



Universiteit
Leiden

TNO innovation
for life

Master Computer Science

Attributed Question Answering for the Dutch Law
using Retrieval augmented Large language models

Name: Felicia Redelaar
Student ID: s1958410
Date: 27/08/2024
Specialisation: Data Science
1st supervisor: Suzan Verberne
2nd supervisor: Maaïke de Boer, Romy van Drie

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Many individuals are likely to face a legal conflict at some point in their lives, but their lack of understanding of how to handle these complicated issues can make them vulnerable. Advances in Natural Language Processing (NLP), Large Language Models (LLM), and techniques like Retrieval-Augmented Generation (RAG) provide new opportunities to bridge this gap in legal understanding by developing legal aid systems to support laymen. This work proposes a modular system designed to generate attributed long-form answers to Dutch conditional law questions. An example of the Attributed Question Answering task involves a tuple comprising a question, an answer, and an attribution. For instance, “When can the court terminate the guardianship of a natural person?” might be the question, with a corresponding answer like, “The court can terminate guardianship if the guardian abuses authority or lacks required consent,” along with an attribution such as “Book 1 Dutch Civil Code, Article 327.”

Our research primarily explores how effectively we can automatically generate detailed and verifiable answers with attributions to questions in the Dutch legal domain using Retrieval-Augmented Large Language Models. To comprehensively address this, we investigate multiple related questions, such as how different retrieval and generation architectures influence the quality of these answers, the impact of expanding the knowledge corpus on system performance, and the development of an automated evaluation framework suited to this task. To answer these questions, our method employs a RAG pipeline. Our experiments compare multiple retrievers extensively, such as BM25 (sparse), DRAGON (dense), and SPLADE (hybrid). We test various LLMs to generate answers, including GPT-3.5, GPT-4o, GEITje-7B-Ultra, and Llama-3. To support this approach, we introduce and release a dataset containing over 100 legal question-and-answer pairs in Dutch, verified by a legal expert. Our experiments show promising results on automatic evaluation metrics based off the Automatic LLMs’ Citation Evaluation (ALCE) Framework, indicating the potential of this approach to aid individuals navigating legal issues. We publicly release our code and dataset.

Acknowledgements

First of all, I would like to express my sincere appreciation and admiration towards my supervisors Prof. Dr. Suzan Verberne, Dr. Maaïke de Boer, and Drs. Romy van Drie for giving me the opportunity to perform this research and for providing me support, guidance and many learning opportunities on both a professional and personal level.

Suzan, I want to express my gratitude for your belief in me and your guidance throughout the entire thesis process. Before starting my thesis, and even before I started my master's program, I hoped that you would agree to be my supervisor. I was overjoyed when you accepted, and the process has been even better than I had imagined. I had a wonderful time when working on my thesis thanks to your constant inspiration and guidance, coupled with your empathy. It means a great deal to me. I am incredibly grateful.

Maaïke and Romy, thank you for all of your guidance and for giving me the opportunity to work with the NLP team in the Data Science department of TNO. From the first day on I felt very welcomed and I always feel inspired and energized during our chats in the office. Thank you for all your guidance, critical thinking, and in believing in me. I sincerely hope to work with you again in the foreseeable future.

To Valérie, thank you for always being there for me and believing in me throughout my academic journey. Eventhough I met quite some difficulties along the way, your love made me able to pull through and show my true potential. I appreciate everything you do for me, and I am deeply grateful.

Finally, I want to thank my family and especially my mother from the bottom of my heart. Without your support I wouldn't have been able to get this far. As a first-generation student in my family, attending university has always seemed daunting. With your love and support, I was able to face this challenge. Thank you for always standing by me, cheering me on, and believing in the best of me.

Contents

1	Introduction	6
2	Related work	9
2.1	Advancements in AI for the Dutch legal domain	9
2.2	Legal question answering	9
2.3	Attributed question answering	10
2.4	Retrieval augmented generation	12
3	Datasets	14
3.1	Custom-created legal dataset	14
3.1.1	Creating question-answer pairs	14
3.1.2	Knowledge corpus	16
3.1.3	Attributions to knowledge corpus	16
3.1.4	Larger knowledge corpus	17
3.2	NLGen dataset	17
4	Methods	21
4.1	Retrievers	22
4.1.1	Sparse retrievers	22
4.1.2	Dense retrievers	23
4.1.3	Hybrid retrievers	25
4.2	Generator for long-form answers	26
4.3	NLGen experiment	27
5	Experiments and results	29
5.1	Evaluation metrics	29
5.1.1	Retriever evaluation	29
5.1.2	Long-form answer evaluation	29
5.2	Experimental setup	33
5.3	Results	34
5.3.1	Retriever component	34
5.3.2	Generator component	37
5.3.3	NLGen dataset	40
6	Discussion	41
7	Conclusion	43
A	Example prompt and output	53
B	G-EVAL prompts	55

1 Introduction

Legal issues are part of everyday life and can have a significant impact on individuals, ranging from minor disputes with neighbors to major challenges in areas such as employment, housing, and family life (Farrow et al., 2016; Ponce et al., 2019). However, many people lack knowledge about their legal rights when they encounter civil justice problems, leading to difficulties in obtaining advice and taking necessary actions (Balmer et al., 2010). Research has identified financial costs and a lack of awareness regarding legal rights and options as the main barriers to accessing justice (Hoekstra and Teeuwen, 2023).

The global implications of this issue are significant. An estimated 1.4 billion people have unmet civil justice needs (Garavano et al., 2019), and 43% of those surveyed in the research by Ponce et al. (2019) reported that their justice problem had a negative impact on their lives. Due to limited access to professional legal advice, many people turn to the internet for information. While the internet can improve public understanding of the law, it is not a perfect tool for public legal education (Denvir, 2016). Experts agree that access to public legal information is highly beneficial in improving legal dispute outcomes (Currie, 2009). In this context, the emergence of artificial intelligence (AI) presents a promising development. AI can significantly revolutionize the accessibility, affordability, and efficiency of legal information, potentially redefining the process of accessing justice altogether (Marwala and Mpedi, 2024).

In recent years, the rapid progress in AI, particularly in natural language processing (NLP), has enhanced the potential to bridge the gap between people and the law. These advancements have resulted in the development of advanced tools and applications that aim to make legal information more accessible. For instance, legal text summarization (Sheik and Nirmala, 2021; Anand and Wagh, 2022) has the potential to simplify complex legal documents for non-experts. Additionally, legal judgement prediction (Trautmann et al., 2022; Long et al., 2019) can reveal valuable parallels between an individual's circumstances and the likely legal outcome. Moreover, researchers focused on information retrieval tasks in the legal domain for expert finding (Askari et al., 2022) and answer retrieval in legal community questions (Askari et al., 2024). Lastly, legal question answering could provide affordable assistance to a wider audience. Various initiatives have been undertaken to assist individuals by creating legal chatbots for various languages, including French, Thai, and Indonesian, respectively (Queudot et al., 2020; Socratianurak et al., 2021; Firdaus et al., 2020).

However, there are still significant gaps in current legal question answering solutions. One major issue is that many existing systems do not provide verifiable

sources in their responses, which limits the verifiability of the answer and affects the trustworthiness of the information. Additionally, the answers generated by these systems are often too general and may not be tailored to an individual's specific legal situation or provide detailed information about their legal options. Furthermore, the responses are often too brief, such as simple "yes" or "no" answers (Do et al., 2017), which fail to capture the complexity of legal issues. Another concern is that many countries still lack a digital legal aid system, as each country operates under its own legal framework.

Based on the above, we see an opportunity for legal Question Answering (QA) with attribution customized for the Dutch law. The Dutch law is especially suitable since all Dutch laws are digitized and well-organized. In this paper, we present a modular approach that is aimed to generate informative long-form answers to Dutch law questions. In order to increase the verifiability of our system, each generated answer contains a list of used sources. Our method uses a Retrieval Augmented Generation (RAG) pipeline. This works in the following way: for a given question, a retriever firstly fetches relevant law articles over a large knowledge corpus. Then, a Large Language Model (LLM) is used as a generator to formulate a long-form answer using these retrieved law articles. After evaluating multiple retrievers for the task, we use the MULTILINGUAL-E5 text embedding model (Wang et al., 2024) as the retriever in our system, which is based off the English E5 variant (Wang et al., 2022). Furthermore, we use an instruction-tuned LLM that we adapt to our task through in-context learning. To support the evaluation of our system, we create and release a dataset containing 102 Dutch legal question and answer pairs with attributions to the law articles that have been directly used for the answer. Each answer is extensive and has been verified by a legal expert on legal correctness. Lastly, we extend and adjust an automatic evaluation framework for the attributed question answering (AQA) task.

In this thesis, we compare different methods for automatically producing long-form answers with attributions to Dutch legal questions. We will do this using Retrieval-Augmented LLMs and experimenting with various retrievers and LLMs as generators to find out what architecture results in the best performance for this specific task. We experiment with two use cases. The first use case is that the user has selected relevant laws beforehand for the RAG system to work with. The second use case is when the user is unaware of which laws might be potentially relevant. Lastly, we aim to establish the most suitable evaluation framework for this task that can be completely automated, ensuring it effectively measures the performance of different retrieval-augmented LLM configurations. We conduct an experiment over a benchmark dataset to find out whether the evaluation framework can be generalized.

Based on the above, we address the following four research questions:

- RQ1: With what quality can we automatically generate detailed and verifiable answers with attributions to questions in the Dutch legal domain using Retrieval-Augmented Large Language Models?
- RQ2: In what ways do different retrieval and generator architectures influence the quality and verifiability of long-form answers in Dutch legal question answering?
- RQ3: What are the effects on the performance of our RAG system when significantly enlarging the knowledge corpus?
- RQ4: What is the most suitable evaluation method for the attributed question answering task, and to what extent can it be generalized over a benchmark dataset?

Our main contributions are summarized below:

- We propose a modular system that allows users to select and evaluate various retrievers and generators, producing answers with attributions to the Dutch law. We demonstrate the effectiveness of our system by conducting extensive experiments and evaluations that highlight the model's ability to generate accurate and verifiable responses with proper attributions.
- We have created an annotated dataset for (attributed) question answering for the Dutch law which 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. Each answer has been verified by a legal expert on legal correctness.
- We extend an existing method for the automatic evaluation of the attributed question answering task. Our extended evaluation method is especially suitable when the answers in the dataset contain ground-truth attributions.

2 Related work

In this chapter, we will describe the relevant literature. We firstly discuss AI applications for the Dutch legal domain in Section 2.1, followed by legal question answering in Section 2.2. Then we dive into the attributed question answering task and the task’s evaluation methods in Section 2.3. We end the chapter discussing RAG in Section 2.4.

2.1 Advancements in AI for the Dutch legal domain

There is an active research community for AI applications in the Dutch legal domain for many tasks. In regards for the question answering task, [van Kuppevelt and van Dijck \(2017\)](#) presented open source software to analyze and visualize networks of Dutch case laws to assist in answering legal research questions. [Rouws et al. \(2022\)](#) published an extractive question answering system based off BERT to answer questions about labour agreements automatically. Furthermore, [Schraagen et al. \(2022\)](#) applied abstractive summarization models on a Dutch legal domain dataset. Furthermore, work has been done that focuses on interpreting Dutch legal texts. [van Drie et al. \(2023\)](#); [Bakker et al. \(2022a,b\)](#) annotated 55 different Dutch laws with roles for semantic role labeling. Another work with the goal of interpreting Dutch legal texts was by [van Doesburg et al. \(2016\)](#), who proposed FLINT. FLINT allows for high-level descriptions of normative systems so legal experts can agree on law interpretations. FLINT’s use case has been shown by interpreting the Dutch Aliens Act ([van Doesburg and van Engers, 2019](#)).

2.2 Legal question answering

Legal QA is a task that involves receiving a query and producing an answer to this query automatically ([Martinez-Gil, 2023](#)). These questions are commonly binary, multiple-choice, multi-span, or long-form. A traditional method used by researchers is to use a two-stage retrieval system. In the first stage, a retriever is used to find relevant content. In the second stage, a model is used to generate answers. Previous older works, for example, have used a convolutional neural network (CNN) to answer “yes” or “no” to Japanese legal questions ([Kim et al., 2017](#); [Do et al., 2017](#)).

The rapid advancements in the field of NLP have led to significant improvements in Legal QA. Traditionally, the first stage of Legal QA relied on sparse retrieval techniques. Sparse retrieval is keyword-based and is utilized by the bag-of-words or TF-IDF methods. With the rise of transformers, several works such as ([Hoppe et al., 2021](#); [Khazaeli et al., 2021](#)) have incorporated BERT ([Devlin et al., 2018](#)) as a dense retriever to improve the first stage in their Legal QA system. Dense

retrievers embed both the query and the document as a vector in a continuous vector space, which allows to capture the semantic meaning of words.

The most recent advancements in the field are regarding the second stage in the Legal QA system. These advancements can be attributed to the rise of LLMs such as GPT-3 (Brown et al., 2020) and open source LLMs such as BLOOM (Le Scao et al., 2022). A notable contribution to Legal QA comes from (Louis et al., 2023). This work employs the conventional two-stage method to answer long-form legal questions. What makes the work novel however, is the use of RAG with an LLM to generate answers. RAG is a technique that will be elaborated upon in Section 2.4. Finally, their methodology involves generating attributed answers that include a pointer towards a knowledge base.

In regards to the data for the task of Legal QA, multiple works have released datasets. Zhong et al. (2020) release JEC-QA, a Chinese dataset for multiple-choice questions. This dataset is sourced from legal exams. For long-form questions, Mansouri and Campos (2023); Chen et al. (2023), and Louis et al. (2023) released English, Chinese, and French datasets, respectively. The first two datasets were sourced from online forums. The last mentioned dataset was sourced through lawyers.

