in-scope questions whose Cypher query returned nothing usable, and the single out-of-PDPA question, where the fallback also returned nothing and the system correctly refused. Table 2 separates the two groups.

Two things follow, and they qualify the headline result. First, KG-RAG's accuracy margin over V-RAG is +0.063 on graph-routed questions but +0.208 on fallback-routed ones; summed across the benchmark, the six fallback questions contribute 41% of the total accuracy gain while being 17% of the questions. A material part of what the aggregate attributes to the graph is produced by dense retrieval over concept nodes. Second, the citation-precision advantage is effectively a graph-path property: KG-RAG cited no section at all on five of the six fallback questions and achieved a citation recall of 0.000 on that path, against 0.200 for V-RAG. The hybrid therefore improves answers on the questions its graph cannot reach, but stops grounding them, which for a citation-oriented system is the more consequential failure.

5.2. By question type, and where KG-RAG loses

V-RAG is best on definition questions, where similarity lookup suffices and the answer usually sits in one section. KG-RAG is more precise on every other type and more complete on the relational ones, most visibly on list and yes-no questions. Both struggle with procedural questions, the smallest and hardest type. With 3-11 questions per cell these are descriptive tendencies, not estimates with usable confidence intervals.

The cases where KG-RAG loses are as informative as the aggregate. On PDPA-027 the graph path returned a grounded row and the system still scored 0.000 against V-RAG's 0.750: the Cypher query retrieved a section, but not the one the reference answer required, which shows that a grounded graph result is not by itself a correct one. On PDPA-001 KG-RAG answered from the graph but cited a section outside the gold set, dropping citation precision to 0.000 where V-RAG scored 0.33. Conversely, V-RAG's characteristic failure is over-citation: it always emits section numbers, correct or not, which is what depresses its precision to 0.333 overall while its recall matches KG-RAG's. The two systems fail in

opposite directions, and for a legal assistant KG-RAG's silence is the safer of the two failure modes only when it is accompanied by an explicit refusal.

6. Discussion and Limitations

The two systems have complementary profiles rather than a single winner. KG-RAG produced more accurate and far more precisely cited answers; V-RAG answered about twice as fast and never went silent. The precision advantage follows from a construction-time decision rather than a retrieval-time one: the citation edges that tie each concept to its governing section were authored into the ontology, and no retrieval strategy can recover a link that was never modelled. That is the part of this work we would expect to transfer to other statutes.

Section 5.1 bounds how much of the rest transfers. Because a disproportionate share of the accuracy gain arises on the dense fallback path, this study does not establish that graph retrieval caused the improvement, and the honest summary is that a graph-first hybrid outperformed a dense baseline under conditions where both used the same generator. Separating the contributions requires ablations we did not run: a graph-only condition with the fallback disabled, and a closed-book condition with no retrieval at all.

Cypher generation. We report the end-to-end rate at which the graph path yielded usable context (82.9%) but we do not evaluate the generated queries themselves. A query that is syntactically valid, executes, and returns a grounded but irrelevant row is counted as a success by our criterion, and PDPA-027 shows such cases exist. Independent evaluation of query correctness against gold Cypher, following the protocols of [13, 14], is the most valuable single addition to this system's evaluation and we did not perform it.

Scalability of manual modelling. The ontology covers 29 of the Act's sections. Extending it is not uniformly costly: the schema, which is where the design effort concentrated, is largely statute-general, and Chapters III-VII introduce new substantive classes (data-subject rights machinery, the regulator's powers, penalties) but reuse the structural branch and the citation-first pattern unchanged. The marginal cost is therefore dominated by instance population and by the manual step of locating each concept's governing provision. We do not report measured effort, having not instrumented the authoring process; this is a gap for anyone planning a comparable resource. An obvious direction is a hybrid workflow in which an LLM proposes candidate concept-to-section links from the statutory text for human verification, retaining the provenance guarantee while reducing the manual burden.

Other threats. The benchmark is single-annotated, not legal-practitioner reviewed, and not independent of the ontology's scope. The generator was selected using this same benchmark. Judge and generator are the same model. With 35 questions and 19 tied pairs on the primary comparator, the study is underpowered for anything beyond a directional claim, and no correction for multiple comparisons was applied across the three tests. One statute, two chapters, one embedding model, one generator, and evaluation in English translation rather than Thai.

