QuALA-NL: Question & Answer with Legal Attribution in Dutch

Romy A.N. van Drie[1], Roos M. Bakker[1,2], Daan L. Di Scala[1,3], Maaike H.T. de Boer[1]
[1] Data Science, TNO, The Hague, The Netherlands
[2] Leiden University Centre for Linguistics, Leiden University, Leiden, The Netherlands
[3] Utrecht University, Utrecht, The Netherlands
{romy.vandrie, roos.bakker, daan.discala, maaike.deboer}@tno.nl

Abstract

Ensuring trustworthy and traceable outputs from Large Language Models (LLMs) is crucial in high-stakes domains such as law. Retrieval-Augmented Generation (RAG) offers a way to enhance LLMs with domain-specific or updated information and provide attribution to the source, and recent work has focused on knowledge-based RAG (K-RAG) for improved factual grounding. However, proper evaluation of such systems requires high-quality datasets. To address this need, we introduce QuALA-NL: a dataset that provides attributions to legal formalizations, enabling experiments with K-RAG in the legal domain. The dataset contains 101 QA pairs on three Dutch laws, with attributions to the law text and a formalization of the interpretation of the legal text. To demonstrate the capabilities of the dataset, we perform experiments using four configurations: LLM-only, RAG using legal texts, K-RAG using a formalization of the legal texts, and RAG combining both legal texts and the formalizations. The results show that K-RAG has the highest retrieval scores, but that this method is outperformed by text-based RAG on generation. A qualitative analysis shows that the use of the knowledge graph for the generation of answers can be improved. QuALA-NL can be used in future work to experiment with knowledge-based Retrieval Augmented Generation methods.

Keywords: Dataset, Attributed Question Answering, Retrieval-Augmented Generation, Knowledge Graphs, Knowledge-Based RAG, Graph RAG

1. Introduction

Large Language Models (LLMs) are currently used in many domains, including the legal domain. In such a high-stakes domain, it is important that the response of the LLM is trustworthy and traceable. In order to verify the performance of LLM-based systems, it is necessary to have proper datasets to evaluate on. As each country operates under its own legal framework (Wiggers, 2023) and in their local language(s), datasets should be jurisdiction- and language-specific. In this paper, we create a dataset with legal question-answer pairs in Dutch, using Dutch laws.

This paper builds upon previous research that introduced a Dutch legal QA dataset containing 102 question-answer pairs, each linked to relevant Dutch law articles (Redelaar et al., 2024). In that dataset, the questions focused specifically on preconditions for legal actions (e.g. "In which conditions can a pronouncement of undesirability of an alien be lifted?"). The dataset covers 25 Dutch laws, with answers attributed to the original statutory texts. Following this work, we introduce a dataset that similarly includes 101 question-answer pairs, but focuses on three different laws and expands the scope beyond preconditions. We name this dataset QuALA-NL (Question & Answer with Legal Attribution in Dutch). In addition to supplying the question, answer, and reference to the source text, we also provide attributions to formalizations to facilitate experiments with knowledge-based retrieval-augmented generation (RAG) systems.

We use FLINT (Formal Language for the Interpretation of Normative Theories) for our formalizations, as introduced by van Doesburg (2017). The creation of FLINT interpretations is supported by open source tools such as a source decomposition tool to convert law texts into RDF representations and an editor to create interpretations of norms (van Gessel et al., 2023). Our work is based on laws with FLINT interpretations[1].

To demonstrate the capabilities of the dataset and its potential applications, we implement and evaluate a Retrieval Augmented Generation (RAG) pipeline. This pipeline generates answers to Dutch law questions, and cites a source for this answer. We compare four configurations: LLM-only (no RAG), RAG using only law text, RAG using only FLINT interpretations, and RAG combining both sources. We evaluate performance using the same metrics as introduced by Redelaar et al. (2024). For retrieval, we use hit rate, precision and recall. For generation, we use ROUGE, METEOR and MAUVE, and several G-EVAL metrics (fluency, consistency, coherence and relevance). The QuALA-NL dataset as well as the methods and evaluation are available in our repository[2].

In the next section, we discuss related work on both legal question answering and (knowledge-based) RAG. Section 3 explains the creation of

[1] https://gitlab.com/normativesystems/use-cases-public
[2] https://gitlab.com/normativesystems/question-and-answer/quala-nl

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 674–684
11–16 May 2026.

Figure 1: Example of legal attributed QA based on formalized knowledge of law texts.

the dataset. Section 4 contains the method for our experiments with the dataset. Section 5 provides the results and Section 6 the discussion. We conclude the paper in Section 7.

2. Related Work

2.1. Legal Question Answering

Several legal QA datasets have already been released: JEC-QA, a Chinese dataset for multiple-choice questions, sourced from legal exams (Zhong et al., 2020), English and Chinese datasets sourced from online forums (Mansouri and Campos, 2023; Chen et al., 2023), a French dataset sourced from lawyers (Louis et al., 2023), a German QA dataset (Hoppe et al., 2021), an English benchmark with a multiple-choice QA task on Case Holdings on Legal Decisions (Chalkidis et al., 2021) and a Dutch legal QA dataset for preconditions (Redelaar et al., 2024).