2.3 Attributed question answering

Attributed question answering (AQA) is a formal task formulated by (Bohnet et al., 2022). In this task, the input is a question, and the output is a tuple of an answer string and its attributions. The attributions are references to a knowledge corpus \mathcal{C} . An example for the AQA task is an 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”, and attribution pointers output: [1][2]. These pointers are directed towards text segments within a knowledge corpus that support the given answer string.

Multiple studies have suggested methods for evaluating the answers and attributions generated by an LLM. Some studies involve manual human evaluation. Menick et al. (2022) for example, introduced the S&P score, which assesses whether the answer is supported by the given attributions and whether the answer itself is plausible. Furthermore, Kamaloo et al. (2023) introduced HAGRID, which is similar to the S&P score, but uses informative and attributive scores. HAGRID measures whether the explanation directly answers the question and whether the explanation is attributable to the attributions.

Moving towards automatic evaluation systems, several studies have made progress in prompting LLMs to generate evaluations. For instance, 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. (2023a) 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), a framework that utilizes 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. Table 1 provides an overview of ALCE’s evaluation’s dimensions, definitions, and metrics.

Evaluation dimension	Definition	Metrics
1). Fluency	Whether the model’s generated text is fluent and coherent	<ul style="list-style-type: none"> • MAUVE
2). Correctness	Whether the answer is accurate and covers all aspects of interest	<ul style="list-style-type: none"> • Exact match recall • Recall-5 • Claim recall
3). Citation Quality	Whether the answer is well supported by the cited passages and no irrelevant passages are cited	<ul style="list-style-type: none"> • Citation recall • Citation precision

Table 1: Evaluation dimensions of ALCE.

The automatic evaluation for ALCE consists of three dimensions: fluency, correctness, and citation quality. Firstly, the fluency metric measures whether the generated text is fluent and coherent. This metric is measured using MAUVE (Pillutla et al., 2021). MAUVE is a comparison measure to evaluate how semantically close a model’s generated text is to that of a human-written text using divergence frontiers. Secondly, the correctness metrics measure whether the answer is accurate and covers all aspects of interest. For each dataset, Gao et al. (2023) uses a different metric, resulting in three metrics. These metrics are *exact match recall*, *recall-5*, and *claim recall*. Claim recall creates three “sub-claims”

from the ground truth answers using a GPT model. Then, TRUE (Honovich et al., 2022), which is a T5 (Raffel et al., 2020) model fine-tuned on natural language inference (NLI) datasets, is used to check whether the model output entails the sub-claims. Lastly, the citation quality measures whether the answer is well supported by the cited passages and whether no irrelevant passages are cited. This is measured by a T5 model that outputs “1” if the output is entirely supported by cited passages. The metrics used for the citation quality dimension are calculated using precision and recall. The scores are averaged over all Q&A pairs in the dataset. In our work, we will adjust and extend upon the ALCE evaluation framework to evaluate our method.

Multiple other works have used the ALCE framework to benchmark their models for the AQA task. Some notable works using the ALCE framework include Li et al. (2023b), who introduced LLATRIEVAL for AQA using RAG. Their main aim is to improve the retriever by prompting an LLM and determining if the retrieved documents are sufficiently informative to form an answer. Another work using ALCE is by Sun et al. (2023), who introduced VTG. This approach incorporates an evolving memory and self-reflection. VTG employs evolving long short-term memory to retain valuable and recently retrieved documents. Additionally, Huang et al. (2024) takes a different approach by utilizing reinforcement learning and optimizing the policy using proximal policy optimization. Finally, Asai et al. (2023) introduced SELF-RAG, which enhances answers using self-reflection.

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 question answering (AQA) as it allows external knowledge to be used to answer questions, while the attributions can be generated by the LLM that is used in the RAG system. 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.

Before the introduction of RAG, significant research had already been dedicated to optimizing retrievers for question answering (QA) tasks. These early efforts focused on enhancing the retrieval step, which was used to create an answer

through extractive techniques or text generation. It is important to note that in RAG systems, virtually any existing retriever can be utilized. Notable examples of this pre-RAG work include [Chen et al. \(2017\)](#), who proposed DRQA which embeds both questions and paragraphs in order to find the most relevant text spans in paragraphs. [Karpukhin et al. \(2020\)](#) proposed Dense Passage Retrieval and showed that a dense retriever can outperform sparse vector space models such as BM25 when adding enough data. After the introduction of RAG, [Formal et al. \(2021\)](#) introduced SPLADE, which is a retriever that combines dense and sparse retrieval. [Lin et al. \(2023\)](#) introduced DRAGON, which is a generalized dense retriever trained through progressive data augmentation. The training sampled difficult negatives which result in high qualitative dense representations. [Ram et al. \(2023\)](#) proposed RALM to optimize the retriever. Their method implements a ranking function that can be back-propagated into.

Besides optimizing the retriever, work has been done to optimize the system as a whole. The paper written by [Lewis et al. \(2020\)](#) that introduced RAG, fine-tuned the system end-to-end in such a way that the retriever and the generator learned jointly. [Izacard and Grave \(2020\)](#) proposed FID, which first encodes each retrieved document independently using a T5 encoder and then performing attention over all the output representations using the decoder to generate the final answer. [Borgeaud et al. \(2022\)](#) introduced RETRO and demonstrated that models augmented by retrieving information from large databases outperform larger models trained on larger datasets. The architecture in their work was 25× more efficient because it could rely on an external memory. [Guu et al. \(2020\)](#) experimented with updating the whole system including the document encoder, making it a costly method. [Izacard et al. \(2022\)](#), however, showed that updating the query in many cases is good enough. This therefore makes updating the document encoder unnecessary. [Izacard et al. \(2022\)](#) also proved the power of RAG systems by showing that substantial improvements can be seen when comparing a closed-book model to a retrieval augmented model.

3 Datasets

In this chapter, we describe the datasets used in our research, which include a custom-created legal dataset and a benchmark dataset. The custom dataset involves two main stages: creating question-answer pairs and developing a knowledge corpus. First, we generated legal questions and annotated them with corresponding answers. Next, we created an expansive corpus of law articles drawn from Dutch legislation. This corpus was then used to enrich the question-answer dataset with attributions to specific legal sources.

In addition to our custom legal dataset, we also incorporated the NLGen dataset. This is a benchmark dataset used in natural language generation tasks. Unlike our custom dataset, which focuses on legal work, the NLGen dataset is not specifically related to legal content.

We elaborate upon each of these datasets in the sections below. We elaborate our custom-created legal dataset in Section 3.1, and briefly discuss the NLGen dataset in Section 3.2.

3.1 Custom-created legal dataset

3.1.1 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. These types of questions are particularly useful because they clarify the specific requirements that must be met before a legal action can be undertaken. According to the Cambridge Dictionary, a precondition is “something that must happen or be true before it is possible for something else to happen.”¹ To make the definition more formal, 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 the following: “When can a student be eligible for student financing?” and “What are the requirements for entering into a marriage?”. The answer to the first question contains pre-conditions such as having a Dutch nationality. The latter question can be answered with the pre-conditions of being at least 18 years old.

We create the questions by carefully reviewing each Dutch law via the official government website² and reading these on an article level. We look for subordinating conjunctions such as “only if” or “on condition (that)”. Whenever we find

¹<https://dictionary.cambridge.org/dictionary/english/precondition>

²<https://wetten.overheid.nl/>

such conjunctions followed by actionable measures in a law text, we formulate a legal question. We answer 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.

In order to create the questions, we must first select laws from which to sample from. We choose the laws by first filtering out all technical or administrative legislation. These are often laws that are intended to adjust, implement, or execute existing legislation without making policy changes. 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). We finally select 25 laws at random for question sampling.

In total, we review 25 Dutch laws, comprising a total of 4441 articles. We were not able to sample from 8 out of 25 Dutch laws due to missing pre-conditions in the text. An overview of the laws we have sampled from can be found in Table 3. In total, we create 110 questions and answer pairs that are suitable for retrieval-based question answering. Next, a legal expert is consulted for quality assurance, checking the legal correctness of all the questions and answers. We implement their feedback regarding question and answer pairs, and discarded questions that were too vague or contained answers that were too complex to verify for legal correctness. Finally, a legal expert is consulted again to make sure the dataset quality is up to par. The legal expert was consulted on the legal correctness and completeness of the questions and answers. This yields a final number of 102 question-answer pairs, each with legal references. Table 2 shows the statistics of the QA dataset.

Statistic	Value
Number of questions	102
Average question character length	104.29
Average answer character length	830.40
Average number of attributions	1.54

Table 2: Summary of QA dataset statistics.

3.1.2 Knowledge corpus

We use the Dutch law as the knowledge corpus to provide evidence that a system can use when generating an answer. Initially, we created a knowledge corpus containing the Dutch law articles from 17 Dutch laws that we sampled the question-answer pairs from. We start by downloading these publicly available Dutch laws via the official government website.³ The laws are available in Extensible Markup Language (XML), which organizes data hierarchically using nested tags and elements. We develop a parsing script that is capable of processing all Dutch laws. It converts the laws from XML into a CSV file for better data processing. In the CSV file, each row contains the text of a law article. To ensure all relevant information for each article is included, we add metadata information of each article's text on its law's name, BWBID, which is a law ID, its chapter's name, section's name, paragraph's name, subparagraph's name, and article's name. The purpose of adding metadata is two-fold. Firstly, it makes references more accessible for verification. Secondly, more information could lead to stronger embeddings for the retriever component of our work. Table 4 shows all data and metadata for one example chunk.

After parsing the laws from XML files into a structured CSV format and adding metadata for each article, the next step involves the creation of the knowledge corpus. The corpus has been created by merging all CSV files vertically. We split articles longer than 150 words 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. We experimented with a sentence cut-off, but this resulted in problematic results when splitting since some sentences are substantially longer than 150 words. Finally, for each created chunk, we assign a unique document ID to facilitate straightforward referencing by the system. In total, we have 4,653 chunks created from 3,241 articles using 17 laws. Most articles (2,433) contain one chunk since their text consists of 150 words or less. The remaining 808 articles are comprised of 2 to 19 chunks. These chunks are the retrieval unit for our retrieval component.

3.1.3 Attributions to knowledge corpus

After establishing the question and answer set and knowledge corpus, we manually place attributions for each question-answer pair to relevant document chunks in the knowledge corpus. By design, 100% of our dataset's collected questions-and-answer pairs contain article-level references. This meticulous approach ensures that each answer can be traced back to a specific legal article, enhancing the

³<https://wetten.overheid.nl/>

reliability and traceability of the dataset. By providing detailed references, we aim to build a robust foundation for a system capable of delivering precise and legally sound responses with verifiable answers. An example of a created attributed question-answer pair can be seen in Table 5.

3.1.4 Larger knowledge corpus

In addition to the original knowledge base that was created using 17 Dutch laws, we have broadened our dataset to include a wider range of legislation. This expansion is necessary for two reasons. Firstly, we want to investigate the robustness of our RAG components and pipeline when the knowledge corpus is substantially larger. Secondly, a smaller and larger knowledge corpus reflect the application in the real world use. For some use cases, the user might be aware of some specific laws to give as input while in other cases the user might not know yet which laws are relevant for its question. To accomplish this, we have gathered all the remaining Dutch laws that were initially excluded based on our criteria described in Section 3.1.1, resulting in a total of 273 laws.

Following the same methodology used for the smaller knowledge corpus, we downloaded these additional laws from the official government website⁴ in XML format. We then applied our parsing script to convert these XML files into CSV format. We maintained consistency by splitting law articles longer than 150 words into multiple chunks and assigning unique Document IDs. The larger knowledge corpus increased the number of articles from 3,241 to 22,462 articles. Furthermore, this increased the chunks from 4,653 to 30,803. Notably, several newly included laws are relatively short, comprising only a few chunks each. Most articles (16,665) contain one chunk since their text consists of 150 words or less. The remaining (14,138) articles are comprised of 2 to 20 chunks.

Finally, we update the attribution pointers for our question-answer set to align with this larger corpus. Using a Python script, each question-answer pair is linked to the relevant document chunks in the expanded knowledge corpus.

3.2 NLGen dataset

The NLGen dataset is designed for tasks involving question answering and natural language generation. The dataset is composed of 153,725 examples in the training set, which are used to train models. Additionally, it includes 12,467 examples in the validation set, which serve to fine-tune the models and assess their performance during the training process.

⁴<https://wetten.overheid.nl/>

Law	Questions	(%)
Bankwet	0	0.00
Burgerlijk Wetboek Boek 1	20	19.61
Elektriciteitswet 1998	4	3.92
Embryowet	5	4.90
Gaswet	0	0.00
Geneesmiddelenwet	6	5.88
Grondwet	1	0.98
Hamsterweg	0	0.00
Jeugdwet	8	7.84
Mediawet	0	0.00
Opiumwet	4	3.92
Participatiewet	12	11.76
Politiewet	0	0.00
TNO-wet	0	0.00
Vreemdelingenwet 2000	22	21.57
Werkloosheidswet	2	1.96
Wet geluidshinder	0	0.00
Wet goed verhuurderschap	0	0.00
Wet Kinderopvang	1	0.98
Wet op de huurtoeslag	0	0.00
Wet op het primaire onderwijs	5	4.90
Wet openbaarheid van bestuur	2	1.96
Wet ruimtevaartactiviteiten	2	1.96
Wet studiefinanciering 2000	6	5.88
Zorgverzekeringswet	2	1.96
Total	102	-

Table 3: Question distribution over laws.