7. Conclusion

We presented three resources for Thai PDPA question answering, a citation-first OWL ontology of Sections 1-29, a retrieval-shaped Neo4j knowledge graph and a 35-question gold benchmark, together with a graph-first hybrid RAG system built on them. The system is more accurate and more precisely cited than a section-aware vector baseline, but a retrieval-path analysis shows that a disproportionate share of the accuracy gain originates on its dense fallback path, and that its citation advantage is confined to questions the graph could answer. We offer the resources with more confidence than the comparison. Priorities for future work follow directly from the limitations above: independent evaluation of Cypher generation against gold queries; a held-out benchmark authored after the ontology was frozen, with practitioner review and measured agreement; ablations isolating the graph path; and a

comparison against Graph RAG systems whose graphs are built automatically by an LLM, which would test directly whether hand-authored provenance earns its construction cost.

Data Availability

The ontology, knowledge-graph build script, benchmark, both pipelines, prompts and evaluation outputs are available at https://github.com/Kyi1914/KyiMEThesis2026, together with a README mapping each artefact and reported table to its exact path, the software and model versions required, and the order of steps needed to reproduce each result. All figures reported here derive from a single designated result file, identified in that README.

Declaration on Generative AI

The authors used generative AI tools for drafting and editing support and for language refinement. All content was subsequently reviewed and verified by the authors, who take full responsibility for the publication's content.

References

[1] Government of Thailand, "Personal Data Protection Act B.E. 2562 (2019)," Government Gazette, vol. 136, ch. 69 Gor, 27 May 2019 (unofficial English translation).

[2] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, 2020.

[3] H. J. Pandit, K. Fatema, D. O'Sullivan, and D. Lewis, "GDPRtEXT: GDPR as a linked data resource," in *The Semantic Web: ESWC 2018 Satellite Events*. Springer, 2018, pp. 481–495.

[4] H. J. Pandit et al., "Data Privacy Vocabulary (DPV): Version 2.0," in *Proc. International Semantic Web Conference (ISWC)*. Springer, 2024.

[5] K. D. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, 2017.

[6] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural legal judgment prediction in English," in *Proc. 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 4317–4323.

[7] P. Akarajaradwong, P. Pothavorn, C. Chaksangchaichot, P. Tasawong, T. Nopparatbundit, K. Pratai, and S. Nutanong, "NitiBench: Benchmarking LLM frameworks on Thai legal question answering capabilities," in *Proc. EMNLP 2025*, pp. 34362–34389.

[8] J. Tom, E. Sing, and R. Matulevičius, "Conceptual representation of the GDPR: Model and application directions," in *Perspectives in Business Informatics Research (BIR 2018)*. Springer, 2018, pp. 18–28.

[9] M. A. Musen, "The Protégé project: A look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015.

[10] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A Graph RAG approach to query-focused summarization," arXiv:2404.16130, 2024.

[11] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "LightRAG: Simple and fast retrieval-augmented generation," arXiv:2410.05779, 2024.

[12] B. J. Gutiérrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, "HippoRAG: Neurobiologically inspired long-term memory for large language models," in *Advances in Neural Information Processing Systems*, 2024.

[13] M. O. Ozsoy, L. Messallem, J. Besga, and G. Minneci, "Text2Cypher: Bridging natural language and graph databases," in *Proc. Workshop on Generative AI and Knowledge Graphs (GenAIK) at COLING 2025*, pp. 100–108.

[14] Y. Feng, F. Papailiopoulou, and E. Kandogan, "CypherBench: Towards precise retrieval over full-scale modern knowledge graphs in the LLM era," arXiv:2412.18702, 2024.

[15] A. Tiwari et al., "SynthCypher: A fully synthetic data generation framework for text-to-Cypher querying in knowledge graphs," arXiv:2412.12612, 2024.

[16] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. of the European Chapter of the ACL (EACL): System Demonstrations*, 2024, pp. 150–158. arXiv:2309.15217.

[17] J. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, "ARES: An automated evaluation framework for retrieval-augmented generation systems," in *Proc. NAACL*, 2024.

[18] N. Pipitone and G. Houir Alami, "LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain," arXiv:2408.10343, 2024.