Legal QA systems include one to answer legal questions in French for the Belgian law (Louis et al., 2023), a chatbot for French (Queudot et al., 2020), Thai (Socatiyanurak et al., 2021), and Indonesian (Firdaus et al., 2020). For English, several LLM models are present, such as a Law-LLM model by AdaptLLM (Cheng et al., 2024) and SaulLm (Colombo et al., 2024). For Dutch, there are several commercial AI-based legal QA tools such as Postbus42[3] and LegalMike[4].

Legal QA sets can contain questions that require an answer that is binary, multiple-choice, multi-span or long-form (Martinez-Gil, 2023). Modern QA systems commonly use two stages following a Retrieval Augmented Generation framework (Martinez-Gil, 2023), using a retrieval step to extract the relevant parts for the answer (Hoppe et al., 2021; Khazaeli et al., 2021; Karpukhin et al., 2020), and a generation step to generate fluent answers (Louis et al., 2023). These kind of systems will be explained further in the next subsection.

2.2. (Knowledge-based) Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) uses external knowledge in the prompt to an LLM (Lewis et al., 2020). A RAG system is often used to add additional information, such as custom data or newer information compared to when the LLM was trained. Substantial research has been dedicated to optimizing retrievers for QA tasks (Chen et al., 2017; Karpukhin et al., 2020; Formal et al., 2021; Lin et al., 2023; Ram et al., 2023).

More recently, RAG methods are extended by using Knowledge Graphs (KGs) as database. The main difference between traditional RAG (NaïveRAG) and Knowledge Graph RAG (KRAG or GraphRAG) is that the retriever component does not only use unstructured chunks of texts as input, but instead retrieves from KGs. Both Chen (2025) and Peng et al. (2024) provide extensive overviews of recent GraphRAG investigations. KGs typically consist of triples (subject, relation, object) that provide additional semantic and relational information over unstructured data (Wang et al., 2017). Different retriever approaches for GraphRAG exist: non-parametric retrievers based on heuristics and traditional graph search algorithms (He et al., 2024), Graph Neural Network-based retrievers that encode graph data (Mavromatis and Karypis, 2024), and retrievers based on Language Models (Zhang et al., 2022). To minimize the difference between methods, while properly comparing KRAG to other LLM embedding approaches, we opt for the latter and include a Language Model-based knowledge retriever method in our work.

Zhu et al. (2025) introduces KG^{2}RAG, a RAG method which uses KGs to expand and organize retrieved chunks. KG-RAG (Sanmartin, 2024) constructs a KG from unstructured text and then uses a graph-based Chain of Explorations retrieval. Microsoft GraphRAG (Edge et al., 2024) uses a pipeline that converts documents to a KG through entity extraction and relation linking, and retrieves based on community summaries of closely related entities. KRAGEN (Matsumoto et al., 2024) utilizes graph-of-thoughts prompting to decompose problems, and retrieves relevant knowledge from a vector database. When targeting content from KGs, different scopes and granularities exist, such as considering content on a node, path, triple or subgraph level (Chen, 2025). Following approaches by Zhang et al. (2022) and Hu et al. (2025), we choose to retrieve on a subgraph level, as this provides the most contextual information.

[3] https://www.postbus42.nl
[4] https://legalmike.ai/

2.3. Formalizations of the Law

Many vocabularies and ontologies are developed to structure and organise normative data (Leone, 2021), such as LegalRuleML (Athan et al., 2015), LKIF (Hoekstra et al., 2007), the UFO-L Core Ontology (Griffo et al., 2018) and FLINT (van Doesburg, 2017). While most ontologies focus on obligations, prohibitions or violations, FLINT focuses on actions (van Doesburg, 2017). The central concepts of FLINT are described in an ontology (Breteler et al., 2023), and a logic is being developed (van Gessel and Goossens). Moreover, studies have been conducted on the extraction of structured knowledge from law texts for the creation of FLINT frames (Bakker et al., 2022b,a, 2025) and a dataset has been created to support that work (van Drie et al., 2023).

FLINT represents interpretations of normative texts in terms of normative acts and their pre- and postconditions. This is done using Fact, Act and Duty frames. Fact frames describe matters that characterize the state of the normative system. Act frames describe the actions that agents might take. Act frames consist of an actor (who can perform the action), action, object (that is affected by the action), recipient (who can undergo the action), precondition (facts that need to be true before it is valid to execute the act) and postcondition (creation or termination of one or more facts or duties). Duty frames describe acts that ought to be performed in the future or, in case of a violation, should have been performed in the past. A duty frame contains a duty holder and a claimant. An example of a FLINT Fact, Act and Duty frame can be found in Table 1.

Among the available frameworks, FLINT appears particularly suitable for representing legal information in our QA context. Many legal questions concern actions - who is permitted or obliged to perform them, under what conditions, and with what consequences - and FLINT's action-oriented perspective aligns naturally with this structure. Moreover, since FLINT is one of the formal languages currently used and explored within the Dutch legal-tech landscape, it provides a relevant and contextually appropriate foundation for research.

