Attributed Question Answering for Preconditions in the Dutch Law

Felicia Redelaar[1,2], Romy van Drie[2], Suzan Verberne[1], Maaike de Boer[2]

[1]Leiden Institute of Advanced Computer Science, Einsteinweg 55, Leiden, The Netherlands
[2]Dep. Data Science, TNO, Anna van Buerenplein 1, The Hague, The Netherlands
Correspondence: s.verberne@liacs.leidenuniv.nl, maaike.deboer@tno.nl

Abstract

In this paper, we address the problem of answering questions about preconditions in the law, e.g. "When can the court terminate the guardianship of a natural person?". When answering legal questions, it is important to attribute the relevant part of the law; we therefore not only generate answers but also references to law articles. We implement a retrieval augmented generation (RAG) pipeline for long-form answers based on the Dutch law, using several state-of-the-art retrievers and generators. For evaluating our pipeline, we create a dataset containing 102 legal QA pairs with attributions. Our experiments show promising results on our extended version for the automatic evaluation metrics from the Automatic LLMs' Citation Evaluation (ALCE) Framework and the G-EVAL Framework. Our findings indicate that RAG has significant potential in complex, citation-heavy domains like law, as it helps laymen understand legal preconditions and rights by generating high-quality answers with accurate attributions.

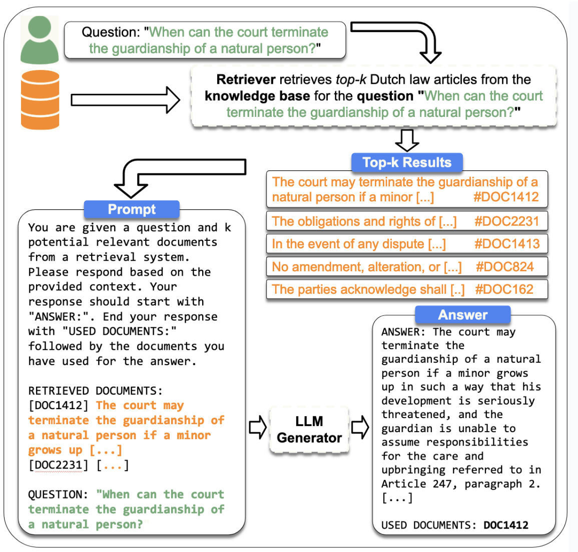

Figure 1: Our RAG framework for legal attributed QA with an example

1 Introduction

Many people encounter civil justice problems at some point in their lives, whether they are disagreements with landlords or issues at work. However, not everyone knows their rights or how to resolve these problems, leaving them unsure of what to do next (Balmer et al., 2010). Studies have shown that the main obstacles to getting justice are the costs involved and a lack of awareness about legal rights and available options (Hoekstra and Teeuwen, 2023). This issue is not just local – it is a global problem. Over 1.4 billion people around the world have unresolved civil justice needs (Ponce et al., 2019), and in a global survey, 43% of respondents said that legal issues had negatively affected their personal lives (Ponce et al., 2019).

Automated legal Question Answering (QA) could provide affordable assistance to a wide audience. One concern is that many countries still lack a digital legal aid system, as each country operates under its own legal framework (Wiggers, 2023) and in their local language. This requires language-specific QA solutions, such as those explored by Louis et al. (2023), which focuses on developing a system capable of answering legal questions in French for Belgian law. Prior initiatives have been undertaken to assist individuals by creating legal chatbots for various languages, including French, Thai, and Indonesian (Queudot et al., 2020; Socatiyanurak et al., 2021; Firdaus et al., 2020).

An important requirement of legal QA systems is that they should provide verifiable sources in their responses, so-called **attributions**, in order to increase the verifiability of the responses. Additionally, the answers generated by these systems should be tailored to an individual's specific legal situation and provide detailed information about their legal options. Furthermore, the responses should not be too brief, such as simple "yes" or "no" answers (Do

Proceedings of the Natural Legal Language Processing Workshop 2024, pages 154–165
November 16, 2024 ©2024 Association for Computational Linguistics

et al., 2017), which fail to capture the complexity of legal issues.

In this paper we focus on **precondition**-related questions. We define a precondition to refer to the specific requirements, criteria, or circumstances that must be fulfilled before a specific action, event, decision, or outcome can legally occur or be finalized. Some examples of precondition-related questions are: "When is a student eligible for student financing?" and "What are the requirements for entering into a marriage?". The answer to the first question contains preconditions such as having a Dutch nationality. The latter question can be answered with the precondition of being at least 18 years old.

We address **Attributed QA (AQA) for the Dutch law**. Our aim is to answer legal questions with a tailored answer, including attributions to the relevant law article. Attributions have the form of references to specific documents, in our case articles of the Dutch law.