The NLGen dataset is diverse in its content, covering a wide range of topics and question types. The data has been collected using real questions sampled from anonymized Bing queries, with all questions and answers presented in English. Each example in the dataset consists of five variables: a `query_id`, a query, a list of 10 passages (with annotations indicating whether they were used to formulate the answer), the answer provided by a human annotator, and finally, a `query_type`. The `query_type` specifies the kind of information being sought, such as a location, number, or person. Notably, for each question, only one out of the ten listed passages is used by the human annotator to generate the answer.

Feature	Document 3846	Document 3847
DOC_ID	DOC3846	DOC3846
law_id	BWBR0003420	BWBR0003420
law_name	Wet op het primair onderwijs	Primary Education Act
hoofdstuk	Hoofdstuk I	Chapter I
hoofdstuk_titel	Basisonderwijs	Primary education
afdeling	Titeldeel IV	Title Part IV
afdeling_titel	Onderwijsachterstandenbeleid	Educational disadvantage policy
paragraaf	Afdeling 9	Department 9
paragraaf_titel		
subparagraaf_titel	Bekostiging	Funding
titel_titel	Artikel 160	Article 160
artikel	Artikel 160	Article 160
article_name	Afspraken voor- en vroegschoolse educatie	Agreements on pre-school and early childhood education
text	Totstandkoming van de afspraken. een houder van een kindercentrum verwerkt persoonsgegevens met als doel de bevordering van een goede doorstroom naar het basisonderwijs. De gegevens die een houder van een kindercentrum verplicht overdraagt aan het bevoegd gezag van een basisschool over leerlingen die op de basisschool zijn ingeschreven, zijn [...]	Establishment of the agreements. A childcare centre owner processes personal data with the aim of promoting a smooth transition to primary education. The data that a childcare centre owner is required to transfer to the competent authority of a primary school about pupils who are enrolled in the primary school are [...]

Table 4: An example chunk from the created knowledge corpus. The chunk can be seen translated and untranslated.

Question	Ground truth answer	Attributions
When is an employee not entitled to unemployment benefits?	<p>An employee who:</p> <ul style="list-style-type: none"> a. receives a benefit under the Sickness Benefits Act or a benefit that corresponds to it in nature and scope is not entitled to a benefit; b. receives a disability benefit or a wage-related benefit from the return to work benefit for partially disabled persons under the Work and Income (Capacity for Work) Act; c. receives a benefit under the Disability Insurance Act, calculated based on a disability of at least 80%, or receives a benefit that corresponds to that benefit in nature and scope; d. receives a benefit under Chapter III of the Military Disability Benefits Act, calculated based on a disability of at least 80%, or an allowance under that chapter, which, whether or not increased by the disability benefit, amounts to 70% or more of the daily wage on which the disability benefit has been or would have been calculated; e. lives or stays outside the Netherlands other than on holiday; f. is not lawfully resident in the Netherlands as referred to in Article 8 of the Aliens Act 2000; g. has been legally deprived of his freedom; h. evades the execution of a custodial sentence or custodial measure; i. has reached retirement age; j. has an exemption on the basis of Article 64 of the Social Insurance Financing Act due to conscientious objections or whose unemployment started within 3 months after the date of withdrawal of such an exemption; k. takes a holiday outside the period to be determined by ministerial regulation, referred to in the ninth paragraph, part b; l. is unemployed as a result of a strike or lockout, unless the employee has been granted an exemption on the basis of Article 8, paragraph 3, of the Extraordinary Labor Relations Decree 1945; m. receives a benefit under the Work and Care Act; n. is a traveller. 	DOC2743, DOC2744

Table 5: Translated example of attributed question-answer pair showing the question on the left, the ground truth answer formulated by a human in the center, and the attributions towards a knowledge corpus on the right side.

4 Methods

The goal of our work is to provide a modular system that generates long-form answers to legal questions through a potentially large collection of law documents by using a instruction-tuned LLM as backbone. The overview of the system is presented in Figure 1. Here, we can see the components of our RAG pipeline. The first component is the *retriever*, which aims to find relevant law articles. Based off a question, the retriever selects a small subset of law articles with the highest similarity score from the knowledge corpus. Then, an LLM is used as the *generator* to generate an answer based on the subset of articles returned by the retriever. We describe the retriever and generator components and their evaluation methods in more details below.

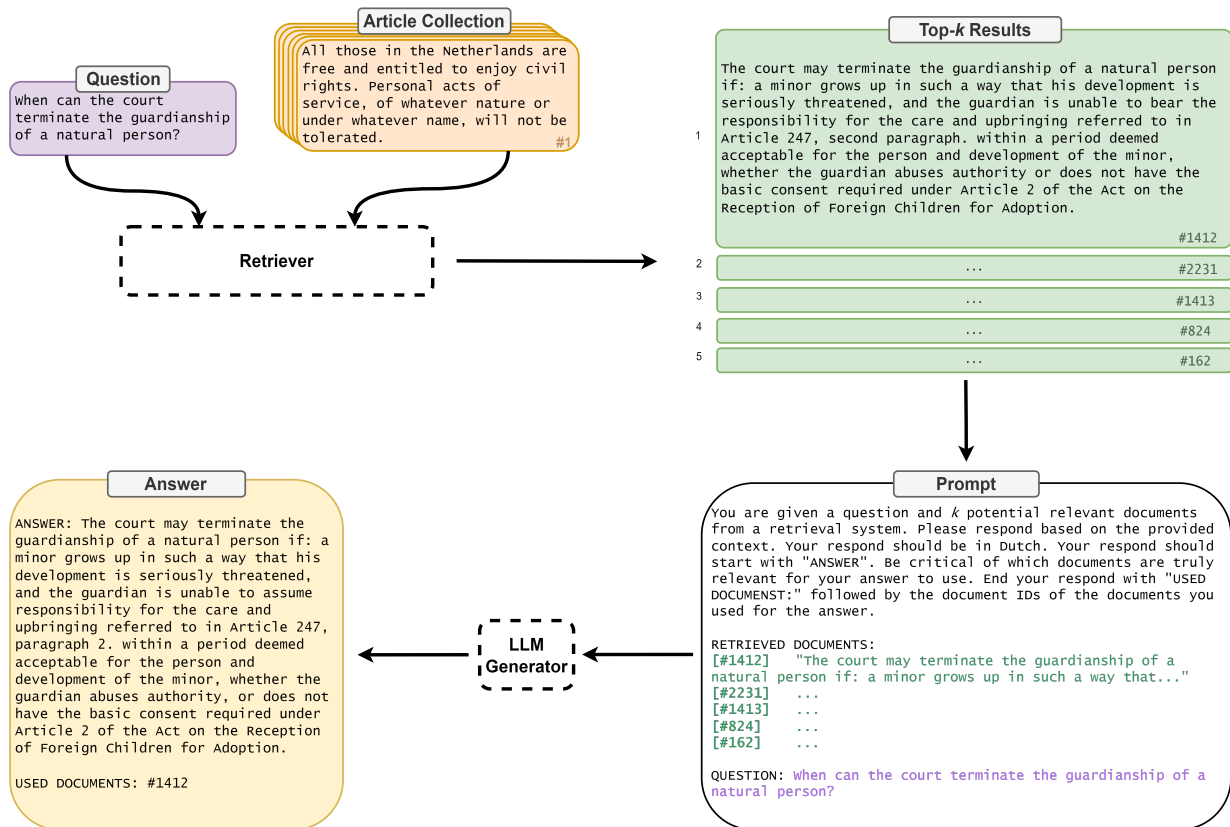


Figure 1: Pipeline overview using the RAG framework to automatically generate long-form answer for Dutch legal questions using a large collection of law documents, such as publicly available law articles.

Our retrieval component finds a small subset of relevant law articles based on a question to answer a legal question. This step is crucial as it significantly reduces the search space, allowing the generator in our system to focus on the most relevant information. More formally, let $\mathcal{D} = \{d_1, d_2, \dots, d_N\}$ represent the entire corpus of documents consisting of law articles and let q represent a query which is a legal question. The retriever function can be expressed as a function R that maps a query and the document corpus to a small retrieved subset of relevant documents $\mathcal{D}' \subset \mathcal{D}$:

$$R : (q, \mathcal{D}) \mapsto \mathcal{D}'$$

The retrieved subset \mathcal{D}' contains documents that are ordered by decreasing relevance. For a given query $q \in \mathcal{Q}$, the function $R(q, \mathcal{D})$ yields the set of documents:

$$\mathcal{D}' = \{d_1, d_2, \dots, d_k \mid d_i \in \mathcal{D}, \text{ and } \text{rank}(d_i) \leq k\}$$

Here, $\text{rank}(d_i)$ represents the rank of document d_i in terms of relevance to query q . The variable k denotes the number of documents we retrieve as potentially relevant and for the generator to use. Specifically, these are the top- k documents with the highest relevance scores.

4.1 Retrievers

In this work, we compare multiple retrievers to find the best performance for our system. Each retriever adheres to the formal notation expressed above. We evaluate sparse, dense, and hybrid retrievers. Each approach has unique characteristics and advantages, which we discuss in the following subsections.

4.1.1 Sparse retrievers

Sparse retrievers focus on the lexical overlap of terms between the query and the documents. Sparse retrievers rely on traditional information retrieval methods such as term frequency-inverse document frequency (TF-IDF) (Sparck Jones, 1972). The term frequency measures how often a term appears in a document. The more often a term appears, the more relevant the term is considered, but with diminished returns to avoid overemphasis on the term. The inverse document frequency measures how common or rare a term is across the entire collection. Rare terms are given more weight since they are more likely to be significant when distinguishing between relevant and non-relevant documents. More formally,

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

where $f_{t,d}$ is the frequency of term t in document d .

$$\text{IDF}(t, D) = \log \left(\frac{N}{|\{d \in D \text{ and } t \in d\}|} \right),$$

where N is the total number of documents in the collection D , and $|\{d \in D : t \in d\}|$ is the number of documents containing the term t . The TF-IDF score is then computed through Eq. 1.

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \cdot \text{IDF}(t, D) \quad (1)$$

Our work uses the BM25 algorithm (Robertson et al., 1995) as a baseline retriever, which heavily relies on TF-IDF. The BM25 algorithm returns the most relevant documents by calculating a relevance score for each document with respect to the query and returning the top k documents that are based on lexical overlap with term weighting. BM25 also incorporates document normalization, which adjusts for the length of the documents to ensure longer documents are not unfairly favored because they contain more terms than shorter documents. Both the term frequency and the length normalization can be tuned using parameter tuning. The parameter k tunes the influence of the term frequency scaling. A higher value makes the algorithm more sensitive to the count, giving higher scores to documents where the query terms are repeated more often. The parameter b controls the degree of length normalization. A higher value will penalize more lengthy documents harsher to avoid giving them an unfair advantage. For our work, we use the default parameters of $b = 0.75$ and $k = 1.2$.

While sparse retrievers are computationally efficient and interpretable, they do not consider the semantic meaning of words in the query or in the documents. This means sparse retrievers potentially miss relevant documents that do not share exact terms with the query.

4.1.2 Dense retrievers

Dense retrievers do not focus on the exact match of terms between the query and the document like sparse retrievers do. Instead, dense retrievers focus on the semantic meaning of the terms between the query and the document by creating vector representations of words containing real numbers. According to the distributional hypothesis (Harris, 1954), words that tend to occur frequently together are semantically more similar than words that do not occur frequently together. Firth (1957) famously summarised this principle as “a word is characterized by the company it keeps”. This allows for document retrieval for our system, where the similarity between the vector representations of a query and all the documents is found by calculating the cosine similarity. The cosine similarity in our work is calculated using Eq. 2.

$$\text{Cosine Similarity}(A, B) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}} \quad (2)$$

In this formula, A_i and B_i represent the components of vectors A and B respectively. The dimensionality of the vectors, i , ranges from 1 to n .

Historically, dense retrievers struggle with words that contain multiple meanings since static embedding methods represent words as a fixed dense vector. Each word is, therefore, represented as a single representation. The issue is highlighted in sentences such as the following, “He was fascinated by the pitch.” In this example, it is unclear whether the term *pitch* is related to the word sense of a sales presentation, the quality of a sound, or a playing field of a sports game such as baseball. A solution for this problem has been developed by works such as BERT (Devlin et al., 2018) to accommodate for a term’s context. Unlike static embeddings, these embeddings reflect the multi-sense nature of words by dynamically adjusting to the context in which the word is used. This is achieved by leveraging transformer models (Vaswani et al., 2017) that utilize attention mechanisms to encode information from surrounding tokens. Consequently, the representation of a word token is influenced by its contextual tokens, allowing for more precise disambiguation of meanings.

In our work, we compare the following embedding models that we use as our dense retrievers: SBERT, E5, BERT and DRAGON. We will discuss each briefly in the following subsections.

BERT

In our work, we use LEGAL-BERT-DUTCH-ENGLISH, an open-source BERT model that has been finetuned using the BERT multilingual base model (uncased) on Dutch and English legal documents. We use it through Hugging Face.⁵ We have chosen to use this model since it has been trained on over 2000 Dutch legal long-documents, which could make it a powerful embedding model for our domain specific task.

SBERT

SBERT, also known as Sentence-BERT introduced the Siamese network concept to focus on the understanding of the entire sentences as a whole, instead of understanding the context of each word within a sentence. SBERT models are therefore more suitable for tasks such as semantic textual similarity and information retrieval. We use two SBERT models in this work. The first we will discuss is ALLNLI-GRONLP-BERT-BASE-DUTCH-CASED, which we use through Hugging

⁵<https://huggingface.co/Gerwin/legal-bert-dutch-english>

Face.⁶ This model maps sentences and paragraphs to a 768 dimensional dense vector space and can be used for tasks like clustering or semantic search. We choose to use this model since this model specifically has been trained on Dutch data. The second SBERT model that we use for our work is PARAPHRASE-MULTILINGUAL-MINI-LM-L12-V2. We use this model through Hugging Face.⁷ This model has been trained using a paraphrasing mining task to maps sentences and paragraphs to a 384 dimensional dense vector space. We include this model in our experiments since the model is multilingual, and the models pre-training task of finding texts with identical or similar meanings in a large corpus of sentences is similar to our task.