Act Frame	
Action	revoke
Object	permanent residence permit
Actor	Our Minister of Justice and Security
Recipient	-
Precondition	OR: The Netherlands is not the main residence of the permit holder. The permit holder has provided incorrect or incomplete information. The permit holder has been sentenced to three years or more in prison for a crime. The permit holder poses a threat to national security.
Postcondition	-
Fact Frame	
Fact	The Netherlands is not the main residence of the permit holder.
Duty Frame	
Duty	to obtain, accept, and retain employment, according to ability
Claimant	municipal executive
Duty holder	the interested party

Table 1: Example of an Act and Fact Frame based on the Aliens Act, and a Duty Frame based on the Participation Act

3. Dataset Creation

Our dataset consists of questions and answers, and their attribution. As our goal is to create a dataset to facilitate work on knowledge-based retrieval-augmented generation, we not only supply a reference to the legal text, but we also supply a reference to a formalization of the interpretation of the relevant legal text. To be specific, each question-answer pair has an attribution to one or more chunks of a law text, and one or more act, fact or duty frames.

The criteria for including laws in the dataset are diversity, availability of a FLINT formalization, and frequency of use or inquiry. Accordingly, three Dutch laws are being included: the Participation Act (PA)[5], the Aliens Act (AA)[6], and the National Old Age Pensions Act (NOAPA)[7]. The purpose of the Participation Act is to help people find and keep a job, and to provide a safety net through social assistance for those who are (temporarily) unable to support themselves. The Aliens Act regulates the access, admission, supervision and deportation of foreigners (so-called 'aliens'). The purpose of the National Old Age Pensions Act is to provide a basic income for people in the Netherlands once they reach the state pension age, ensuring financial security in old age. The full dataset is available in our repository[2].

3.1. Creating question-answer pairs

We create questions using the original law text from the official government website and relevant governmental information websites with frequently asked

[5] https://wetten.overheid.nl/BWBR0015703/2024-01-01
[6] https://wetten.overheid.nl/BWBR0011823/2024-01-01
[7] https://wetten.overheid.nl/BWBR0002221/2025-01-01

questions[8,9,10]. Drawing upon our expertise and available sources, we developed a range of questions that may arise at the municipal counter, such as the following: 'I've received an inheritance. Does this affect my social assistance?', 'When is a regular residence permit for an indefinite period revoked?' and 'Can my AOW (state pension) benefits be suspended if I have provided the wrong address?'.

A ground truth answer for each question is developed through several steps. First, passages from legal texts that address the question are selected, and attribution to the source text is recorded. Second, the answer is rewritten to respond directly to the question, with simplified language while maintaining the complexity and potential ambiguities of the original material as much as possible. Finally, relevant FLINT frames to the question are identified and added to the dataset. The answers are collaboratively prepared by a team of three FLINT and legal domain experts, along with two authors who possess expertise in natural language processing and foundational knowledge in law and semantics.

The dataset includes 34 question-answer pairs for two laws, and 33 for the third law, attributing both the law text and FLINT frames, resulting in 101 pairs. Some of the questions overlap with the work of Redelaar et al. (2024). The main difference is that they only focus on questions about preconditions, whereas we present a more diverse set of questions. Another difference with our dataset is that Redelaar et al. (2024) does not attribute answers to FLINT frames but only provides attributions to the law text.

In total, the average length of the question is 11.47 words (± 4.11), and the human answers have 46.53 words (± 33.14). The average number of human attributions for the law texts is 1.90 (± 1.49) and for the FLINT frames this is 1.22 (± 0.64).

3.2. Law Knowledge Corpus

Following Redelaar et al. (2024), we obtain each law's XML from the official government website[5, 6, 7]. The XML is parsed to CSV, and separated into articles (one article per row). Articles over 150 words are split into multiple rows. Each chunk, corresponding to one row, receives a unique document ID to facilitate straightforward referencing.

In total, the law knowledge corpus contains 784 chunks, of which 150 are from the NOAPA, 312 are from the PA and 319 are from the AA.

[8] https://ind.nl/nl/service-en-contact/veelgestelde-vragen

[9] https://www.rijksoverheid.nl/onderwerpen/immigratie-naar-nederland

[10] https://www.rijksoverheid.nl/onderwerpen/algemene-ouderdomswet-aow

Law	#Act	#ComplexFact	#Duty	Total
NOAPA	8	10	0	18
PA	9	15	12	36
AA	55	4	0	59
Total	72	26	12	114

Table 2: Statistics of the FLINT Knowledge Corpus

3.3. FLINT Knowledge Corpus

For the FLINT knowledge corpus, we use the existing interpretations as publicly available[11]. The .ttl file is loaded in and the frames are extracted. The frame ID is used as an identifier. The frames are transformed to text using the type of frame and then the component name and each of the components, such as here:

K-RAG FLINT-to-Chunk Templates

Act Template. "Act: ***{act}*** hasAction: ***{action}*** hasActor: ***{actor}*** hasObject: ***{object}*** hasPrecondition: ***{precondition}*** hasRecipient: ***{recipient}*** terminates: ***{fact}*** creates: ***{fact}***."