To this end, we create and publish a Dutch legal QA dataset, consisting of 102 question-answer pairs with attributions to Dutch law articles. We implement and evaluate a Retrieval Augmented Generation (RAG) pipeline that generates informative long-form answers to Dutch law questions, where each generated answer contains a list of attributions (references) to sources. Our approach is illustrated in Figure 1. Our contributions are as follows:

- We implement a RAG solution for attributed QA for the Dutch law.
- We have created and released an annotated dataset for (attributed) QA for the Dutch law that can be used in legal QA tasks. The dataset consists of 102 question-and-answer pairs that have an attribution to the used law articles. The answers have been verified by a legal expert on legal correctness.
- We extend an existing method for the automatic evaluation of attributed QA tasks. Our extended evaluation method is especially suitable when the answers in the dataset contain ground-truth attributions.

We publicly release our code and dataset at `https://gitlab.com/normativesystems/flintfillers/aqa_preconditions`.

2 Related work

2.1 Legal Question Answering

Legal questions can expect binary, multiple-choice, multi-span, or long-form answers (Martinez-Gil, 2023). QA systems are commonly implemented as two-stage pipelines, consisting of a retrieval step followed by an extraction or generation step (Martinez-Gil, 2023). Traditionally, the first stage of legal QA relied on sparse (keyword-based) retrieval techniques. With the rise of transformers, several works have incorporated dense retrievers to improve the first stage in their legal QA system (Hoppe et al., 2021; Khazaeli et al., 2021; Karpukhin et al., 2020). Dense retrievers embed both the query and the document as a vector in a continuous vector space, which allows to find relevant documents that have semantic similarity to the query but no or very little word overlap.

The most recent advancements in the field are in the second stage of the legal QA pipeline, using LLMs to generate fluent answers (Louis et al., 2023). This work employs the conventional two-stage method to answer long-form legal question, using an LLM to generate answers. Finally, their methodology involves generating rationale for answers that include a pointer towards a knowledge base.

Datasets for legal QA have been released in prior work: Zhong et al. (2020) released JEC-QA, a Chinese dataset for multiple-choice questions, sourced from legal exams. For long-form questions, Mansouri and Campos (2023); Chen et al. (2023) released English and Chinese datasets sourced from online forums, and Louis et al. (2023) released a French dataset sourced from lawyers. We are the first initiative creating, curating, and releasing a dataset for Dutch legal QA.

2.2 Attributed QA

In Attributed Question Answering (AQA), the input is a question, and the output is a tuple of an answer string and its attributions (Bohnet et al., 2022). The attributions are references to a knowledge corpus $\mathcal{C}$. An example for the AQA task is the input question "Which movies have Cate Blanchet as a member of their cast?", which should produce an output answer string: "Carol, The Lord of the Rings, Tár, and Don't Look Up", with attribution references e.g. in the form [DOC1][DOC2]. These references are pointers to text segments in a knowledge corpus that support the given answer string.

AQA is commonly solved with RAG (Li et al., 2024; Muller et al., 2023; Stolfo, 2024; Hu et al., 2024; Menick et al., 2022). While most of these works achieve attributions through prompting, Ye et al. (2024) propose an approach in which they fine-tune an LLM to generate references. In the context of cross-lingual QA, Muller et al. (2023) improve attribution quality using Natural Language Inference.

2.3 Evaluation of AQA

Multiple studies have suggested methods for evaluating the answers and attributions generated by an LLM. Some studies involve manual human evaluation assessing whether the answer is supported by the given attributions and whether the answer itself is plausible (Menick et al., 2022). Kamalloo et al. (2023) introduced HAGRID, which measures whether the explanation directly answers the question and whether the explanation is attributable to the attributions. For automatic evaluation, several studies have proposed prompting LLMs to generate evaluations. In a study by Yue et al. (2023), the AttrScore was proposed, which evaluates three binary metrics. These are whether the answer is attributable, extrapolatory, and contradictory. Additionally, Li et al. (2023) proposed KALMA, an automatic evaluation framework that assesses the generated text and its citations. The generated text is evaluated using G-EVAL (Liu et al., 2023), an evaluation suite that uses LLMs with chain-of-thoughts to measure coherence, consistency, fluency, and relevance. In KALMA, the citations are automatically evaluated using precision and recall.

