Leveraging LLMs to Build a Semi-Synthetic Dataset for Legal Information Retrieval: a Case Study on the Italian Civil Code and GPT4-o

Mattia Proietti[1,*], Lucia Passaro[1,2] and Alessandro Lenci[1]

[1]CoLing Lab, Department of Philology, Literature and Linguistics, University of Pisa
[2]Department of Computer Science, University of Pisa

Abstract

Although raw textual data in the legal domain is abundant, making it easy to collect large amounts of material from several sources, structured and annotated data needed to fine-tune machine learning models is limited and difficult to obtain. Creating human-annotated datasets is both time- and money-consuming, which often makes impractical to get quality data to train machines on various legal language tasks. AI models such as *Large Language Models* (LLMs) are becoming appealing to generate synthetic data, judge model responses, and annotate textual information, so to cope with such shortcomings. In this work, we wish to evaluate the applicability of LLMs for the automatic generation of a dataset of legal query-passage pairs to train retrieval systems. Indeed, *Legal Information Retrieval* (LIR) has been crucial for the creation of robust search systems for legal documents and is now gaining new importance in the context of the *Retrieval Augmented Generation* (RAG) framework, which is becoming a widespread tool to cope with LLMs hallucinating behaviours. Our goal is to test the feasibility of building a query-passage dataset in which the queries are generated by an LLM about real textual passages and assess the reliability of such a process in terms of the generation of hallucination-free data points in a delicate domain, as the legal one. We do so in a two-step pipeline spelt out as follows: i) we use the Italian Civil Code as a source of self-contained, semantically coherent legal textual passages and ask the model to generate hypothetical questions on them; ii) we use the LLM itself to judge the coherence of the questions to spot those inconsistent with the passage. We then select a random subset of the question-passage pairs and ask humans to evaluate them. Finally, we compare human and model evaluations on the randomly selected subset. We show that the model generates many questions easily, and while it lags behind humans when evaluating the appropriateness of the generated questions with respect to the reference passages in zero-shot settings, it substantially reduces the gap with human judgements when only two examples are provided.

Keywords

Large Language Models, Legal Information Retrieval, Synthetic data generation, LLM-as-a-judge, Legal-NLP

1. Introduction

In recent years, we have witnessed great advancements in the field of Artificial Intelligence (AI), in particular in its sub-domain of Natural Language Processing (NLP). The advent of Large Language Models (LLMs), especially on the wave initiated by the GPT family [1, 2], has revolutionised the way we produce, understand, and manipulate textual content. This revolution has permeated all domains, and the legal field is no exception. Indeed, NLP for legal applications is spreading and is gaining a core role in the discussion about the integration of AI into legal practice. However, due to its high degree of specialization, the intellectual complexity of legal tasks, and the technical specificity of its language, the legal domain — similarly to other specialized fields — has progressed more slowly toward a mature integration of language technologies. Despite the vast volume of textual material generated daily by legal practitioners, the field still faces a significant shortage of machine-readable and annotated resources needed to train and fine-tune AI systems for Legal NLP (LNLP) tasks — a process that is complex and presents numerous challenges [3]. The lack of data encompasses all the devisable LNLP tasks. In this work, we focus on data formats necessary to train systems to perform Legal Information Retrieval (LIR) tasks. LIR is a crucial task in the field of LNLP, primarily concerned with retrieving relevant documents in response to a given textual query. A typical application scenario involves a system capable of identifying and returning pertinent legal documents based on a user's question. To effectively perform this task, it is essential to train models on in-domain data—specifically, question-passage pairs derived from legal documents and expressed in legal language—in order to address domain shifts [4]. However, building such datasets purely through human annotation is both extremely time-consuming and costly as it requires coming up with questions and associate them with relevant documents that may be used to answer those questions. To cope with such shortcomings, syn-

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy
*Corresponding author.
mattia.proietti@phd.unipi.it (M. Proietti); lucia.passaro@unipi.it (L. Passaro); alessandro.lenci@unipi.it (A. Lenci)
0009-0002-0447-680X (M. Proietti); 0000-0003-4934-5344 (L. Passaro); 0000-0001-5790-4308 (A. Lenci)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-11-30

thetic data generation and annotation through LLMs is arising as a promising strategy and it is now being explored within the legal domain as well. Despite its ease, the increasing application of LLMs to generate synthetic data calls for a major assessment of their reliability and real applicability for the task at hand.

This paper aims to answer the following research question: "How reliable are automated methods for generating and evaluating semi-synthetic datasets in the context of Legal Information Retrieval?" In turn, the motivation behind this question is two-fold. On the one hand, we want to generate a dataset that can be used to train machine learning systems to perform the task of LIR. On the other hand, we aim to assess the feasibility of this process by evaluating the reliability of using a state-of-the-art LLM both to generate questions and to assess their relevance to reference text passages, as well as the efficiency of this approach in terms of time and cost. We consider this process as a proxy to evaluate the model's ability to understand legal texts at a basic level, since formulating a good question is an index of the degree of understanding reached by the system formulating that question.