Complex Fact Template. "ComplexFact: ***{fact}***"

Duty Template. "Duty: hasClaimant: ***{claimant}*** hasHolder: ***{holder}***"

If for example multiple actors are present, the "hasActor: ***{actor}***" section is repeated.

In total, the FLINT knowledge corpus contains 114 FLINT frames, of which 18 are from the NOAPA, 36 are from the PA and 59 are from the AA. In general, there are more Acts (72) compared to Facts (26) and Duties (12). The complete list of FLINT frames per law can be found in Table 2.

4. Method

To explore whether using attribution to a formalization improves performance, we compare four different approaches (as shown in Figure 2) on the created dataset:

- LLM-only: using the general knowledge of the LLM (no RAG).
- RAG: retrieval from the law text.
- K-RAG: retrieval from the FLINT frames. The FLINT-to-chunk sentences are used and retrieved similarly to RAG.
- C-RAG: combined RAG based on the law text and the FLINT frames.

For all RAG methods, we use MULTILINGUAL-E5-LARGE text embedding model (Wang et al., 2024)

[11] https://gitlab.com/normativesystems/use-cases-public

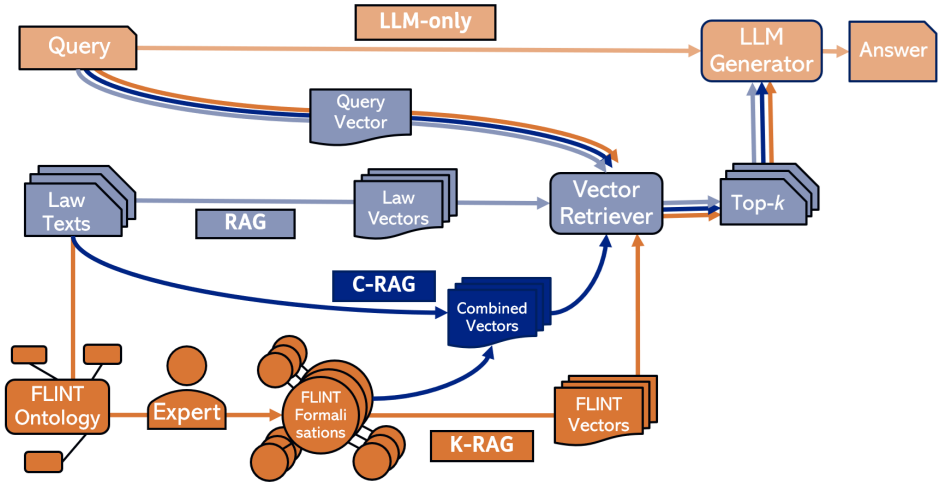

Figure 2: A diagram illustrating the four different methods that we compare: LLM-only, RAG, K-RAG (based on knowledge from FLINT frames) and C-RAG (RAG and K-RAG combined).

with batches of $batchsize = 32$, as this performed best on the Dutch law dataset for preconditions (Redelaar et al., 2024). We use $k = 5$, which means that we retrieve up to 5 relevant documents. As a generator, we use GPT-4.1-mini (2024-12-01-preview), as this has a good balance in performance and cost (Muennighoff et al., 2022)[12]. We set the $temperature = 0.00000001$ since we want less "creative" results. Similar to (Redelaar et al., 2024), we instruct the model to respond with "ANSWER:" before giving their answer to the question and "DOC IDs:" before citing the used documents (either text, FLINT frame or both). The full prompt can be found in Appendix A.1. Additionally, we run the generation step for each method with the ground truth retrieval to view performance with optimal retrieval.

4.1. Evaluation

Following Redelaar et al. (2024), we use the following metrics for retrieval:

- Hit Rate: $h_r = \frac{\#h}{\#q}$, where $\#h$ is the number of hits: whether a relevant item is present in the retrieved k documents (0 or 1), $\#q$ is the number of queries.
- ALCE's Citation Precision (Gao et al., 2023): the average of the relevant citations across all citations in the dataset. Relevance is defined as supporting a statement independently or after combining with a subset of the remaining citations.
- ALCE's Citation Recall (Gao et al., 2023): the average of all correctly cited passages that fully support the statement (as determined by an NLI model).

We use the following metrics for generation, also based on Redelaar et al. (2024):

- ROUGE-L-SUM (Lin, 2004): ROUGE-L-SUM splits the text into sentences and computes the Longest Common Subsequence for each pair of sentences and takes the average score for all sentences.
- METEOR (Banerjee and Lavie, 2005): takes the harmonic mean of precision and recall.
- MAUVE (Pillutla et al., 2021): computes Kullback–Leibler (KL) divergences between the two distributions in a quantized embedding space of a foundation model.
- G-EVALs Coherence (Liu et al., 2023): "measures the quality of all sentences collectively, as whether they fit together and sound natural. This metric considers the quality of the answer as a whole and takes into account whether the answer is well-structured."
- G-EVAL's Consistency (Liu et al., 2023): "measures the factual alignment between the human answer and the language model answer. A factually consistent answer contains only statements that are entailed by the source document. Answers are penalized when there are hallucinated facts."
- G-EVAL's Fluency (Liu et al., 2023): "the quality of the language model's answer in terms of grammar, spelling, punctuation, word choice, and sentence structure. The answer should be easy to read and follow."