Based on the work of Bohnet et al. (2022), Gao et al. (2023) introduced Automatic LLMs' Citation Evaluation (ALCE), which is the first benchmark for AQA. The benchmark contains three datasets: ASQA (Stelmakh et al., 2022), QAMPARI (Amouyal et al., 2022), and ELI5 (Fan et al., 2019). Our work will not use these benchmark datasets because they do not contain ground truth attributions to a knowledge corpus. ALCE serves as a framework for automatically evaluating answer strings and their corresponding cited attributions generated by LLMs. The authors of ALCE developed automatic metrics along three dimensions and demonstrated their strong correlation with human judgments. In our evaluation, we extend ALCE for the evaluation of AQA with ground truth references.

2.4 Retrieval Augmented Generation

RAG, introduced in the work by Lewis et al. (2020), is a technique that augments the prompt to an LLM with external knowledge. RAG is particularly relevant for attributed QA as it allows external knowledge to be used to answer questions, while the attributions can be generated by the LLM. The main components of RAG are the retriever and the generator. The retriever aims to find the most relevant documents in a large knowledge corpus for a specific query or question. An LLM is then used to generate an answer. The main motivation of RAG is two-fold. Firstly, to use custom data, since LLMs have been trained on a huge amount of data that might not be aligned for a specific task. On top of that, the data the LLM has been pre-trained with could be outdated or contain inaccuracies. Secondly, to give the user access to the sources of the generated information, allowing them to verify its correctness and ensure the information is accurate and reliable.

Substantial research has been dedicated to optimizing retrievers for QA tasks (Chen et al., 2017). Karpukhin et al. (2020) propose Dense Passage Retrieval and showed that a dense retriever can outperform sparse vector space models such as BM25 when adding enough data. SPLADE (Formal et al., 2021) is a retriever that combines dense and sparse retrieval and has been successfully used in RAG contexts. Lin et al. (2023) introduced Dragon, which is a generalized dense retriever trained through progressive data augmentation. Ram et al. (2023) proposed Ralm to optimize the retriever for in-context retrieval-augmented LLMs. In this paper, we follow this line of work and evaluate state-of-the-art retrieval models, both dense and sparse, in the context of RAG for attributed legal QA.

3 Dataset

3.1 Creating question-answer pairs

Our work aims to help users better understand when they are legally permitted to take certain actions, which is why we focus exclusively on precondition-related questions.

To select sources for our questions, we carefully review Dutch law texts via the official government website[1] and reading these on the **article** level. We filter out all technical or administrative legislation.

[1] https://wetten.overheid.nl/

These are laws that are intended to adjust, implement, or execute existing legislation without making policy changes.[2] From the remaining laws, we sample 25 laws at random for question formulation. These 25 laws comprise a total of 4441 articles.

We formulate the questions by looking for subordinating conjunctions such as “only if” or “on condition (that)”. Whenever we find such conjunctions followed by actionable measures in a law text, we formulate a legal question. We formulate a ground truth answer to the question by referencing the relevant law texts, aiming to maintain the original meaning as closely as possible, considering the complexity and potential ambiguities in legal texts. In other words, we first look for the answers by finding pre-conditions in law texts, and then formulate legal questions around these pre-conditions. We formulated questions to 17 of the 25 laws and created 110 questions–answer pairs based on these laws.

Next, a legal expert is consulted for quality assurance, checking the legal correctness of all the questions and answers. The legal expert assessed whether the answer to the question was an accurate representation of the source document. We implement the expert’s feedback regarding question and answer pairs, and discard questions that are too vague or contained answers that are too complex to verify for legal correctness according to the expert. Finally, the legal expert is consulted again to make sure the dataset quality is up to par in terms of legal correctness and completeness of the questions and answers. This yields a final number of 102 question-answer pairs, each with legal attribution references.

3.2 Knowledge corpus

We use the Dutch law as the knowledge corpus to provide references that a system can use when generating an answer.

The laws are publicly available and downloadable in XML format from the official government website.[1] With a parsing script we convert the laws from XML into a CSV file in which each row contains the text of a law article. Articles longer than 150 words are split into new rows to make the references in the answer easier to verify since some articles are over 1000 words in length. We use a hard cut-off after 150 words. This results in some law articles having multiple chunks in our knowledge corpus, each following the other on a word basis. A downside of the cut-off is some loss in the meaning of the split chunks which might affect retrieval and generation performance. Finally, for each created chunk, we assign a unique document ID to facilitate straightforward referencing by the system.

The resulting number of articles is 22,462 and the number of chunks is 30,803. Most articles (16,665) contain one chunk since their text consists of 150 words or less. The remaining articles are comprised of 2 to 20 chunks. By design, 100% of our curated QA pairs contain article-level references through chunks in the knowledge corpus. This approach ensures that each answer can be traced back to a specific legal article, enhancing the reliability and traceability of the dataset. By design, 100% of our questions–answer pairs contain article-level references. This approach ensures that each answer can be traced back to a specific legal article.