E5

In our work, we use MULTILINGUAL-E5, an open source text embedding model (Wang et al., 2024) through Hugging Face.⁸ The model is an E5 model that has been trained on a mixture of multilingual datasets. The authors have released three embedding models of different sizes (small, base⁹, and large¹⁰). The training of the model adheres to the English E5 model recipe, which involves contrastive pre-training. The contrastive learning method improves the sequence-level representations from pairs of text, making it appropriate for retrieval and text-matching tasks. In this work, we compare all three sizes for the retrieval experiments.

DRAGON

DENSE RETRIEVER TRAINED WITH DIVERSE AUGMENTATION (DRAGON), (Lin et al., 2023) is the last dense embedding model that we use for our work. We use the model through Hugging Face.¹¹ DRAGON is trained using progressive data augmentation. In other words, it samples very difficult negatives which theoretically gives better vector representations. We choose to use this model since it is seen as the state-of-the-art in generalized dense retrievers. A limitation of this model, however, is that this model is trained on English datasets and is not multilingual.

4.1.3 Hybrid retrievers

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

⁶<https://huggingface.co/textgain/allnli-GroNLP-bert-base-dutch-cased>

⁷<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>

⁸<https://huggingface.co/intfloat/multilingual-e5-small>

⁹<https://huggingface.co/intfloat/multilingual-e5-base>

¹⁰<https://huggingface.co/intfloat/multilingual-e5-large>

¹¹<https://huggingface.co/facebook/dragon-plus-context-encoder>

EMBEDDINGS (SPLADE) (Formal et al., 2021) for our retrieval experiments through Hugging Face.¹² We choose this model to experiment with hybrid retrievers. A limitation of this model is similar to DRAGON, that it has been trained on English datasets and is not a multilingual model.

4.2 Generator for long-form answers

After retrieving the most relevant documents, our generator is instructed through a prompt to generate a long-form answer using the potentially relevant documents and the corresponding question. We use one-shot in-context learning, and provide the prompt in the language in which the LLM's has been mostly pre-trained on. An example of the instruction prompt used can be seen in Table 18. An example of the output of an LLM can be seen in Table 19 in Appendix A.

For our generators, we choose to experiment with four GPT models by OpenAI.¹³ We choose the models GPT-3.5-TURBO, GPT-4, GPT-4-TURBO and GPT-4O. The first model is a fast, inexpensive model used for simple tasks. The second and third models are the previous set of high-intelligence models and the last model is currently the fastest and most affordable flagship model by OpenAI. Furthermore, we experiment with three open-source models. These are GEITJE-7B-ULTRA¹⁴, LLAMA-3-8B-DUTCH¹⁵, and FIETJE-2-INSTRUCT.¹⁶ We choose GEITJE-7B-ULTRA since it currently is the largest open-source Dutch language model. The model is based on MISTAL-7B, which reports to outperform LLAMA 2 on all benchmarks.¹⁷ We therefore choose to experiment with 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 choose to experiment with FIETJE-2-INSTRUCT since we want to see the potential results of a substantially smaller model.

For each experiment involving the generators, we generate answers with the OpenAI models ten times for robust results. We use a temperature of 0.0 since we want less “creative” results, yet still see variations in the output of OpenAI models. For the open-source models, we generate answers five times since their output is less often in the instructed format, which means that their answers needs to be extracted manually before automatic evaluation is possible. We use top-K sampling with $K = 50$ and a temperature of 0.2 to create deviation in the

¹²<https://huggingface.co/naver/splade-cocondenser-ensembledistil>

¹³<https://platform.openai.com/docs/models>

¹⁴<https://huggingface.co/BramVanroy/GEITje-7B-ultra>

¹⁵<https://huggingface.co/ReBatch/Llama-3-8B-dutch>

¹⁶<https://huggingface.co/BramVanroy/fietje-2-instruct>

¹⁷<https://mistral.ai/news/announcing-mistral-7b/>

responses by the open-source models.

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.

4.3 NLGen experiment

In order to show that our adjusted evaluation framework can be generalized to a benchmark dataset, we perform an experiment over the NLGen dataset. This dataset is a subtask of MSMARCO for natural language generation.¹⁸¹⁹ The task is to select one passage given a query and ten candidate passages, and use this candidate passage to answer the question in a fluent way that should make sense without any additional context. The dataset contains three features: queries, 10 passages, and answers generated by humans. The passages all indicate whether they have been used for the answer or not, making the dataset suitable for the attributed question answering task. An example can be seen in Table 6.

For the experiment, we use the following three setups: zero-shot, one-shot, and using a two-agent approach. The two agent approach uses two GPT-4O agents of which one firstly selects the most relevant passage, and the second agent generates an answer based on the query and the selected passage. The prompts can these approaches can be found in Table 24, 25, 26, and 27, respectively, in Appendix C. In these experiments we do not perform the retrieval aspect of our RAG pipeline since only 10 passages are given for each query. We also do not provide the system with the information whether the passage was selected by the human, since we use this for evaluation purposes. Furthermore, we only perform the experiments using GPT-4O, since we only want to showcase the generalizability of our extended evaluation framework. We only use the first 1000 entries of NLGEN’s *dev* dataset since we are not training any model. Afterward, we evaluate the model’s answers and report for each metric the score averaged over all used entries.

¹⁸<https://microsoft.github.io/msmarco/>

¹⁹<https://huggingface.co/datasets/din0s/msmarco-nlgen>

Query	Passages	Human Answer
What is the capital of France?	<p>1. Paris is known as the capital city of France, with iconic landmarks like the Eiffel Tower. Url: “https://en.wikipedia.org/wiki/Paris”; is_selected=1</p> <p>2. Paris is known for its delicious cuisine, offering a range of cafes and fine dining experiences. Url: “www.aparisguide.com”; is_selected=0</p> <p>3. The Louvre in Paris houses thousands of artworks, including the Mona Lisa. Url: “www.louvre.fr/en”; is_selected=0</p> <p>4. Paris, the City of Light, dazzles with stunning architecture and vibrant culture. Url: “https://www.parisinsidersguide.com/”; is_selected=0</p> <p>5. The Seine River in Paris is lined with iconic sights like the Eiffel Tower and Notre-Dame. Url: “https://en.wikipedia.org/wiki/Paris”; is_selected=0</p> <p>6. Paris is a global hub for fashion, famous for its haute couture and luxury boutiques. Url: “https://en.wikipedia.org/wiki/Paris”; is_selected=0</p> <p>7. Montmartre in Paris has been a haven for artists and bohemians for over a century. Url: “www.aparisguide.com”; is_selected=0</p> <p>8. The Paris Metro is one of the world’s oldest and most efficient underground transit systems. Url: “www.aparisguide.com”; is_selected=0</p> <p>9. Paris’s parks, like the Luxembourg Gardens, offer peaceful retreats in the bustling city. Url: “https://www.parisinsidersguide.com/”; is_selected=0</p> <p>10. Paris is celebrated for its literary history, home to many famous authors and poets. Url: “www.aparisguide.com”; is_selected=0</p>	The capital of France is Paris.

Table 6: NLGen example row of query, ten passages, and a human answer.

5 Experiments and results

This chapter will cover all the experiment metrics, settings and results carried out in the project. We will discuss these sequentially in Sections 5.1, 5.2, and 5.3.

5.1 Evaluation metrics

In this subsection, we start by elaborating our evaluation metrics for the retriever component of our RAG system. Afterwards, we elaborate on the evaluation metrics for the generated long-form answers produced by our RAG system.

5.1.1 Retriever evaluation

We evaluate the performance of the retrievers using the following metrics: Recall@k and HitRate@k. The formulas can be found in Eq. 3 and Eq. 4, respectively. We calculate these scores for each question in our dataset and return the average score.

$$\text{Recall@k} = \frac{\text{Number of Relevant Sources in Top k}}{\text{Total Number of Relevant Sources}} \quad (3)$$

$$\text{HitRate@k} = \begin{cases} 1 & \text{if at least one relevant source is in the top k} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

As previously mentioned, we evaluate the retriever components of our RAG pipeline using Recall@k and HitRate@k metrics, selecting values of k at 3, 5, and 10. Our goal is to determine which k identifies enough relevant items, particularly as we experiment with shorter prompts. The study by Liu et al. (2024) demonstrated that current LLMs can become “lost” in information when dealing with lengthy input contexts. Furthermore, they found that altering the position of relevant information can significantly degrade the LLM’s output.

5.1.2 Long-form answer 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 Attributed QA 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 7 provides an overview of our and ALCE’s evaluation’s

dimensions, definitions, and metrics. From Table 7, it can be seen that we use nine metrics for the evaluation of the long-form answers of the generator in the RAG pipeline. These metrics are coherence, fluency, ROUGE-L, METEOR, consistency, relevance, recall, precision and hitrate. We will elaborate the fluency and coherence in the fluency subsection, the ROUGE-L, METEOR, consistency, and relevance metrics in correctness subsection, and the citation recall and precision metrics in citation subsection.

Evaluation dimension	Definition	ALCE's Metrics	Our Metrics
1). Fluency	Whether the model's generated text is fluent and coherent	<ul style="list-style-type: none"> • MAUVE 	<ul style="list-style-type: none"> • Fluency • Coherence
2). Correctness	Whether the answer is accurate and covers all aspects of interest	<ul style="list-style-type: none"> • Exact match recall • Recall-5 • Claim recall 	<ul style="list-style-type: none"> • ROUGE-L • METEOR • Consistency • Relevance
3). Citation Quality	Whether the answer is well supported by the cited passages and no irrelevant passages are cited	<ul style="list-style-type: none"> • Citation recall • Citation precision 	<ul style="list-style-type: none"> • Citation recall • Citation precision • HitRate@k

Table 7: Evaluation dimensions and associated metrics of ALCE and our work.

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 argue to disregard MAUVE due to its unstableness and propose using 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 coherence and fluency 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. The prompts can be found in Table 20 and Table 21 in Appendix B, respectively.

Correctness

The ALCE framework uses three different datasets in their work and uses a different method for each to calculate the model response’s correctness. In their work, they use exact match recall, recall-5, and a 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 ROUGE and METEOR. We are able to use ROUGE and METEOR since the dataset in our work contains ground truths long-form answers. These metrics calculate the overlap between the ground truth and model’s answer. We will discuss these individually in the sections below.

ROUGE (Lin, 2004) is originally used to calculate the overlap between man made and machine made summaries. In our work, we use ROUGE as a metric to calculate the overlap between the given long-form ground truth answer, and the generated answer by the LLM. The calculation can be seen in Eq. 5. Here, L stands for the longest common sequence. In this Equation, $count_{match}$ represents the number of L-grams that co-occur in the generated summary and the reference answers. $Count$ is the total number of L-grams in the reference answers.

$$ROUGE-L = \frac{\sum_{S \in \text{reference answers}} \sum_{L\text{-gram} \in S} count_{match}(L\text{-gram})}{\sum_{S \in \text{reference answers}} \sum_{L\text{-gram} \in S} count(L\text{-gram})} \quad (5)$$

METEOR (Banerjee and Lavie, 2005) is originally used to evaluate machine translation output. In our work, METEOR is used as the second metric to evaluate the quality of long-form ground truth answers. It does so by calculating the overlap of the long-form answer a human wrote and the long-form answer our RAG system generated. It calculates a score based on the harmonic mean of precision and recall of unigram matches. METEOR offers several features that ROUGE does not, including stemming and synonymy matching.

The METEOR score for an alignment is computed using Eq. 6a. Here, a score is calculated for each reference answer and generated answer, then the aggregated score over the entire set is reported. Eq. 6a uses two subequations. The first we will discuss is Eq. 6b, this calculates the F_{mean} via a harmonic mean. In this equation, the unigram recall (R) is computed as the ratio of the number of unigrams in the system translation that are mapped to the total number of unigrams in the reference translation. The unigram precision (P) is computed as the ratio of the number of unigrams in the system translation that are mapped to the total number of unigrams in the system translation. The second subequation we will discuss is Eq. 6c. Here, a penalty is calculated using *chunks* within answer. For example, if the ground truth sentence is “The duck swam across the lake”, and the generated sentence is “The duck then swam across the lake”, then there are two chunks: “the duck” and “swam across the lake”. METEOR uses

the penalty to reduce the score when the word order in the translation is different from the word order in the reference translation.

$$\text{METEOR} = (1 - \text{penalty}) \cdot F_{mean} \quad (6a)$$

$$F_{mean} = \frac{10 \cdot P \cdot R}{9 \cdot P + R} \quad (6b)$$

$$\text{penalty} = 0.5 \left(\frac{\# \text{ chunks}}{\# \text{ unigrams_matched}} \right)^3 \quad (6c)$$

Even though ROUGE and METEOR provide valuable insights, we argue that ROUGE and METEOR are not giving a complete picture since there are limitations of basing the correctness based of word overlap. Therefore, we additionally use 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. The instruction prompts can be found in Table 22 and Table 23 in Appendix B, respectively.