[12] https://platform.openai.com/docs/models/gpt-4.1-mini

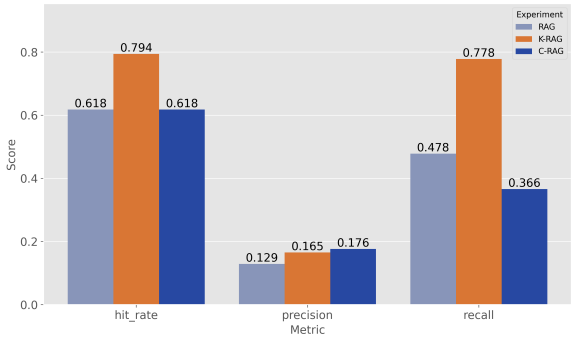

Figure 3: E5-multilingual-large Retrieval results on QuALA-NL

- G-EVAL's Relevance (Liu et al., 2023): "measures whether the answer merely contains important and relevant information to the question. Answers are penalized when containing redundancies and excess information."

For ROUGE, METEOR and MAUVE the evaluate module by Huggingface is used. G-EVAL metrics are run with GPT-4.1-mini, using a low temperature (0.00000001). All G-EVAL metrics are divided by 100 to obtain a score between 0 and 1.

We evaluate all methods per law, assuming users know which law is relevant to the question. We average the performance on each law to create the scores across all laws.

5. Results

5.1. Retrieval

Figure 3 shows the retrieval scores. The results show that the hit rate for K-RAG is higher (0.794) than that of RAG and C-RAG (0.618). C-RAG has the highest precision, whereas K-RAG has the highest recall, followed by RAG.

A closer inspection of the results reveals that typically two law texts (1.96 $\pm$ 1.03) and two FLINT frames (2.26 $\pm$ 1.40) are retrieved, and it is uncommon for all 5 (k) to be obtained. For C-RAG the average number of documents (including law texts and FLINT frames) retrieved is 2.02 ($\pm$ 1.08), and in more than 80% of the cases it retrieved law texts, not FLINT frames.

Comparing the results from the different laws, we observe that the Participation Act has a hit rate of 0.471 for C-RAG, followed by 0.412 for RAG and 0.324 for K-RAG, mainly due to a low precision (0.065). For the other two laws, the results are comparable to the averaged result in Figure 3.

5.2. Generation

Figure 5 shows the generation scores. The runs with optimal retrieval are shown as a dotted line. The results show that the ROUGE scores of RAG and C-RAG are higher than K-RAG. LLM-only has the lowest score. This is consistent with the MAUVE scores, although these scores are much higher. The METEOR scores show a slightly different trend: K-RAG scores better than RAG and C-RAG. When looking at the optimal retrieval, all methods would have had almost the same METEOR score, except the LLM-only.

The G-EVAL scores are generally very high, especially in fluency. The K-RAG method has the lowest performance in consistency, coherence and relevance. LLM-only has very high performance on all metrics except for consistency. Both RAG and K-RAG have a higher consistency score with the ground truth retrieval, where K-RAG would have a lower score for consistency, relevance and MAUVE.

On average, RAG answers contain 92.01 words ($\pm$ 34.66), whereas K-RAG contains 53.37 words ($\pm$ 23.96) and C-RAG contains 90.02 words ($\pm$ 35.41). For LLM-only, the average number of words in the answers is 164.72 words ($\pm$ 59.97).

A qualitative analysis of the results reveals that the generated answers are strongly dependent on the provided source, both in terms of content and style. To illustrate this, a translated example is shown of the generated answers of the various methods to a question based on the Aliens Act. The Act frame that corresponds to this question was given in Table 1.

Figure 4 shows an example of generated answers. The example shows that the K-RAG answer does not make use of the preconditions, even though they are provided to the LLM. A qualitative analysis indicates this is the case in most if not all cases. Interestingly, in most cases, the C-RAG answers are almost identical to the RAG answers, even in our optimal retrieval setup.

6. Discussion

The retrieval results in Section 5.1 show that K-RAG has the highest hit rate. This could partly be explained by the number of FLINT frames compared to law text chunks, as there are less FLINT frames than law text chunks. However, it does show that it is easier to retrieve the relevant FLINT frames, especially in terms of recall. As the frames are action-based and often capture all information about a certain question, there is only one FLINT frame needed to answer the question, whereas typically multiple chunks are needed to answer the question. The performance on C-RAG shows that it is easier to retrieve attributions to the law text than those to FLINT frames. In future work it is interesting to explore a retrieval method that improves the retrieval of Knowledge Graphs.

The generation results in Section 5.2 are not as

Example of generated answers

Question. When can a permanent residence permit be revoked?