4 Methods: RAG system

4.1 Retrievers

We experiment with three types of retrievers: 1) sparse retrievers; 2) dense retrievers; 3) hybrid retrievers. Sparse retrievers focus on the lexical overlap of terms between the query and the documents, relying on traditional information retrieval methods such as term frequency-inverse document frequency (TF-IDF) (Sparck Jones, 1972). Our work uses the BM25 ranking model (Robertson et al., 1995) as a baseline retriever, which relies on TF-IDF.

While sparse retrievers are computationally efficient and interpretable, they are limited to word overlap between the query and the relevant documents. This means that sparse retrievers potentially miss relevant documents that do not share exact terms with the query. Dense retrievers do incorporate richer semantic information in the form of embeddings. In our work, we compare the following embedding models that we use as our dense retrievers:

- ALLNLI-GRONLP-BERT-BASE-DUTCH-

[2]These laws have the following words in their title: aanpassingswet (adjustment act), aanwijzingswet (designation act), verzamelwet (collection act), implementatiewet (implementation act), belastingplan (tax plan), intrekkingswet (withdrawal act), invoeringswet (introduction act), overige fiscale maatregelen (other fiscal measures), tijdelijke wet (temporary act), uitvoeringswet (execution act), wet aanpassing (law adjustment), wet aanvullende (supplementary law), and wijzigingswet (amendment act).

CASED: a sentence-BERT model (Reimers and Gurevych, 2019) trained on Dutch text;
- PARAPHRASE-MULTILINGUAL-MINILM-L12-V2: a multilingual sentence BERT model;
- MULTILINGUAL-E5, an open source text embedding model (Wang et al., 2024), the small, base and large version;
- DRAGON (Lin et al., 2023), trained using progressive data augmentation, but not multilingual (trained on English).

Note that several legal BERT models exist, but none of these was pre-trained for retrieval (in a bi-encoder or cross-encoder setting). Hybrid retrievers combine the strengths of sparse and dense retrievers to enhance the performance and accuracy of information retrieval systems by integrating precise keyword matching of sparse retrievers with the semantic understanding of dense retrievers. In this work, we use Sparse Lexical and Dense Embeddings (SPLADE) (Formal et al., 2021) as hybrid retriever. SPLADE has been trained on English datasets and is not a multilingual model.

4.2 Generators

After retrieving the most relevant documents, our generator is instructed through a prompt to generate a long-form answer using the potentially relevant retrieved documents and the corresponding question. We use one-shot in-context learning, and provide the prompt in the language (Dutch or English) in which the LLM has been mostly pre-trained on. Our prompt is shown in Table 4 in Appendix A.

We experiment with four commercial GPT models by OpenAI.[3] We choose the models GPT-3.5-TURBO and GPT-4O. The first model is a fast, inexpensive model used for simple tasks and the latter model is currently the fastest and most affordable flagship model by OpenAI.

Furthermore, we experiment with three open-source models. These are GEITJE-7B-ULTRA[4], LLAMA-3-8B-DUTCH[5], and FIETJE-2-INSTRUCT.[6] We select GEITJE-7B-ULTRA since it currrently is the largest open-source Dutch language model. The model is based on MISTRAL-7B, which reports to outperform LLAMA 2 on all benchmarks.[7] We select the LLAMA-3-8B-DUTCH model, since it has been trained on the same Dutch texts that GEITJE-7B-ULTRA has been trained on, but is based on Llama 3. Lastly, we experiment with FIETJE-2-INSTRUCT since we want to see the potential results of a substantially smaller model. Fietje is based on Microsoft's phi-2, further trained for Dutch. It has only 2.7 billion parameters. The instruct version of the model was created by finetuning the base model on Dutch-language chat datasets.

[3]https://platform.openai.com/docs/models
[4]https://huggingface.co/BramVanroy/GEITje-7B-ultra
[5]https://huggingface.co/ReBatch/Llama-3-8B-dutch
[6]https://huggingface.co/BramVanroy/fietje-2-instruct
[7]https://mistral.ai/news/announcing-mistral-7b/

5 Experiments

We first conduct experiments to select the best retriever component for our RAG pipeline. Once the best-performing retriever for each scenario is identified, we proceed to conduct experiments on the entire RAG pipeline using the best-performing retriever based on its recall@k score.