Citation quality

The ALCE framework calculates the citation quality using recall and precision through prompting a language model to automatically examine whether the cited passages entail the answer. In the datasets used in ALCE there are no ground truth attributions. In their work, the citation quality is calculated using the recall and precision of *each statement (0 or 1)*, averaged over all statements in the model response. We use a more accurate 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. The formulas can be found in Eq. 7 and Eq. 8, respectively. Lastly, we also use HitRate@k to measure whether there was at least one relevant article found for a query. The formula can be found in Eq. 4.

$$\text{Recall} = \frac{\text{Correctly Cited Sources}}{\text{Correctly Cited Sources} + \text{Missed Relevant Sources}} \quad (7)$$

$$\text{Precision} = \frac{\text{Correctly Cited Sources}}{\text{Correctly Cited Sources} + \text{Incorrectly Cited Sources}} \quad (8)$$

5.2 Experimental setup

We first conduct experiments on the retriever component of 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. Additionally, we experiment with our entire RAG system on a larger knowledge corpus. Lastly, we perform an experiment on the NLGen dataset to demonstrate our evaluation metrics for attributed question answering based on ALCE.

We perform two experiments on our system’s retrieval component. The experiments focus is on the input data. The first experiment involves embedding the text from the legal articles, while the second experiment incorporates the text from the legal articles and all available metadata for the legal articles. These two experiments are conducted for a small and a large knowledge corpus, to reflect the two use cases mentioned earlier. This results to a total of four experiments.

We conduct three experiments with our RAG pipeline using the small knowledge corpus. We then selected the best performing setting and four best performing LLMs for an experiment using the larger knowledge corpus.

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 a *temperature* = 0 and with *max_tokens* = 1000. The open source LLMs generated text using a *temperature* = 0.4 and with *max_tokens* = 5000. These parameters differ because of GPT models’ non-deterministic output when using *temperature* = 0. Furthermore, the difference in tokens between the GPT models and open-source differ due to the open-source LLMs often requiring more tokens since they first analyze all given documents to formulate their answer. 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.

In addition to these experiments with our custom created dataset on our RAG system, we also experiment with the NLGen dataset. For our experiment on the NLGen dataset, we use the first 1000 rows of the development set.²⁰ In this experiment, we used a *temperature* = 0.3 with a maximum of 300 output tokens for the GPT generators. The *temperature* variable was increased from the previous experiments with GPT since the questions are open-domain and can be rephrased in various ways.

²⁰<https://huggingface.co/datasets/din0s/msmarco-nlgen/viewer/default/dev>

In regards to the evaluation of the datasets, we firstly processing the LLM's answers and evaluate these answers automatically on the evaluation metrics through numerical values as floating-point numbers between 0 and 1. We repeat each experiment for GPT models ten times, and five times for open-source models due to the manual extraction as mentioned before. After evaluating all results, we calculate a mean and standard deviation for each metric.

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 TNO 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 use Pyserini²¹ for BM25 and TREC Eval²² to calculate the *recall@k* score.

5.3 Results

In the following subsection, we first present the results of the experiments regarding the retriever component of our RAG pipeline in Section 5.3.1. Then, we present the results regarding the entire RAG pipeline with the legal dataset in Section 5.3.2. Lastly, we present the results regarding the NLGen dataset in Section 5.3.3.

5.3.1 Retriever component

The results of using only the text from articles as input for the retriever embeddings on the small and large knowledge corpus can be found in Tables 8 and 9. Similarly, the results for using both the text data and metadata on the small and large knowledge corpus can be found in Tables 10 and 11. In Tables 8 and 9, we compare the baseline model BM25 with dense and hybrid retrievers on the smaller knowledge corpus. The tables demonstrate that the baseline is consistently outperformed by all E5 models in both settings. Upon examining the best-performing model, we find that the E5_LARGE model achieved the highest scores for all evaluation metrics in Table 8. However, in Table 9, we observe that the E5_BASE model outperformed the others on 5 out of 6 metrics. To highlight which data embedding type and model performed best for each evaluation metric, we have underlined these values in the tables. Specifically, the E5_LARGE model embedded only on the legal text shows the highest recall@3 and hitrate@3 scores. Conversely, all other metrics were outperformed by the E5_LARGE model embedded on the article's text and its metadata. Furthermore, it can be seen that some models, such as E5_BASE and SBERT_DUTCH_NLI, consistently perform better when adding metadata. On the other hand, most retrieval models

²¹<https://github.com/castorini/pyserini>

²²https://github.com/cvangysel/pytrec_eval

show a weaker performance when metadata was included when generating the embeddings.

Model	#Param	R@3	R@5	R@10	Hit@3	Hit@5	Hit@10
Sparse							
1 BM25	-	0.692	0.741	0.834	0.814	0.853	0.971
Dense							
2 SBERT _{MULTILINGUAL}	117.7M	0.433	0.529	0.602	0.510	0.637	0.735
3 SBERT _{DUTCH}	109.1M	0.566	0.610	0.693	0.686	0.725	0.824
4 BERT _{DUTCH LEGAL}	167.4M	0.198	0.247	0.309	0.245	0.304	0.373
5 E5-multilingual _{SMALL}	117.7M	0.731	0.787	0.839	0.853	0.912	0.951
6 E5-multilingual _{BASE}	278.0M	0.735	0.795	0.856	0.873	0.912	0.971
7 E5-multilingual _{LARGE}	559.9M	0.785	0.815	0.885	0.941	0.941	0.980
8 DRAGON	109.5M	0.392	0.476	0.564	0.480	0.588	0.657
Hybrid							
9 SPLADE	109.5M	0.630	0.709	0.797	0.765	0.863	0.931

Table 8: Retrieval scores for sparse, dense, and hybrid retrievers were evaluated using text from the article to generate embeddings for each document in a smaller knowledge corpus, which consists of 17 laws.

Model	#Param	R@3	R@5	R@10	Hit@3	Hit@5	Hit@10
Sparse							
1 BM25	-	0.478	0.511	0.583	0.588	0.598	0.686
Dense							
2 SBERT _{MULTILINGUAL}	117.7M	0.425	0.521	0.595	0.520	0.627	0.716
3 SBERT _{DUTCH}	109.1M	0.575	0.619	0.707	0.716	0.755	0.824
4 BERT _{DUTCH LEGAL}	167.4M	0.265	0.247	0.309	0.333	0.304	0.373
5 E5-multilingual _{SMALL}	117.7M	0.597	0.708	0.832	0.706	0.804	0.931
6 E5-multilingual _{BASE}	278.0M	0.747	0.838	0.900	0.882	0.971	0.980
7 E5-multilingual _{LARGE}	559.9M	0.744	0.784	0.890	0.892	0.931	0.971
8 DRAGON	109.5M	0.406	0.472	0.540	0.500	0.569	0.627
Hybrid							
9 SPLADE	109.5M	0.565	0.653	0.745	0.666	0.758	0.863

Table 9: Retrieval scores of sparse, dense, and hybrid retrievers using all available metadata, including *document ID*, *law ID*, *law name*, *chapter*, *chapter title*, *section*, *section title*, *paragraph*, *paragraph title*, *subparagraph title*, *title title*, *article*, *article name*, and *text* in the smaller knowledge corpus, which consists of 17 laws.

In Tables 10 and 11, we compare the baseline model BM25 to dense and hybrid retrievers on the larger knowledge corpus. We can see a clear pattern in the performance of the retriever models when comparing the retrievers based on the smaller and larger corpus. For both sizes, we can see that the E5-MULTILINGUAL_{LARGE} model gives the highest performance when embedding only on the text of the articles. Furthermore, we can see that the E5-MULTILINGUAL_{BASE} model is the best performer when including metadata in the embedding. Lastly, we can see

a pattern of the E5-MULTILINGUAL_{LARGE} model reaching the highest scores for the metrics R@3 and Hit@3, while the E5-MULTILINGUAL_{LARGE} model performs best on all other metrics.

Additionally, despite the larger knowledge corpus (which increased from 17 laws (4,653 chunks) to 273 laws (30,803 chunks)), we find that most retriever models only experience a slight decrease in performance scores. The models that are not multilingual or trained for the Dutch language decrease the most in their performance.

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 SBERT _{MULTILINGUAL}	117.7M	0.404	0.426	0.500	0.510	0.529	0.627
3 SBERT _{DUTCH}	109.1M	0.516	0.583	0.616	0.618	0.696	0.745
4 E5-multilingual _{SMALL}	117.7M	0.674	0.732	0.803	0.794	0.853	0.912
5 E5-multilingual _{BASE}	278.0M	0.696	0.755	0.816	0.843	0.892	0.941
6 E5-multilingual _{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 10: Retrieval scores of sparse, dense, and hybrid retrievers using only the text from the article of each document in the larger knowledge corpus, which consists of 273 laws.

Model	#Param	R@3	R@5	R@10	Hit@3	Hit@5	Hit@10
Sparse							
1 BM25	-	0.419	0.467	0.521	0.520	0.549	0.627
Dense							
2 SBERT _{MULTILINGUAL}	117.7M	0.358	0.425	0.522	0.451	0.539	0.637
3 SBERT _{DUTCH}	109.1M	0.489	0.544	0.639	0.598	0.657	0.755
4 E5-multilingual _{SMALL}	117.7M	0.588	0.659	0.795	0.686	0.745	0.902
5 E5-multilingual _{BASE}	278.0M	0.706	0.814	0.866	0.843	0.951	0.971
6 E5-multilingual _{LARGE}	559.9M	0.700	0.771	0.861	0.843	0.912	0.961
7 DRAGON	109.5M	0.252	0.308	0.424	0.324	0.392	0.520
Hybrid							
8 SPLADE	109.5M	0.469	0.516	0.656	0.549	0.588	0.755

Table 11: Retrieval scores of sparse, dense, and hybrid retrievers using all available metadata, including *document ID*, *law ID*, *law name*, *chapter*, *chapter title*, *section*, *section title*, *paragraph*, *paragraph title*, *subparagraph title*, *title title*, *article*, *article name*, and *text* in the larger knowledge corpus, which consists of 273 laws.

5.3.2 Generator component

Table 12, shows the results for our RAG pipeline using our Question and Answer dataset with the smaller knowledge corpus. In this setup, $k=5$, and the MULTILINGUAL-E5-BASE model was used for the retrieval of the documents. Here, we observe that all LLMs except for FIETJE-2B can generate coherent and fluent answers. Regarding the correctness of the answers and the citation metrics, the propriety 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.

		Fluency		Correctness				Citation quality		
		COH	FLU	ROU	MET	CON	REL	R	P	Hit
GPT										
1	GPT-3.5-turbo-0125	0.829 \pm 1.3	0.974 \pm 0.1	0.571 \pm 0.5	0.742 \pm 0.3	0.945 \pm 0.8	0.957 \pm 0.3	0.564 \pm 0.9	0.776 \pm 1.5	0.856 \pm 0.8
2	GPT-4-0613	0.879 \pm 1.0	0.965 \pm 0.1	0.634 \pm 0.5	0.706 \pm 0.6	0.934 \pm 0.8	0.946 \pm 0.2	0.594 \pm 1.3	0.580 \pm 1.7	0.825 \pm 1.6
3	GPT-4-turbo-2024-04-09	0.877 \pm 0.8	0.970 \pm 0.3	0.586 \pm 0.4	0.708 \pm 0.7	0.951 \pm 0.4	0.950 \pm 0.5	0.594 \pm 1.0	0.694 \pm 0.9	0.850 \pm 0.8
4	GPT-4o	0.875 \pm 0.6	0.973 \pm 0.2	0.652 \pm 0.4	0.745 \pm 0.4	0.950 \pm 0.6	0.956 \pm 0.2	0.625 \pm 1.0	0.754 \pm 1.5	0.897 \pm 0.5
Open source LLMs										
5	GEITje-7B-ultra	0.861 \pm 1.8	0.953 \pm 0.5	0.295 \pm 1.2	0.274 \pm 1.2	0.704 \pm 5.2	0.835 \pm 2.7	0.140 \pm 3.0	0.149 \pm 2.2	0.206 \pm 4.8
6	Llama-3-8B-dutch	0.743 \pm 5.3	0.951 \pm 0.7	0.321 \pm 0.6	0.374 \pm 2.4	0.593 \pm 4.1	0.725 \pm 3.6	0.224 \pm 1.4	0.170 \pm 1.0	0.325 \pm 2.1
7	Fietje-2B	0.091 \pm 0.4	0.066 \pm 0.1	0.074 \pm 0.2	0.106 \pm 0.1	0.010 \pm 0.0	0.030 \pm 0.2	0.000 \pm 0.0	0.000 \pm 0.0	0.000 \pm 0.0

Table 12: Performances of the LLMs on our dataset with the smaller knowledge corpus using the best performing retriever, $mE5_{base}$, with $k = 5$. 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)

Table 13 shows the results for our RAG pipeline using our Question and Answer dataset with the small knowledge corpus. In this setup, $k=3$, and the MULTILINGUAL-E5-LARGE model was used for the retrieval of the documents. Here, we observe that the highest recall score of all settings has been found using GPT-4O. Furthermore, we observe a substantial improvement in the citation metrics for the open-source models GEITJE-7B-ULTRA and LLAMA-3-8B-DUTCH.

The results in Table 14 show the results for our RAG pipeline using our Question and Answer dataset with the smaller knowledge corpus. In this setup, $k=3$, and the MULTILINGUAL-E5-LARGE model was used for the retrieval of the documents. Furthermore, a shortened instruction prompt was used. Here, we observe that the highest precision score of all settings has been found using GPT-4O. Furthermore, we observe a substantial difference in all metrics for the open-source model FIETJE-2B, while the citation metrics for the other open-source models declined.