LLM-only answer. A regular permanent residence permit can be revoked in the following situations:
1. **Obtaining under false pretenses**: If the permit was obtained by providing incorrect or incomplete information.
2. **Threat to public order or national security**: When the permit holder poses a threat to public order, security, or national interests of the Netherlands.
3. **Long-term residence outside the Netherlands**: If the permit holder resides outside the Netherlands for more than six consecutive years, the permit can be revoked.
4. **Loss of right of residence**: For example, due to acquiring Dutch nationality or voluntarily surrendering the permit.
5. **Other legal grounds**: As stipulated in the Aliens Act and associated regulations.

RAG answer. A permanent residence permit will be revoked if the holder is granted another residence permit ex officio as referred to in Article 33. In addition, the permanent residence permit may be revoked if: the holder has established their principal residence outside the Netherlands; the alien has provided incorrect information or withheld information that would have led to the application being rejected; the holder has been convicted by a final court judgment of a crime punishable by a prison sentence of three years or more, or a measure as referred to in Article 37a of the Dutch Criminal Code has been imposed; or the alien poses a threat to national security. Further rules on these grounds may be established by or pursuant to a general administrative order.

K-RAG answer. A permanent residence permit can be revoked by Our Minister of Justice and Security.

C-RAG answer. *See RAG answer*.

Figure 4: An example question and answers generated by various methods: LLM-only, RAG, K-RAG and C-RAG.

straightforward. Overall, RAG and C-RAG have the highest scores, sometimes matched by LLM-only and, on one occasion, by K-RAG. A qualitative analysis of the results indicates that K-RAG responses do not consistently address the question, and the information omitted is typically found in the precondition of the FLINT frames, despite this information being provided to the LLM. Moreover, the C-RAG and RAG answers are nearly identical, even with optimal retrieval. This indicates that the FLINT frames are not used to generate answers to the extent that the law texts are. Future research could concentrate on using the provided knowledge graph more effectively.

When comparing the results with Redelaar et al. (2024) (p=0.692, r=0.539, $h_r@3 = 0.87$), we observe that the retrieval performance on QuALA-NL is slightly lower (RAG: p=0.129, r=0.478, $h_r@5$=0.618). This shows that our questions are more challenging. Additionally, the generation performance of the G-EVAL methods is similarly high on coherence, fluency and relevance, as most GPT models have a very high performance on these metrics. Performance on consistency (0.734 vs. 0.934) and METEOR (0.298 vs. 0.754) is much lower. This performance is expected, as the answers by Redelaar et al. (2024) are closer to the original source, and therefore the answers are closer to the answer in the dataset.

The present study has several limitations. First, it was not possible to use real-world questions from the municipality, nor was it feasible to validate the answers of QuALA-NL with the intended user group. We did, however, prepare the answers with three legal and FLINT experts. Second, the dataset comprises questions derived from only three laws. Expanding both the number of questions and the legislative sources requires FLINT interpretations for additional laws, a process that remains resource-intensive. Third, we used the same GPT model for both generation and evaluation with G-EVAL, which may slightly bias the performance. Future work could involve using GPT-5 for generation, although this would need updated prompts and parameters. Finally, the K-RAG method itself can be improved. The method implemented in the paper is one of many possible approaches to use KGs as a database for RAG-based QA. Future work includes conducting a comprehensive comparison of different GraphRAG approaches that incorporate the formalization across various settings.

7. Conclusion

In this paper we introduce QuALA-NL, a dataset that provides over 100 question-answer pairs with explicit attributions to both the law text and a legal formalization. This dataset was developed specifically to support research on knowledge-based Retrieval Augmented Generation and is intended for use in that context. To demonstrate the capabilities of the dataset and its potential applications, we implemented a RAG pipeline, comparing different implementations and evaluating with an extensive number of metrics. Compared to the dataset by Redelaar et al. (2024), our dataset yields lower re-