5.1 Setup

Regarding the parameters used for the retrievers, we have generated the embeddings of all SBERT and MULTILINGUAL-E5 models using batches of $batchsize = 32$. Regarding the parameters used for the generators, all GPT models generated text using $temperature = 0.0$ and $max\_tokens = 1000$. The open source LLMs generated text using a $temperature = 0.2$ and with $max\_tokens = 5000$. The temperature parameters were selected to be as low as possible since we have a dataset on legal work, and we want to minimize chances for rewording or creative output. For the GPT models, we use a temperature of 0.0 since we want less “creative” results, and still see variations in the output of the OpenAI models. With a temperature of 0.0, the GPT models' output still is non-deterministic. For the open source models, we use a temperature of 0.2 to create deviation in the responses. Furthermore, the max_token differs between the GPT models and open-source models is due to the open-source LLMs often requiring more tokens since they first describe that analyzed all given documents to formulate their answer, before actually providing their answer.

We generate answers with the OpenAI models ten times for robust results. For the open-source models, we generate answers five timesfor computational cost reasons. Their output, however, often deviated from the instructed format. For example, the models frequently ignore the required structure and begin their responses with a detailed analysis for each document. This means that their answers needs to be extracted manually before automatic evaluation is possible. We use top-K sampling with $K = 50$. We instruct all models to respond starting with "ANSWER:" before giving their answer to the question and "DOC IDs:" before citing the used documents. We automatically process the model's answers using a regular expression. We select the answers using the following rules whenever the models do not respond in this format. Firstly, if a variation of "ANSWER:" is present, we select all text that comes afterward as their response til the term "DOC IDs" appears. We disregard all texts after "DOC IDs" that are not DOC IDs. If no variation of "ANSWER:" exists in their response, we select the entire response as the output.

To run the experiments, we either used a laptop with an *Intel i7-1225U* processor, *Intel Iris Xe graphics* with 8GB, 16GB of RAM, or to accelerate the process, a part of the clusters containing dual *AMD EPYC 9354* CPUs (2x 32-core), 1TB of RAM, 8TB of SSD storage, and 4x *Nvidia L40S* GPUs. As for software, we used Huggingface for the transformer models, Pyserini[8] for BM25 and TREC Eval[9] to calculate the $recall@k$ score.

5.2 Evaluation

We base our automatic evaluation framework on the Automatic LLMs' Citation Evaluation (ALCE) (Gao et al., 2023). This framework developed automatic metrics among three dimensions – fluency, correctness, and citation quality. We argue that this framework could be more suitable for the AQA task by implementing small adjustments and we present these in our work. In our method, we still use the three dimensions but introduce different automatic evaluation methods for each dimension. We will discuss these in more detail in the following sections. Table 1 provides an overview of our and ALCE's evaluation's dimensions, definitions, and metrics.

5.2.1 Fluency

The ALCE framework uses MAUVE (Pillutla et al., 2021) to evaluate the fluency of the output as a sanity check, as most LLMs are capable of generating fluent text. However, the authors discovered that MAUVE is sensitive to the length of the output and found that its results become unstable for responses longer than 100 words. We therefore believe that MAUVE should not be used because of its instability. We instead use G-EVAL (Liu et al., 2023) to evaluate the output's fluency. Using G-EVAL with GPT-3.5-turbo, we prompt a detailed instruction to evaluate the fluency and coherence of an answer. The fluency metric measures 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. Coherence measures the quality of all sentences collectively, as whether they fit together and sound naturally. This metric considers the quality of the answer as a whole and takes in the account whether the answer is well-structured.

5.2.2 Correctness

The ALCE framework uses three different datasets and a different method for each to calculate the model response's correctness: exact match recall, recall@5, and a Natural Language Inference (NLI) model that is fine-tuned to check whether the model output entails sub claims created by another model based of the original model's response. In our work, we propose to use four metrics for the correctness score. Firstly, we decide to use the common metrics ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005). We are able to use ROUGE and METEOR since the dataset in our work contains ground truth answers. These metrics calculate the overlap between the ground truth and model's answer. Secondly, we add G-EVAL to calculate the consistency and relevance scores. The consistency 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. The relevance metric measures whether the answer merely contains important and relevant information to the question. Answers are penalized when containing redundancies and excess information.

5.2.3 Citation quality

The ALCE framework computes the citation quality using a Natural Language Inference (NLI) model. Specifically, the recall and precision of the entailment of each statement with its attribution (0 or 1) is averaged over all statements in the model response. The recall of a statement in the model's generated answer is 1 if the concatenation of all cited passages fully support the statement.

[8] https://github.com/castorini/pyserini
[9] https://github.com/cvangysel/pytrec_eval

Evaluation dimension	Definition	ALCE's Metrics	Our Metrics
1) Fluency	Whether the model's generated text is fluent and coherent	• MAUVE	• Fluency • Coherence
2) Correctness	Whether the answer is accurate and covers all aspects of interest	• Exact match recall • Recall@5, • Claim recall	• ROUGE-L • METEOR • Consistency • Relevance
3) Citation Quality	Whether the answer is well supported by the cited passages and no irrelevant passages are cited	• Citation recall • Citation precision	• Citation recall • Citation precision • HitRate@k

Table 1: Evaluation dimensions and associated metrics of ALCE and our work.