		Fluency		Correctness				Citation quality		
		COH	FLU	ROU	MET	CON	REL	R	P	Hit
GPT										
1	GPT-3.5-turbo-0125	0.846 \pm 0.5	0.974 \pm 0.2	0.565 \pm 0.5	0.746 \pm 0.4	0.961 \pm 0.4	0.967 \pm 0.2	0.582 \pm 0.3	0.785 \pm 0.8	0.867 \pm 0.5
2	GPT-4o	0.886 \pm 0.5	0.971 \pm 0.2	0.667 \pm 0.5	0.750 \pm 0.4	0.953 \pm 0.4	0.961 \pm 0.2	0.634 \pm 0.3	0.774 \pm 0.8	0.889 \pm 0.5
Open source LLMs										
3	GEITje-7B-ultra	0.844 \pm 1.5	0.962 \pm 0.4	0.433 \pm 2.4	0.426 \pm 2.3	0.813 \pm 1.8	0.899 \pm 1.5	0.327 \pm 1.1	0.279 \pm 1.7	0.463 \pm 1.6
4	Llama-3-8B-dutch	0.742 \pm 2.2	0.962 \pm 0.8	0.460 \pm 4.7	0.581 \pm 5.0	0.739 \pm 5.2	0.778 \pm 5.1	0.551 \pm 5.8	0.376 \pm 3.0	0.551 \pm 7.8
5	Fietje-2b	0.349 \pm 1.1	0.527 \pm 1.0	0.023 \pm 0.4	0.066 \pm 0.9	0.293 \pm 0.8	0.351 \pm 0.7	0.000 \pm 0.0	0.000 \pm 0.0	0.000 \pm 0.0

Table 13: Performances of the LLMs on the small knowledge corpus using the best performing retriever for $k = 3$, $mE5_{large}$ embeddings trained on only the text feature.

		Fluency		Correctness				Citation quality		
		COH	FLU	ROU	MET	CON	REL	R	P	Hit
GPT										
1	GPT-3.5-turbo-0125	0.843 \pm 0.5	0.970 \pm 0.1	0.594 \pm 0.6	0.762 \pm 0.3	0.961 \pm 0.8	0.960 \pm 0.2	0.590 \pm 0.5	0.751 \pm 0.8	0.862 \pm 0.3
2	GPT-4o	0.867 \pm 0.7	0.966 \pm 0.2	0.674 \pm 0.4	0.782 \pm 0.3	0.958 \pm 0.3	0.959 \pm 0.2	0.631 \pm 0.4	0.816 \pm 1.0	0.895 \pm 0.5
Open source LLMs										
3	GEITje-7B-ultra	0.814 \pm 3.7	0.958 \pm 1.0	0.433 \pm 1.9	0.419 \pm 1.8	0.822 \pm 1.4	0.865 \pm 1.7	0.207 \pm 2.6	0.274 \pm 1.3	0.278 \pm 3.1
4	Llama-3-8B-dutch	0.742 \pm 3.4	0.955 \pm 0.7	0.354 \pm 1.6	0.438 \pm 3.6	0.612 \pm 2.7	0.708 \pm 4.2	0.307 \pm 2.9	0.322 \pm 2.2	0.414 \pm 2.5
5	Fietje-2B	0.680 \pm 2.4	0.957 \pm 0.6	0.297 \pm 1.3	0.421 \pm 2.1	0.567 \pm 3.0	0.675 \pm 2.3	0.076 \pm 1.7	0.124 \pm 4.3	0.118 \pm 2.6

Table 14: Performances of the LLMs on the small knowledge corpus using the best performing retriever for $k = 3$, $mE5_{large}$ embeddings which are trained using only the text feature, with a 50% shortened instruction prompt.

Table 15 shows us the results of our RAG pipeline on the larger knowledge corpus using the two best GPT models and the two best open-source LLMs. In this experiment, three chunks were retrieved for the answer using the short prompt. In the table we can see that all models performed slightly worse in regards to the recall, precision, Hitrate, ROUGE, METEOR and coherence and consistency metrics. When comparing Tables 13 and 14, we can see that especially the recall and precision metrics decrease when using a larger knowledge corpus.

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

Table 15: Performances of the LLMs on our dataset with the larger knowledge corpus using the best performing retriever for $k = 3$, $mE5_{large}$ embeddings which are trained using only the text feature, with a 50% shortened instruction prompt.

In Table 16, we show a qualitative result comparison for five different settings. The same question is being asked in each setting. Additionally, each setting uses $k = 3$. In this table we also show the reference (human-formulated) answer from our dataset. We can see that most models generate an answer similar to the human answer, except from GEITJE’s answer being substantially longer. The only model that was unable to answer the question somewhat similarly to the human answer is FIETJE 2B.

<p>Question: Wanneer kan de rechtbank het gezag van een ouder beëindigen?</p>
<p>Human answer: De rechtbank kan het gezag van een ouder beëindigen indien: a. een minderjarige zodanig opgroeit dat hij in zijn ontwikkeling ernstig wordt bedreigd, en de ouder niet de verantwoordelijkheid voor de verzorging en opvoeding, bedoeld in artikel 247, tweede lid, in staat is te dragen binnen een voor de persoon en de ontwikkeling van de minderjarige aanvaardbaar te achten termijn, of b. de ouder het gezag misbruikt.</p>
<p>GPT-4o, small dataset: De rechtbank kan het gezag van een ouder beëindigen indien een minderjarige zodanig opgroeit dat hij in zijn ontwikkeling ernstig wordt bedreigd en de ouder niet in staat is de verantwoordelijkheid voor de verzorging en opvoeding binnen een aanvaardbare termijn te dragen, of indien de ouder het gezag misbruikt. Het gezag kan ook worden beëindigd indien het gezag is geschorst, mits aan de genoemde voorwaarden is voldaan.</p>
<p>GPT-3.5, small dataset: De rechtbank kan het gezag van een ouder beëindigen indien een minderjarige zodanig opgroeit dat hij in zijn ontwikkeling ernstig wordt bedreigd, en de ouder niet de verantwoordelijkheid voor de verzorging en opvoeding kan dragen binnen een voor de persoon en de ontwikkeling van de minderjarige aanvaardbaar te achten termijn, of indien de ouder het gezag misbruikt.</p>
<p>GEITje-7B-ultra, small dataset: De rechtbank kan het gezag van een ouder beëindigen indien: a. een minderjarige zodanig opgroeit dat hij in zijn ontwikkeling ernstig wordt bedreigd, en de ouder niet de verantwoordelijkheid voor de verzorging en opvoeding, bedoeld in artikel 247, tweede lid, in staat is te dragen binnen een voor de persoon en de ontwikkeling van de minderjarige aanvaardbaar te achten termijn, of de ouder het gezag misbruikt. het gezag van de ouder kan ook worden beëindigd indien het gezag is geschorst, mits aan het eerste lid is voldaan. b. de ouders gezamenlijk het gezag uitoefenen, wordt na de beëindiging van het gezag van een van hen voortaan het gezag door de andere ouder alleen uitgeoefend. in geval van beëindiging van het gezag van een ouder, die het gezag alleen uitoefent, kan de andere ouder de rechtbank te allen tijde verzoeken met de uitoefening van het gezag te worden belast. dit verzoek wordt ingewilligd indien de rechtbank dit in het belang van de minderjarige noodzakelijk oordeelt. de rechtbank die het verzoek bedoeld bij het vorige lid heeft afgewezen, kan deze beschikking steeds wijzigen. zij doet dit echter slechts op verzoek van de betrokken ouder, en niet dan op grond van omstandigheden, waarmede de rechter bij het geven van de beschikking geen rekening heeft kunnen houden. c. de ouder wiens gezag is beëindigd, op zijn verzoek in het gezag hersteld kan worden indien: herstel in het gezag in het belang van de minderjarige is, en de ouder duurzaam de verantwoordelijkheid voor de verzorging en opvoeding van de minderjarige, bedoeld in artikel 247, tweede lid, in staat is te dragen. indien ter gelegenheid van de beëindiging van het gezag het gezag aan de andere ouder is opgedragen, belast de rechtbank de ouder wiens gezag was beëindigd en deze alleen het in het eerste lid bedoelde verzoek doet, niet met het gezag, tenzij de omstandigheden na het nemen van de beschikking waarbij het gezag aan de andere ouder werd opgedragen, zijn gewijzigd of bij het nemen van de beschikking van onjuiste of onvolledige gegevens is uitgegaan. artikel 253e is van overeenkomstige toepassing.</p>
<p>Llama-3-8B-dutch, small dataset: De rechtbank kan het gezag van een ouder beëindigen wanneer een minderjarige zodanig opgroeit dat hij in zijn ontwikkeling ernstig wordt bedreigd en de ouder niet de verantwoordelijkheid voor de verzorging en opvoeding in staat is te dragen binnen een voor de persoon en de ontwikkeling van de minderjarige aanvaardbaar te achten termijn, of de ouder het gezag misbruikt.</p>
<p>Fietje-2B, small dataset: Het koninklijk gezag kan worden beëindigd indien de koning de leeftijd van achttien jaar niet heeft bereikt.</p>

Table 16: Example of long-form answers from various generators for the same setup.

5.3.3 NLGen dataset

In Table 17, we present the results of our experiment on the NLGen dataset, highlighting the utility of our extended evaluation framework. The results show that the one-shot setting of GPT-4o outperformed other configurations across most performance metrics, including fluency, coherence, correctness, and citation quality. Specifically, the one-shot model achieved a strong performance in fluency and correctness metrics, with scores that are competitive with or slightly below those of the supervised models on the NLGen leaderboard, which were trained on thousands of examples. It is worth noting that precision, recall, and hit rate metrics show the same score because there is only one relevant document to be selected.

A critical insight from our evaluation approach is the broader understanding of model performance that it provides compared to traditional single-metric evaluations like ROUGE-L, which the NLGen leaderboard uses. ROUGE-L is useful for measuring word overlap, but contains limitations since there are multiple correct formulations possible.

By incorporating additional metrics in three dimensions: fluency, correctness and citation quality, our framework offers a more nuanced and bigger picture of the model's performance. For example, while ROUGE-L provides insight into the content overlap, our evaluation also considers the factual accuracy (consistency), the relevance of the information provided, and the correctness of citation attributions. This multidimensional approach allows us to identify strengths and weaknesses that would otherwise be overlooked, such as the observation that the two-agent setup resulted in the lowest citation quality despite being fluent and coherent.

		Fluency		Correctness				Citation quality		
		COH	FLU	ROU	MET	CON	REL	R	P	Hit
GPT										
1	GPT-4o (zero shot)	0.805	0.930	0.427	0.443	0.912	0.960	0.497	0.497	0.497
2	GPT-4o (one shot)	0.873	0.970	0.456	0.466	0.932	0.964	0.511	0.511	0.511
3	GPT-4o (two agents)	0.849	0.957	0.408	0.418	0.914	0.963	0.464	0.464	0.464
NLGen Leaderboard										
4	Masque NLGEN Style (2019)	–	–	0.496	–	–	–	–	–	–
5	Selector+Combine-Content-Generator (2019)	–	–	0.487	–	–	–	–	–	–
6	MultiLM NLGen Model (2019)	–	–	0.482	–	–	–	–	–	–
7	MNet-Base(Single) NLGEN (2019)	–	–	0.350	–	–	–	–	–	–

Table 17: Performances of GPT-4o on the NLGen dataset. 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@3* (Hit)

6 Discussion

In our research, we designed, implemented and evaluated a RAG system designed to generate attributed long-form answers to Dutch conditional law questions. Our main research questions were centered around the quality of automatically generated answers, the impact of different retrieval and generator architectures, the effects of enlarging the knowledge corpus, and the suitability and generalizability of evaluation methods for the attributed question answering task.

Overall, the results show that our RAG system can generate fluent and correct answers with an 89.5% hit-rate on the small knowledge corpus (associated with the first user setting) and an 83.0% hit-rate on the large knowledge corpus (associated with the second user setting). The answers are often highly coherent with the ground truth, and the models are capable of citing their sources accurately. Our work revealed that the retrievers act as a bottleneck for our system, as they enable correct answers with a high citation quality. A comparison of the retriever results on the small and large knowledge corpora shows a decrease in Hit@3 by 6.8% point and a decrease in the recall score by 5.6% point. This decrease has an adverse effect on the entire system, leading to a further decrease in these metrics. In fact, the Hit@3 decreased by 6.5% point and the recall by 9.2% point in our entire RAG system when increasing the knowledge corpus in size. This underscores the significance of a robust retrieval component in a RAG system.

Furthermore, the results of the experiments showed substantial differences between the retrievers. We observed that BERT and 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. One possible reason for DRAGON still outperforming BERT may be DRAGON's training approach with contrastive learning. 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.

We also observe substantial differences in the performances of the retrievers when adding metadata. Some models consistently underperformed since they probably perceive the additional data as noise. However, it is interesting to note that the highest scores for R@5, R@10, Hit@5, and Hit@10 were achieved when *including* metadata, while the best scores for R@3 and Hit@3 were obtained when *excluding* metadata. In most of our experiments, using a value of 3 for k resulted in better

scores. As a result, we often do not include the metadata when creating retriever embeddings.

The results 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 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. We often observed that for example, FIETJE-2B, the smallest model used in this work, was only able to generate fluent text when the prompt size was shortened by 50%. 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.

We conclude our discussion by addressing three limitations of our work. The first limitation of our work is related to the law since it can be interpreted in various ways due to varying interpretations. As the saying goes, “Ask two lawyers, get three opinions.” This means that our dataset could be considered incomplete or partly incorrect when reviewed by other legal experts, due to the diverse interpretations of the law. In our research, we consulted only one legal expert, primarily due to the high cost associated with seeking advice from multiple experts. The second limitation of our work is regarding our automatic evaluation framework. While the ALCE framework uses MAUVE to evaluate the fluency of answers, we relied on G-EVAL’s fluency and coherence for assessing fluency, and G-EVAL’s consistency and relevance for evaluating correctness. It is important to note that G-EVAL is based on a GPT model, which may lead to G-EVAL favoring RAG models that also use a GPT model as the generator. More research is needed in order to validate the impartiality of the G-EVAL metrics across different model architectures. The last limitation of our work is the lack of human evaluations. Due to the constraints of time and resources, we were unable to incorporate extensive human evaluations for assessing the quality of our evaluation method and of our results. Future research should include comprehensive human evaluations to ensure the findings are robust and applicable in real-world scenarios.