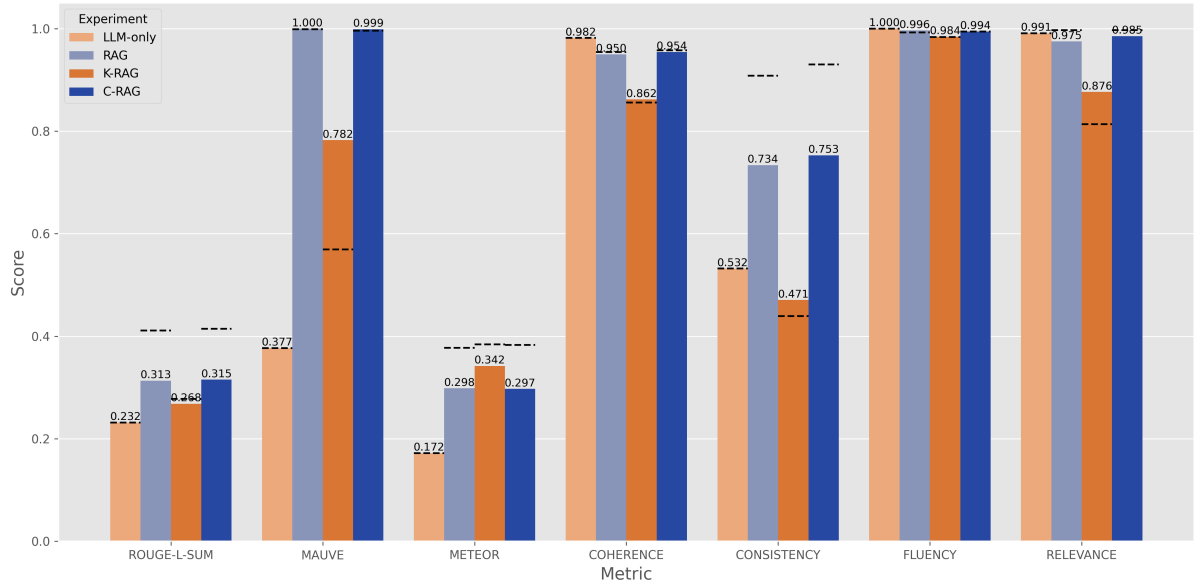

Figure 5: Generation results on QuALA-NL. The dotted line represents scores using optimal retrieval.

trieval and generation performance using the same method, indicating the dataset presented here is more challenging. Our current pipeline leaves room for improvement in both retrieval and generation, particularly in enhancing the knowledge-based RAG approach.

In future work, we intend to use the dataset in more experiments, extending the pipeline with other K-RAG methods. We also aim to use the pipeline in a real-world setting, such as at the municipality where people are working with questions from citizens.

Acknowledgements

We would like to thank the Dutch Ministry of the Interior and Kingdom Relations for funding our research. Special thanks to Thom van Gessel, Jeroen Breteler, and Robert van Doesburg for their help on interpreting legal text in FLINT, and to Ioannis Tolios and Erik Boertjes for developing the editor used to create these interpretations.

8. Bibliographical References

Tara Athan, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Wyner. 2015. Legalruleml: Design principles and foundations. In *Reasoning Web International Summer School*, pages 151–188. Springer.

Roos Bakker, Romy van Drie, Maaike de Boer, Robert van Doesburg, and Tom van Engers. 2022a. Semantic role labelling for dutch law texts. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 448–457.

Roos M Bakker, Maaike HT de Boer, Romy AN van Drie, and Daan Vos. 2022b. Extracting structured knowledge from dutch legal texts: A rule-based approach. In *EKAW (Companion)*.

Roos M. Bakker, Akke J. Schoevers, Romy A. N. van Drie, Marijn P. Schraagen, and Maaike H. T. de Boer. 2025. Semantic role extraction in law texts: a comparative analysis of language models for legal information extraction. *Artificial Intelligence and Law*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jeroen Breteler, Thom van Gessel, Giulia Biagioni, and Robert van Doesburg. 2023. The flint ontology: an actor-based model of legal relations. In *Knowledge Graphs: Semantics, Machine Learning, and Languages*, pages 227–234. IOS Press.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Ruixi Chen. 2025. Retrieval-augmented generation with knowledge graphs: A survey. In *Computer Science Undergradaute Conference 2025@ XJTU*.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. 2024. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

VAH Firdaus, PY Saputra, and D Suprianto. 2020. Intelligence chatbot for indonesian law on electronic information and transaction. In *IOP Conference Series: Materials Science and Engineering*, volume 830, page 022089. IOP Publishing.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Cristine Griffo, João Paulo A Almeida, and Giancarlo Guizzardi. 2018. Conceptual modeling of legal relations. In *International Conference on Conceptual Modeling*, pages 169–183. Springer.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Rinke Hoekstra, Joost Breuker, Marcello Di Bello, Alexander Boer, et al. 2007. The lkif core ontology of basic legal concepts. *LOAIT*, 321:43–63.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2025. Grag: Graph retrieval-augmented generation.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Soha Khazaeli, Janardhana Punuru, Chad Morris, Sanjay Sharma, Bert Staub, Michael Cole, Sunny Chiu-Webster, and Dhruv Sakalley. 2021. A free format legal question answering system. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 107–113.

Valentina Leone. 2021. Legal knowledge extraction in the data protection domain based on ontology design patterns.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Jorge Martinez-Gil. 2023. A survey on legal question–answering systems. *Computer Science Review*, 48:100552.

Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. 2024. Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6):btae353.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Marc Queudot, Éric Charton, and Marie-Jean Meurs. 2020. Improving access to justice with legal chatbots. *Stats*, 3(3):356–375.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Felicia Redelaar, Romy Van Drie, Suzan Verberne, and Maaike De Boer. 2024. Attributed question answering for preconditions in the dutch law. In *Proceedings of the natural legal language processing workshop 2024*, pages 154–165.

Diego Sanmartin. 2024. Kg-rag: Bridging the gap between knowledge and creativity. *arXiv preprint arXiv:2405.12035*.

Vorada Socatiyanurak, Nittayapa Klangpornkun, Adirek Munthuli, Phongphan Phienphanich, Lalin Kovudhikulrungsri, Nantawat Saksakulkunakorn, Phonkanok Chairaungsri, and Charturong Tantibundhit. 2021. Law-u: Legal guidance through artificial intelligence chatbot for sexual violence victims and survivors. *IEEE Access*, 9:131440–131461.