The NLI model is used to determine "full support". The precision in ALCE detects irrelevant citations. A cited passage is seen as irrelevant if the citation alone does not support a claim, and if removing it does not affect other citations combined to support the claim.

We use a more precise method for citation quality, which is possible since our dataset QA-pairs contain attribution ground truths. We simply use regular recall and precision for the citation quality.

6 Results

6.1 Retrieval

Table 2 shows the results on the retrieval part of the RAG. We compare the baseline model BM25 to dense and hybrid retrievers. We can see that the E5-MULTILINGUAL$_{\text{LARGE}}$ model provides the highest performance on all metrics.

6.2 Generation

Table 3 shows the results for our RAG pipeline using our QA dataset with the knowledge corpus. In this setup, *k=3*, and the MULTILINGUAL-E5-BASE model was used for the retrieval of the documents. Regarding the correctness of the answers and the citation metrics, the GPT models perform substantially better than the open-source LLMs. Specifically, GPT-4O showed the best performance across most metrics, while the GPT-3.5-TURBO model had the highest precision score. An example of the output of the generation per model is shown in Appendix B.

7 Discussion

The results show that our RAG system can generate fluent and correct answers with an 83.0% hit-rate. The answers are often highly coherent with the ground truth, and the models are capable of citing their sources accurately.

Looking at the retrievers, there are substantial differences. DRAGON consistently underperformed our baseline model, BM25. We hypothesize that this might be because DRAGON is a dense retriever trained solely in the English language, lacking multilingual capabilities. The E5 models, which were the best retriever models in our RAG system, were also trained using contrastive learning. Following E5, the hybrid model SPLADE, and the Dutch-trained SBERT performed best. This is an interesting finding, especially considering that SPLADE was trained only for the English language, leading us to hypothesize that its performance could be attributed to its partly sparse characteristics, enabling lexical overlap. While the models (4, 5, and 6 in Table 2) performed similarly in retrieval, using the best model improves the likelihood of correct attributions in the generated answers.

The results on the generation show that the proprietary models scored higher on all evaluation metrics than the open-source models. There are several explanations for the substantial difference, but the main one probably lies in the parameter sizes between the models. Although the number of parameters for the proprietary models used in our work remains undisclosed, it is reasonable to assume that they are significantly larger than the open-source models that we have used for our work which are relatively small models ranging between

	Model	#Param	R@3	R@5	R@10	Hit@3	Hit@5	Hit@10
Sparse								
1	BM25	-	0.586	0.672	0.739	0.696	0.775	0.873
Dense								
2	$\text{SBERT}_{\text{MULTILINGUAL}}$	117.7M	0.404	0.426	0.500	0.510	0.529	0.627
3	$\text{SBERT}_{\text{DUTCH}}$	109.1M	0.516	0.583	0.616	0.618	0.696	0.745
4	$\text{E5-multilingual}_{\text{SMALL}}$	117.7M	0.674	0.732	0.803	0.794	0.853	0.912
5	$\text{E5-multilingual}_{\text{BASE}}$	278.0M	0.696	0.755	0.816	0.843	0.892	0.941
6	$\text{E5-multilingual}_{\text{LARGE}}$	559.9M	**0.729**	**0.780**	**0.845**	**0.873**	**0.922**	**0.961**
7	DRAGON	109.5M	0.251	0.300	0.366	0.314	0.382	0.461
Hybrid								
8	SPLADE	109.5M	0.508	0.589	0.678	0.627	0.735	0.843

Table 2: Retrieval scores of sparse, dense, and hybrid retrievers using only the text from the article of each document in the knowledge corpus consisting of 273 laws.

		Fluency		Correctness				Citation quality		
		COH	FLU	ROU	MET	CON	REL	R	P	Hit
GPT										
1	GPT-3.5-turbo-0125	0.807 ±0.8	0.974 ±0.1	0.561 ±0.7	0.732 ±0.3	0.943 ±0.4	0.964 ±0.2	0.510 ±0.0	0.615 ±1.1	0.784 ±0.0
2	GPT-4o	0.847 ±0.7	0.970 ±0.2	0.629 ±0.4	0.754 ±0.4	0.934 ±0.3	0.961 ±0.1	0.539 ±0.3	0.692 ±0.5	0.830 ±0.5
Open source LLMs										
3	GEITje-7B-ultra	0.794 ±2.8	0.952 ±1.3	0.382 ±1.8	0.369 ±1.7	0.822 ±2.3	0.856 ±1.7	0.146 ±2.3	0.189 ±1.4	0.225 ±3.6
4	Llama-3-8B-dutch	0.744 ±3.1	0.957 ±1.1	0.341 ±2.0	0.427 ±1.9	0.632 ±3.4	0.728 ±3.4	0.237 ±2.5	0.274 ±1.7	0.365 ±3.9