7 Conclusion

In this thesis, we proposed a modular system using a RAG pipeline to answer Dutch legal questions. To support this approach, we created and released a dataset containing 102 legal question-and-answer pairs in Dutch, verified by a legal expert on legal correctness. We also created two knowledge corpora using 17 and 273 Dutch laws. Each question-and-answer pair contains attributions towards the knowledge corpus that we have created. We demonstrated our system's performance by extending the ALCE framework and extensively performing experiments on the retriever component and the entire RAG system. Lastly, we show that our extension of the ALCE framework can be generalized for benchmark datasets such as NLGen.

In our method, we have pre-processed 273 Dutch law documents to a structured format on an article level. We split these into chunks of a maximum of 150 words and added document identifiers for each. In our experimental setup, we tested various retrievers. These are BM25, SBERT multilingual, SBERT dutch, BERT trained for Dutch legal work, multilingual-E5, DRAGON and SPLADE. We experimented with and without using metadata. Furthermore, we have experimented with LLMs as generators using GPT-3.5-turbo, GPT-4, GPT-4-turbo GPT-4o, GEITje, LLama and Fietje for our RAG system. We have experimented with multiple setups by using *top-k* documents retrieved by the retriever and using a shorter instruction prompt.

The results showed that our RAG system is able to create fluent, correct answers with correct attributions to the Dutch law for the majority of the questions in our dataset when using the multilingual-E5 model as the retriever component and GPT-4o as the generator component in our RAG system. The GPT models achieve the highest scores in all metrics, especially in the correctness and citation quality dimensions. We do notice an impact of error propagation on the performance of our system. The retriever component is the bottleneck for the generator component since it fetches the documents that are used in the prompt. We notice a substantial decrease in the precision and recall metrics for the entire RAG pipeline whenever the retriever component decreases their hitrate performance. Furthermore, our experiment on the NLGen dataset showed that our evaluation framework offers a more nuanced and comprehensive picture of the model's performance. Based on our experiment results, we can answer the research questions as follows:

- **RQ1: With what quality can we automatically generate detailed and verifiable answers with attributions to questions in the Dutch legal domain using Retrieval-Augmented Large Language Models?**

In this work, we demonstrate that our RAG system can automatically generate detailed and verifiable answers with attributions in the Dutch legal domain. The system achieves an 89.5% hit rate on a small knowledge corpus, and an 83.0% hit rate on a larger knowledge corpus. This indicates that the system is able to retrieve the relevant law articles for questions. Regarding the automatically generated answers of our RAG system, we consistently observe that the answers receive high fluency scores. However, while fluency remains strong, we notice a slight decline in correctness scores and a more pronounced drop in citation quality as the knowledge corpus expands. This suggests that although the system maintains coherent language generation, accurately citing sources becomes more challenging with a larger corpus.

- **RQ2: In what ways do different retrieval and generator architectures influence the performance and verifiability of long-form answers in Dutch legal question answering?**

Our research reveals that the choice of retrieval and generator components substantially impacts the performance and verifiability of long-form answers in Dutch legal question answering. We found that multilingual retrievers trained for information retrieval tasks performed the best. Future research is needed to investigate whether hybrid retrievers can outperform dense retrievers in the Dutch legal domain. Additionally, our research revealed that proprietary LLM models by OpenAI currently outperform open-source models specifically trained for the Dutch language. The OpenAI models demonstrate a higher capacity to follow complex instructions, making them more suitable for the task of Dutch legal question answering using attributions. Furthermore, we found that including metadata contributes to better retrieval scores in our use cases when $k > 3$. When $k = 3$, excluding metadata leads to higher scores.

- **RQ3: What are the effects on the performance of our RAG system when significantly enlarging the knowledge corpus?**

Our research reveals that significantly enlarging the knowledge corpus has both positive and negative effects on the performance of our RAG system. On the positive side, a larger knowledge corpus provides more laws, reducing the need for user expertise in identifying relevant laws. However, the negative effect is that the retrieval task becomes more challenging for the retrieval component of our RAG system. As the retrieval component is the bottleneck, fetching the most relevant chunks becomes harder, leading to error propagation throughout the system. Consequently, while the correctness of the answers only slightly decreases, the citation quality drops

significantly. This substantial drop in citation quality is likely due to the increased difficulty in pinpointing precise legal references within a larger knowledge corpus.

- **RQ4: What is the most suitable evaluation method for the attributed question answering task, and to what extent can it be generalized over a benchmark dataset?**

Our research proposed an extensive evaluation method for the attributed question answering task. We use the evaluation dimensions from the ALCE framework (Gao et al., 2023), while incorporating (Li et al., 2023a)'s work using G-EVAL to assess the answers' fluency, coherence, consistency, and relevance. Specifically, for fluency, we use metrics such as fluency and coherence. For correctness, we employ ROUGE-L, METEOR, consistency, and relevance. Citation quality is evaluated using citation recall and citation precision, aligning with ALCE's metrics. This approach ensures that the answers are fluent, accurate, and well-supported by relevant citations.

To assess the generalizability of our evaluation method, we conducted experiments using the NLGen benchmark dataset. Our results showed that our one-shot LLM approach achieved performance levels comparable or slightly below those of models extensively trained with 153,725 training examples. Our evaluation method provides a more detailed picture of the results, particularly for citation quality. The limitation here is that precision, recall, and hit rate are all the same due to the dataset containing only one relevant item per query and the model being instructed to output a single response. Overall, we believe that our evaluation method generalizes well for different datasets and offers insights into model performance across various dimensions.

For future work there are multiple directions worth exploring. Firstly, it would be interesting to collect large amounts of data to be able to fine-tune LLMs for the AQA task in the Dutch language. Given the strong performance of the multilingual retrievers, expanding the dataset to include more jurisdictions and document types could provide further insights. Secondly, it would be interesting to include a multilingual hybrid retriever in experiments for our retrieval component. In our experiments we have seen that SPLADE, a hybrid retriever, performed stronger in all metrics than BM25 even though it has been trained for the English language. Lastly, further research investigating the correlation between our evaluation framework's results with human judgments would provide valuable insights.

References

- Amouyal, S. J., Rubin, O., Yoran, O., Wolfson, T., Herzig, J., and Berant, J. (2022). Qampari: An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *ArXiv, abs/2205.12665*.
- Anand, D. and Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University - Computer and Information Sciences*, 34(5):2141–2150.
- Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2023). Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.
- Askari, A., Verberne, S., and Pasi, G. (2022). Expert finding in legal community question answering. In *European Conference on Information Retrieval*, pages 22–30. Springer.
- Askari, A., Yang, Z., Ren, Z., and Verberne, S. (2024). Answer retrieval in legal community question answering. In *European Conference on Information Retrieval*, pages 477–485. Springer.
- Bakker, R., van Drie, R. A., de Boer, M., van Doesburg, R., and van Engers, T. (2022a). Semantic role labelling for dutch law texts. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 448–457.
- Bakker, R. M., de Boer, M. H., van Drie, R. A., and Vos, D. (2022b). Extracting structured knowledge from dutch legal texts: A rule-based approach. In *EKAW (Companion)*.
- Balmer, N. J., Buck, A., Patel, A., Denvir, C., and Pleasence, P. (2010). Knowledge, capability and the experience of rights problems. *London: PLEnet*.
- Banerjee, S. and Lavie, A. (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.
- Bohnet, B., Tran, V. Q., Verga, P., Aharoni, R., Andor, D., Soares, L. B., Ciaranita, M., Eisenstein, J., Ganchev, K., Herzig, J., et al. (2022). Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.
- Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G. B., Lespiau, J.-B., Damoc, B., Clark, A., et al. (2022). Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

-
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chen, A., Yao, F., Zhao, X., Zhang, Y., Sun, C., Liu, Y., and Shen, W. (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.
- Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.
- Currie, A. (2009). The legal problems of everyday life. In *Access to justice*, volume 12, pages 1–41. Emerald Group Publishing Limited.
- Denvir, C. (2016). Online and in the know? public legal education, young people and the internet. *Computers & Education*, 92-93:204–220.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Do, P.-K., Nguyen, H.-T., Tran, C.-X., Nguyen, M.-T., and Nguyen, M.-L. (2017). Legal question answering using ranking svm and deep convolutional neural network. *arXiv preprint arXiv:1703.05320*.
- Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M. (2019). Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.
- Farrow, T. C., Currie, A., Aylwin, N., Jacobs, L., Northrup, D., and Moore, L. (2016). Everyday legal problems and the cost of justice in canada: Overview report. *Osgoode Legal Studies Research Paper*, 12(57).
- Firdaus, V., Saputra, P., and Suprianto, D. (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.
- Firth, J. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32.
- Formal, T., Piwowarski, B., and Clinchant, S. (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.

-
- Gao, T., Yen, H., Yu, J., and Chen, D. (2023). Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.
- Garavano, G. C., Kaag, S., Schwartz, P., Jilani, H., Alvarez, A., Ardyanto, D., Goldston, J., de Greiff, P., Hossain, S., Kennou, K., Maru, V., Maynard-Gibson, A., Molokomme, A., Pell, O., Pais, M. S., Rodriguez, M. F., van Wieren, J., and Osho, B. (2019). *Justice for All: The Report of the Task Force on Justice*. Center on International Cooperation.
- Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. (2020). Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.
- Hoekstra, M. and Teeuwen, G. (2023). Drempels in de toegang tot het recht.
- Honovich, O., Aharoni, R., Herzig, J., Taitelbaum, H., Kukliansy, D., Cohen, V., Scialom, T., Szpektor, I., Hassidim, A., and Matias, Y. (2022). True: Re-evaluating factual consistency evaluation. *arXiv preprint arXiv:2204.04991*.
- Hoppe, C., Pelkmann, D., Migenda, N., Hötte, D., and Schenck, W. (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.
- Huang, C., Wu, Z., Hu, Y., and Wang, W. (2024). Training language models to generate text with citations via fine-grained rewards. *arXiv preprint arXiv:2402.04315*.
- Izacard, G. and Grave, E. (2020). Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.
- Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. (2022). Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.
- Kamalloo, E., Jafari, A., Zhang, X., Thakur, N., and Lin, J. (2023). Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*.
- Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

-
- Khazaeli, S., Punuru, J., Morris, C., Sharma, S., Staub, B., Cole, M., Chiu-Webster, S., and Sakalley, D. (2021). A free format legal question answering system. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 107–113.
- Kim, M.-Y., Xu, Y., and Goebel, R. (2017). Applying a convolutional neural network to legal question answering. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2015 Workshops, LENLS, JURISIN, AAA, HAT-MASH, TSDAA, ASD-HR, and SKL, Kanagawa, Japan, November 16-18, 2015, Revised Selected Papers*, pages 282–294. Springer.
- Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Li, X., Cao, Y., Pan, L., Ma, Y., and Sun, A. (2023a). Towards verifiable generation: A benchmark for knowledge-aware language model attribution. *arXiv preprint arXiv:2310.05634*.
- Li, X., Zhu, C., Li, L., Yin, Z., Sun, T., and Qiu, X. (2023b). Latrieval: Llm-verified retrieval for verifiable generation. *arXiv preprint arXiv:2311.07838*.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Lin, S.-C., Asai, A., Li, M., Oguz, B., Lin, J., Mehdad, Y., Yih, W.-t., and Chen, X. (2023). How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Liu, Y., Iyer, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Long, S., Tu, C., Liu, Z., and Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 558–572. Springer.

-
- Louis, A., van Dijck, G., and Spanakis, G. (2023). Interpretable long-form legal question answering with retrieval-augmented large language models. *arXiv preprint arXiv:2309.17050*.
- Mansouri, B. and Campos, R. (2023). Falqu: Finding answers to legal questions. *arXiv preprint arXiv:2304.05611*.
- Martinez-Gil, J. (2023). A survey on legal question–answering systems. *Computer Science Review*, 48:100552.
- Marwala, T. and Mpedi, L. G. (2024). Artificial intelligence and the law. In *Artificial Intelligence and the Law*, page 27. Springer.
- Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., et al. (2022). Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.
- Pillutla, K., Swayamdipta, S., Zellers, R., Thackstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. (2021). Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.
- Ponce, A., Chamness Long, S., Andersen, E., Gutierrez Patino, C., Harman, M., A Morales, J., Piccone, T., Rodriguez Cajamarca, N., Stephan, A., Gonzalez, K., VanRiper, J., Evangelides, A., Martin, R., Khosla, P., Bock, L., Campbell, E., Gray, E., Gryskiewicz, A., Ibrahim, A., Solis, L., Hearn-Desautels, G., and Tinucci, F. (2019). *Global Insights on Access to Justice 2019: Findings from the World Justice Project General Population Poll in 101 Countries*. World Justice Project.
- Queudot, M., Charton, É., and Meurs, M.-J. (2020). Improving access to justice with legal chatbots. *Stats*, 3(3):356–375.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Ram, O., Levine, Y., Dalmedigos, I., MuhlGay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. (2023). In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.
- Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., Gatford, M., et al. (1995). Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

-
- Rouws, N. J., Vakulenko, S., and Katrenko, S. (2022). Dutch squad and ensemble learning for question answering from labour agreements. In *Artificial Intelligence and Machine Learning: 33rd Benelux Conference on Artificial Intelligence, BNAIC/Benelearn 2021, Esch-sur-Alzette, Luxembourg, November 10–12, 2021, Revised Selected Papers 33*, pages 155–169. Springer.
- Schraagen, M., Bex, F., Van De Luytgaarden, N., and Prijs, D. (2022). Abstractive summarization of dutch court verdicts using sequence-to-sequence models. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 76–87.
- Sheik, R. and Nirmala, S. J. (2021). Deep learning techniques for legal text summarization. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–5. IEEE.
- Socatiyanurak, V., Klangpornkun, N., Munthuli, A., Phienphanich, P., Kovudhikulrungsri, L., Saksakulkunakorn, N., Chairaungsri, P., and Tantibundhit, C. (2021). Law-u: Legal guidance through artificial intelligence chatbot for sexual violence victims and survivors. *IEEE Access*, 9:131440–131461.
- Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.
- Stelmakh, I., Luan, Y., Dhingra, B., and Chang, M.-W. (2022). Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.
- Sun, H., Cai, H., Wang, B., Hou, Y., Wei, X., Wang, S., Zhang, Y., and Yin, D. (2023). Towards verifiable text generation with evolving memory and self-reflection. *arXiv preprint arXiv:2312.09075*.
- Trautmann, D., Petrova, A., and Schilder, F. (2022). Legal prompt engineering for multilingual legal judgement prediction. *arXiv preprint arXiv:2212.02199*.
- van Doesburg, R., van Der Storm, T., and van Engers, T. (2016). Calculemus: towards a formal language for the interpretation of normative systems. *AI4J Artif Intell Justice*, 1:73.
- van Doesburg, R. and van Engers, T. M. (2019). Explicit interpretation of the dutch aliens act. In *AIAS@ ICAIL*, pages 27–37.
- van Drie, R. A., de Boer, M. H., Bakker, R. M., Tolios, I., and Vos, D. (2023). The dutch law as a semantic role labeling dataset. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 316–322.