Robert van Doesburg. 2017. A formal method for interpretation of sources of norms. Technical report, Leibniz Center for Law, University of Amsterdam.

Thom van Gessel, Giulia Biagioni, Jeroen Breteler, Ioannis Tolios, and Erik Boertjes. 2023. A toolset for normative interpretations in flint. In *SEMANTiCS (Posters & Demos)*.

Thom van Gessel and Frank Goossens. Formalizing flint: A logic of preconditions and consequences of actions. *17 DEON*, page 407.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering*, 29(12):2724–2743.

G Wiggers. 2023. *The relevance of impact: bibliometric-enhanced legal information retrieval*. Ph.D. thesis, Leiden University.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. *arXiv preprint arXiv:2202.13296*.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025. Knowledge graph-guided retrieval augmented generation. *arXiv preprint arXiv:2502.06864*.

9. Language Resource References

Chalkidis, Ilias and Jana, Abhik and Hartung, Dirk and Bommarito, Michael and Androutsopoulos, Ion and Katz, Daniel Martin and Aletras, Nikolaos. 2021. *LexGLUE: A benchmark dataset for legal language understanding in English*.

Chen, Andong and Yao, Feng and Zhao, Xinyan and Zhang, Yating and Sun, Changlong and Liu, Yun and Shen, Weixing. 2023. *EQUALS: A real-world dataset for legal question answering via reading chinese laws*.

Hoppe, Christoph and Pelkmann, David and Migenda, Nico and Hötte, Daniel and Schenck, Wolfram. 2021. *Towards intelligent legal advisors for document retrieval and question-answering in German legal documents*. IEEE.

Louis, Antoine and van Dijck, Gijs and Spanakis, Gerasimos. 2023. *Interpretable long-form legal question answering with retrieval-augmented large language models*.

Mansouri, Behrooz and Campos, Ricardo. 2023. *FALQU: Finding Answers to Legal Questions*.

van Drie, Romy AN and de Boer, Maaike HT and Bakker, Roos M and Tolios, Ioannis and Vos, Daan. 2023. *The Dutch Law as a Semantic Role Labeling Dataset*.

Zhong, Haoxi and Xiao, Chaojun and Tu, Cunchao and Zhang, Tianyang and Liu, Zhiyuan and Sun, Maosong. 2020. *JEC-QA: a legal-domain question answering dataset*.

A. Appendix

A.1. Prompt

The following prompt is translated from Dutch to English, and the document text in the example is

shortened. The original Dutch prompt can be found in repository 2.

```
You will be given a question and a list of
    5 documents that are retrieved by BM25.
     The retrieved documents contain
    content that is the most relevant to
    the question from a large corpus.
Your task is to generate 2 things as an
    output. 1: An answer to the question
    based on the set of documents provided,
     and 2: A list of attributions to the
    documents you have used to generate
    your answer. Note that not all of these
     5 documents are relevant to the answer
    . BM25 simply returned the documents
    most likely to be relevant to the
    question.
Please make sure you read and understand
    these instructions carefully. Please
    keep this document open while reviewing
    , and refer to it as needed.
Steps:
1. Read the question carefully and identify
     the main topic and key points.
2. Read the documents provided by BM25 and
    check if they contain information that
    are directly relevant for the answer to
     the question.
3. Generate the answer string that answers
    the question based on the documents
    provided. Starting your answer with "
    ANSWER:".
4. Create a list of the document IDs that
    you have used for your answer, so the
    user can cross-check. Do only use the
    IDs of documents you have actually used
     to generate your answer. Start your
    answer with "DOC IDS:".
Example 1:
Question (EXAMPLE OF THE INPUT YOU WILL
    RECEIVE):
Question: In which conditions can a
    pronouncement of undesirability of an
    alien be lifted?
Documents (EXAMPLE OF THE INPUT YOU WILL
    RECEIVE):
Potential relevant documents: [['DOC0068',
    'Alien act 2000 Residence Lawful
    residence Article 9... *the full text
    can be found in the repository]]
(EXAMPLE OF YOUR OUTPUT - ALWAYS DO IT IN
    THIS FORMAT, CAPITALIZING THE "ANSWER
    :"):
ANSWER: The declaration of undesirability
    is lifted under the following
    conditions: if the foreign national has
     stayed outside the Netherlands for ten
     consecutive years and the foreign
    national has not been declared
    undesirable.
(EXAMPLE OF YOUR OUTPUT - ALWAYS DO IT IN
    THIS FORMAT, CAPITALIZING "DOC IDS:",
    THERE CAN BE ONE OR MORE.):
DOC IDS: DOC0226
REMEMBER, YOUR TASK IS TO GENERATE AN
    ANSWER STRING AND A LIST FOR THE
    DOCUMENT IDs USED IN YOUR ANSWER.
ALWAYS START THE ANSWER WITH: "ANSWER:",
    AND THE ATTRIBUTION WITH "DOC IDS:".
ALWAYS USE CAPITALIZATION FOR ANSWER AND
    DOC IDS IN YOUR OUTPUT! DO NOT HAVE ANY
     OTHER OUTPUT.
Here comes your task:
```