Table 3: Performances of the LLMs on our dataset with the knowledge corpus using the best performing retriever, $mE5_{large}$, with $k = 3$. We show the mean and standard deviation scaled by a factor of 100. The performances are evaluated on Fluency, Correctness and Citation through nine evaluation metrics: *G-EVAL Coherence* (COH), *G-EVAL Fluency* (FLU), *ROUGE-L* (ROU), *METEOR* (MET), *G-EVAL Consistency* (CON), *G-EVAL Relevance* (REL), *Precision* (P), *Recall* (R), and *Hitrate@5* (Hit)

2 and 7 billion parameters. It is well established that larger model sizes often result in better performance due to a better natural language understanding and ability to handle larger context windows. Additionally, we noted that GEITJE produced the most fluent responses, while LLAMA generated the most correct answers and maintained the highest citation quality across all settings. We hypothesize that GEITJE is more proficient in Dutch, while LLAMA is better in understanding instructions.

8 Conclusion

In this paper, we create and evaluate a retrieval augmented generation (RAG) pipeline for attributed Question Answering for the Dutch law, generating long-form answers to precondition questions. We experiment with several state-of-the-art retrievers and generators. For evaluating our pipeline, we create and release a dataset containing 102 legal QA pairs with attributions, as well as an automated evaluation framework suited to this task. The results show that our RAG system can generate fluent and largely correct answers with an 83.0% hit-rate.

Future work includes an extension of the dataset with other document types such as jurisdictions, include more retrievers such as a multilingual hybrid retriever and compare the results from the evaluation framework with human judgements.

Limitations

One of the limitations of this work is that, although we validated the answers with a domain expert, it is not validated whether the answers are indeed understandable to laypeople, or whether multiple experts agree with each other. Another limitation is that the legislative provisions selected often include conditional phrases. This raises the question of whether the retrieval approach may have been inadvertently biased towards these specific linguistic patterns.

References

Samuel Joseph Amouyal, Ohad Rubin, Ori Yoran, Tomer Wolfson, Jonathan Herzig, and Jonathan Berant. 2022. Qampari: An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *ArXiv, abs/2205.12665*.

Nigel J Balmer, Alexy Buck, Ash Patel, Catrina Denvir, and Pascoe Pleasence. 2010. Knowledge, capability and the experience of rights problems. *London: PLEnet*.

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.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Andong Chen, Feng Yao, Xinyan Zhao, Yating Zhang, Changlong Sun, Yun Liu, and Weixing Shen. 2023. Equals: A real-world dataset for legal question answering via reading chinese laws. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 71–80.

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

Phong-Khac Do, Huy-Tien Nguyen, Chien-Xuan Tran, Minh-Tien Nguyen, and Minh-Le Nguyen. 2017. Legal question answering using ranking svm and deep convolutional neural network. *arXiv preprint arXiv:1703.05320*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

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*.

MS Hoekstra and G Teeuwen. 2023. Drempels in de toegang tot het recht.

Christoph Hoppe, David Pelkmann, Nico Migenda, Daniel Hötte, and Wolfram Schenck. 2021. Towards intelligent legal advisors for document retrieval and question-answering in german legal documents. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 29–32. IEEE.

Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z. Pan. 2024. Benchmarking large language models in complex question answering attribution using knowledge graphs. *arXiv preprint arXiv:2401.14640*.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*.

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.

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.

Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2023. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. *arXiv preprint arXiv:2310.05634*.

Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2024. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 493–516, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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*.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Interpretable long-form legal question answering with retrieval-augmented large language models. *arXiv preprint arXiv:2309.17050*.

Behrooz Mansouri and Ricardo Campos. 2023. Falqu: Finding answers to legal questions. *arXiv preprint arXiv:2304.05611*.

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

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

Benjamin Muller, John Wieting, Jonathan H Clark, Tom Kwiatkowski, Sebastian Ruder, Livio Soares, Roee Aharoni, Jonathan Herzig, and Xinyi Wang. 2023. Evaluating and modeling attribution for cross-lingual question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 144–157.

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.