-
- van Kuppevelt, D. and van Dijck, G. (2017). Answering legal research questions about dutch case law with network analysis and visualization. In *Legal knowledge and information systems*, pages 95–100. IOS Press.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. (2022). Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. (2024). Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.
- Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., and Sun, M. (2020). Jecqa: a legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9701–9708.

A Example prompt and output

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: {*document 1*}, {*document 2*}, {...} , {document k}

(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: {*document 1*}, {*document 2*}, {...}, {*document k*}

Table 18: Prompt template to generate long-form answers. Blue text is the input data that is different for each question.

ANSWER: The court may terminate the guardianship of a natural person if:

1. a minor grows up in such a way that his development is seriously threatened, and the guardian is not able to bear the responsibility for the care and upbringing referred to in Article 247, second paragraph, within an environment that is important for the person and development of the child. minor to be considered acceptable term, or
2. the guardian abuses authority, or
3. does not have the basic consent required under Article 2 of the Adoption of Foreign Children Act.

DOC IDS: DOC0588, DOC0562, DOC0557, DOC0552, DOC0539

Table 19: Output example by the generator that has been translated from Dutch to English, which is an answer for the legal question: “When can the court terminate the guardianship of a natural person?”.

B G-EVAL prompts

You will be given a string generated by a language model. It's an answer to a question. Your task is to rate the string generated by the language model based on one metric. **YOUR TASK IS TO ONLY OUTPUT A NUMBER!** Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Fluency (0-100): 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. 0 is the worst possible rating, and 100 the best possible rating.

Example 1:

Language model answer: "Als een werknemer recht heeft op een uitkering, heeft hij recht op ondersteuning bij het vinden van werk."

Fluency Evaluation: 100

Example 2:

Language model answer: "kan Kaas worden hoed gedragen als en leekan je weten."

Fluency Evaluation: 0

Here comes your task. **REMEMBER, YOUR TASK IS TO ONLY OUTPUT YOUR EVALUATION AS A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT SUCH AS TEXT. ONLY WRITE A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT SUCH AS ENTERS/BREAKS etc.:**

ONLY OUTPUT A NUMBER BETWEEN 0 AND 100!

Model Answer:

{Model answer}

Table 20: Prompt for G-EVAL fluency. Blue text is the input data that is different for each question.

You will be given two strings. These are answers on questions. One string is generated by a human, and the other string by a language model. Your task is to rate the string generated by the language model based on one metric. **YOUR TASK IS TO ONLY OUTPUT A NUMBER!** Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Consistency (0-100) - 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. Annotators were also asked to penalize summaries that contained hallucinated facts. 0 is the worst possible rating, and 100 the best possible rating.

Evaluation Steps:

1. Read the answer by the human carefully and identify the main topic and key points.
2. Read the answer given by the language model and compare it to the answer by the human. Check if the answer by the language model covers the main topic and key points of the human answer, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 0 to 100, where 0 is the lowest and 100 is the highest based on the Evaluation Criteria.

Example 1:

Human answer: "Een werknemer heeft recht op ondersteuning bij arbeidsinschakeling als de werknemer recht heeft op een uitkering."

Language model answer: "Als een werknemer recht heeft op een uitkering, heeft hij recht op ondersteuning bij het vinden van werk."

Consistency evaluation: 100

Example 2:

Human answer: "Een werknemer heeft recht op ondersteuning bij arbeidsinschakeling als de werknemer recht heeft op een uitkering."

Language model answer: "Kaas kan hoed worden als en leer gedragen kan je eten."

Consistency evaluation: 0

Here comes your task. REMEMBER, YOUR TASK IS TO ONLY OUTPUT YOUR EVALUATION AS A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT SUCH AS TEXT. ONLY WRITE A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT such as ENTERS/BREAKS etc.:

ONLY OUTPUT A NUMBER BETWEEN 0 AND 100!

Model Answer:

{Model answer}

Table 21: Prompt for G-EVAL coherence. Blue text is the input data that is different for each question.

You will be given two strings. These are answers on questions. One string is generated by a human, and the other string by a language model. Your task is to rate the string generated by the language model based on one metric. YOUR TASK IS TO ONLY OUTPUT A NUMBER! Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Consistency (0-100) - 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. Annotators were also asked to penalize summaries that contained hallucinated facts. 0 is the worst possible rating, and 100 the best possible rating.

Evaluation Steps:

1. Read the answer by the human carefully and identify the main topic and key points.
2. Read the answer given by the language model and compare it to the answer by the human. Check if the answer by the language model covers the main topic and key points of the human answer, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 0 to 100, where 0 is the lowest and 100 is the highest based on the Evaluation Criteria.

Example 1:

Human answer: "Een werknemer heeft recht op ondersteuning bij arbeidsinschakeling als de werknemer recht heeft op een uitkering."

Language model answer: "Als een werknemer recht heeft op een uitkering, heeft hij recht op ondersteuning bij het vinden van werk."

Consistency evaluation: 100

Example 2:

Human answer: "Een werknemer heeft recht op ondersteuning bij arbeidsinschakeling als de werknemer recht heeft op een uitkering."

Language model answer: "Kaas kan hoed worden als en leer gedragen kan je eten."

Consistency evaluation: 0

Here comes your task. REMEMBER, YOUR TASK IS TO ONLY OUTPUT YOUR EVALUATION AS A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT SUCH AS TEXT. ONLY WRITE A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT such as ENTERS/BREAKS etc.:

ONLY OUTPUT A NUMBER BETWEEN 0 AND 100!

Human Answer:

{Human answer}

Model Answer:

{Model answer}

Table 22: Prompt for G-EVAL consistency. Blue text is the input data that is different for each question.

You will be given two strings. One string is a question, and the other is an answer string by a language model. Your task is to rate the string generated by the language model based on one metric. **YOUR TASK IS TO ONLY OUTPUT A NUMBER!** Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Relevance (0-100) - The answer should include only important relevant to the question. Penalize answers containing redundancies and excess information. 0 is the worst possible rating, and 100 the best possible rating.

Evaluation Steps:

1. Read the question carefully.
2. Assess how well the answer covers the main points of the question, and how much irrelevant or redundant information it contains.
3. Assign a relevance score from 1 to 100.

Example 1:

Question: “Wanneer heeft een werknemer recht op ondersteuning bij arbeidsinschakeling?”

Language model answer: “Een werknemer heeft recht op ondersteuning bij arbeidsinschakeling als de werknemer recht heeft op een uitkering.”

Relevance Evaluation: 100

Example 2:

Question: “Wanneer heeft een werknemer recht op ondersteuning bij arbeidsinschakeling?”

Language model answer: “Mick Jagger is de zanger van de band The Rolling Stones. Zij hebben meerdere hits zoals Paint it Black and Satisfaction.”

Relevance Evaluation: 0

Here comes your task. REMEMBER, YOUR TASK IS TO ONLY OUTPUT YOUR EVALUATION AS A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT SUCH AS TEXT. ONLY WRITE A NUMBER BETWEEN 0 AND 100. DO NOT HAVE ANY OTHER OUTPUT such as ENTERS/BREAKS etc.:

ONLY OUTPUT A NUMBER BETWEEN 0 AND 100!

Human Answer:

{Human answer}

Model Answer:

{Model answer}

Table 23: Prompt for G-EVAL relevance. Blue text is the input data that is different for each question.

C NLGen instruction prompt

Given the following query and passages, generate an answer based on the most relevant passage. Only base your answer on one passage.

Query:
{query}

Passages:
{passage_texts}

Your answer should be in the format:
'Answer: [Your answer here]
Selected Passage: [Passage number here]'.
Make sure to always write only a number!

Table 24: NLGen instruction prompt for zero-shot prompting. Blue text is the input data that is different for each question.

You will be given a task to generate an answer on the most relevant passage. First an example follows.
Example Query: what is a omurice omelet

Passages:

1. In one common rendition, the rice is fried with ketchup, and extra ketchup is squeezed on top as a garnish. In another popular one, seen in the Kyoto video, the chef uses demi-glace (a rich, veal stock-based sauce) to both fry the rice and top the omelette. Japanese mayo is often also squeezed on top.
2. Omurice is a contemporary Asian dish consisting of an omelette made with fried rice. Its name derives from the combination of the English words omelette and rice. 2 A relatively simple dish, it typically calls for rice fried with ketchup, chicken and onions wrapped in a thin sheet of fried egg.
3. In cuisine, an omelette or omelet is a dish made from beaten eggs quickly fried with butter or oil in a frying pan (without stirring as in scrambled egg). It is quite common for the omelette to be folded around a filling such as cheese, chives, vegetables, meat (often ham or bacon), or some combination of the above. Whole eggs or sometimes only egg whites are beaten with a small amount of milk or cream, or even water.
4. How to Make Omelet Rice (Omurice) -Stir frying rice (You will need a wok) Dice carrot, onion, red capsicums, ham and crab stick. Mix the tomato sauce (3 tbsp) and worcestershire sauce in a bowl. Pre heat the wok on high heat for 10 seconds and add some oil. Add all diced ingredients and saute for 1 minute. Reduce the heat to half. Add the steamed rice and the mixed sauce.
5. For those unfamiliar with omurice, it's a Japanese invention that combines an omelette with fried rice. You'll often hear it referred to as omuraisu (a contraction of the words omuretsu and raisu, the Japanese pronunciations of omelette and rice), or omumeshi, which fully translates rice into Japanese.
6. In it, a chef in Kyoto makes a plate of omurice with a deftness and perfection of technique that may be unrivaled. He starts by frying rice in a carbon steel skillet, tossing it every which way until each grain is coated in a sheen of demi-glace and oil.
7. Omurice is a contemporary Asian dish consisting of an omelette made with fried rice. Its name derives from the combination of the English words omelette and rice. Omurice is said to have originated from Japan and it became a popular dish at a western-style restaurant in Tokyo's Ginza district around the turn of the 19th century.
8. Recipe 16 - Omurice. Today in Moto's Kitchen we're going to learn how to make Omurice! This popular dish, notorious in maid cafe's, is an interesting and delicious take on the western omelette. With a base of fried rice, chicken and ketchup, the dish is topped with a simple egg omelette. Eat this tasty dish anytime throughout the day!
9. Another way to change this up is to top the finished omurice with Hayashi sauce or Japanese curry. Omurice With a fluffy omelette covering a bed of savory sweet chicken fried rice, omurice is a modern Japanese classic that kids love.Marc Matsumoto.
10. A cut-open omurice with ketchup. Omurice or omu-rice is an example of yōshoku (a Western-influenced style of Japanese cuisine) consisting of an omelette made with fried rice and usually topped with ketchup. With omu and raisu being contractions of the words omelette and rice, the name is an example of wasei-eigo.

Answer: An omurice omelet is a contemporary Asian dish consisting of an omelette made with fried rice. Selected Passage: 7

Given the following query and passages, generate an answer based on the most relevant passage. Only base your answer on one passage.

Query:
{query}

Passages:
{passage.texts}

Your answer should be in the format:
'Answer: [Your answer here]
Selected Passage: [Passage number here]'

Make sure to always write only a number!

Table 25: NLGen instruction prompt for one-shot prompting. Blue text is the input data that is different for each question.

Given the following query and passages, select the most relevant passage number. Only base your decision on one passage.

Query:
{query}

Passages:
{passage_texts}

Your response should be in the format: 'Selected Passage: [Passage number here]'. Make sure to always write only a number!

Table 26: NLGen instruction prompt for the two-agent approach. This prompt is for the agent tasked to retrieve the most relevant passage. *Blue text is the input data that is different for each question.*

Given the following query and the most relevant passage, generate an answer.

Query:
{query}

Passage:
{passage_texts}

Your answer should be in the format: 'Answer: [Your answer here]'.

Table 27: NLGen instruction prompt for the two-agent approach. This prompt is for the agent tasked to generate an answer based off the given query and passage. *Blue text is the input data that is different for each question.*