Alejandro Ponce, Sarah Chamness Long, Elizabeth Andersen, Camilo Gutierrez Patino, Matthew Harman, Jorge A Morales, Ted Piccone, Natalia Rodriguez Cajamarca, Adriana Stephan, Kirssy Gonzalez, Jennifer VanRiper, Alicia Evangelides, Rachel Martin, Priya Khosla, Lindsey Bock, Erin Campbell, Emily Gray, Amy Gryskiewicz, Ayyub Ibrahim, Leslie Solis, Gabriel Hearn-Desautels, and Francesca Tinucci. 2019. *Global Insights on Access to Justice 2019: Findings from the World Justice Project General Population Poll in 101 Countries*. World Justice Project.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

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.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.

Alessandro Stolfo. 2024. Groundedness in retrieval-augmented long-form generation: An empirical study. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1537–1552.

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*.

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

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024. Effective large language model adaptation for improved grounding and citation generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6237–6251.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635, Singapore. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Jec-qa: a legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9701–9708.

A Example prompt

You will be given a question and a list of 5 documents that are retrieved by BM25. The retrieved documents contain content that are 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: Wanneer wordt een ongewenstheid verklaring opgeheven?

Documents (EXAMPLE OF THE INPUT YOU WILL RECEIVE):
Potential relevant documents: $\{document1\}, \{document2\}, \{...\}, \{documentk\}$

(EXAMPLE OF YOUR OUTPUT - ALWAYS DO IT IN THIS FORMAT, CAPITALIZING THE "ANSWER:"):
ANSWER: De ongewenstheid verklaring wordt opgeheven bij de volgende condities: Indien de vreemdeling tien jaar onafgebroken buiten Nederland verblijf heeft gehad en de vreemdeling niet ongewenst is verklaard.

(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:

Question: $\{question\}$
Potential relevant documents: $\{document1\}, \{document2\}, \{...\}, \{documentk\}$

Table 4: Prompt template to generate long-form answers. Blue text is the input data that is different for each question.

B Comparison of Generation outputs

Question: When can the court terminate a parent's custody?
Human answer: The court can terminate a parent's custody if: a. a minor is growing up in such a way that their development is seriously threatened, and the parent is not capable of taking responsibility for the care and upbringing, as referred to in Article 247, second paragraph, within a time frame that can be deemed acceptable for the person and development of the minor, or b. the parent abuses their custody.
GPT-4o: The court can terminate a parent's custody if a minor is growing up in such a way that their development is seriously threatened and the parent is not capable of taking responsibility for the care and upbringing within an acceptable time frame, or if the parent abuses their custody. Custody can also be terminated if it has been suspended, provided that the mentioned conditions are met.
GPT-3.5: The court can terminate a parent's custody if a minor is growing up in such a way that their development is seriously threatened, and the parent is not able to take responsibility for the care and upbringing within a time frame that can be deemed acceptable for the person and development of the minor, or if the parent abuses their custody.
GEITje-7B-ultra: The court can terminate a parent's custody if: a. a minor is growing up in such a way that their development is seriously threatened, and the parent is not capable of taking responsibility for the care and upbringing, as referred to in Article 247, second paragraph, within a time frame that can be deemed acceptable for the person and development of the minor, or if the parent abuses their custody. The custody of the parent can also be terminated if the custody has been suspended, provided that the conditions of the first paragraph are met. b. If both parents jointly exercise custody, after the termination of one parent's custody, the other parent will henceforth exercise custody alone. In the case of the termination of custody of a parent who exercises custody alone, the other parent can at any time request the court to be granted custody. This request will be granted if the court deems it necessary in the interest of the minor. The court that has rejected the request referred to in the previous paragraph can always change this decision. However, it will only do so at the request of the parent involved, and only based on circumstances that the court could not have taken into account when making the decision. c. The parent whose custody has been terminated can, at their request, be reinstated in custody if: the reinstatement of custody is in the best interest of the minor, and the parent is capable of permanently taking responsibility for the care and upbringing of the minor, as referred to in Article 247, second paragraph. If, at the time of termination of custody, custody has been assigned to the other parent, the court will not assign custody to the parent whose custody was terminated, and who makes the request referred to in the first paragraph, unless circumstances have changed after the decision assigning custody to the other parent, or the decision was based on incorrect or incomplete information. Article 253e applies correspondingly.
Llama-3-8B-dutch: The court can terminate a parent's custody when a minor is growing up in such a way that their development is seriously threatened and the parent is not capable of taking responsibility for the care and upbringing within a time frame that can be deemed acceptable for the person and development of the minor, or if the parent abuses their custody
Fietje-2B: Royal authority can be terminated if the king has not reached the age of eighteen.

Table 5: Translated examples of long-form answers from various generators for the same setup, receiving the same input chunks from the E5 retriever.