To this end, we integrate two established paradigms of LLMs applications: (i) synthetic data generation[5, 6], employed to automatically construct the dataset, and (ii) LLM-as-a-judge[7], used to evaluate and filter out noisy or inaccurate outputs. Specifically, we apply a multi-step strategy involving a state-of-the-art LLM, namely GPT4-o, to generate questions on articles of the Italian Civil Code and evaluate whether the generated questions are answerable by reading the reference article text. We subsequently sample subsets of the generated questions at random and have them evaluated by human annotators using the same criteria as the model, in order to compare the results of automatic and manual evaluation. In that way, we estimate both the question-generation abilities of the LLM and its self-evaluation ability, both of which are crucial for assessing the feasibility of fully automating the process of creating a legal question-answering dataset.

Given the aforementioned lack of datasets to train machine learning models for tasks related to the legal domain and the costs related to manually annotating corpora from the ground up, integrating LLMs in the process of dataset creation is nowadays a promising approach. This work contributes to the understanding of how much we can rely on state-of-the-art LLMs to generate synthetic textual data that are free from hallucinations and that may actually be useful in practical downstream tasks, particularly focusing on the generation of question-passage pairs to be used to train retriever models for LIR and RAG in the legal domain. This aspect is particularly important for low-resource languages and vertical domains, where annotated data is especially scarce. We found that not only the model's performance on generating questions is pretty remarkable in terms of quantity, but it can be almost as good as human judges in the self-evaluation task in 2-shot settings, though it lags behind humans when a 0-shot prompt is used.[1]

2. Related Works

Our work falls in between two paradigms that are becoming standard practice in the NLP community, that is **synthetic data generation** and **LLM-as-a-judge**. As such it is related to a number of works in both those lines of research.

Synthetic Data – Making use of LLMs to generate synthetic datasets is becoming commonplace among NLP practitioners at different stages of the data lifecycle, from generation to curation and evaluation [8]. For example, the Huggingface team has recently released a Python library to automatically generate evaluation benchmarks using LLMs [9]. They implement a protocol they call Document-Evaluation-Generation, dubbed as DG2E. This is relevant to our work, as this framework allows the generation of domain-specific, tailored evaluation benchmarks. However, they used a far more complex strategy, involving multiple LLMs and focusing on the creation of evaluation questionnaires, while we are interested in applying LLMs to generate questions to construct a domain-specific retrieval dataset.

Several relevant works have explored the possibility of generating synthetic questions to build retrieval datasets, either involving LLMs or not. Wang et al. [10] proposed Generative Pseudo Labelling (GPL) to build unsupervised datasets for retrieval, using the encoder-decoder model T5 [11] to generate queries and a cross-encoder to assign pseudo-labels. Ma et al. [12] makes use of synthetic question generation to enhance the zero-shot retrieval abilities of models in target domains. Meng et al. [13] implemented a framework called Augtriever, with which synthetic pseudo-queries are generated by both extracting salient spans from the target reference passage and using NLP text-generation trained on other tasks, such as text summarisation. Tong et al. [14] have applied LLMs to generate synthetic questions to train retrieval models in a protocol they dubbed IGFT (iterative Generation Filtering and Tuning), consisting of iterating the three steps of generating, filtering and tuning synthetic questions to cope with low-quality generated data. Bonifacio et al. [15] leveraged LLMs few-shot generation abilities to build domain-specific synthetic datasets which they used to fine-tune retrievers reported to outperform strong standard baselines trained on data obtained by supervised annotation. Saad-Falcon et al. [16] implements a pipeline of synthetic question generation involving LLMs to build retrieval datasets tailored to target low-resource domains.

[1] Code and the data available at https://github.com/aittam9/cc_qa

LLM-as-a-judge/annotator – LLMs have been recently involved in the process of both annotating data and evaluating model-generated responses. Aldeen et al. [17] evaluates the performance of ChatGPT in annotating texts comparing it with those of human annotators. Savelka [18] use GPT to semantically annotate legal texts in a zero-shot fashion. Wang et al. [19] deploy a human-LLM collaborative protocol for data annotation.

More broadly, LLMs have been used as judges in a variety of works that are relevant to ours, both for the methods employed and the aims pursued. For example, Sun et al. [20] uses LLMs to judge if a the knowledge retrieved as a triplet from s graph is sufficient to answer a given question. Bavaresco et al. [21] tested LLMs as judges on 20 tasks, comparing their judgements with human ones through Spearman's correlation [22] for graded scores and Cohen's k annotator agreement [23] for categorical ones. We refer to Gu et al. [24] for a comprehensive overview of works that have adapted the LLM-as-a-judge paradigm in several ways.

Although a variety of works have addressed the problem of augmenting data for IR through synthetic question generation, to the best of our knowledge, a gap exists both for the Italian language and the Italian legal domain. The same holds for the application of an LLMs as a judge/annotator to evaluate and label data points to build a dataset for LIR. The contribution of our work resides precisely within that frame.

3. Data and Model

Data. We used articles from the Italian Civil Code (ICC) as our source data in order to take advantage of it as a source of short, self-contained and semantically coherent texts. We extracted the articles from the publicly available copy of the ICC offered in Wikisource [2] and saved them as textual passages in plain text. In doing so, we removed all the code meta-textual macro-structure information (*Capi, Titoli, Sezioni*) except for the division in books. We discarded the repealed articles as well as some ill-extracted ones before cleaning and preprocessing the remaining. This process of filtering, cleaning and preprocessing left us with 2927 textual passages. It has to be noted that we considered the Italian Civil Code as a mine of legal textual passages, and our aim was not to model its content or its structure, but to have a reliable source of short legal passages.

Model. We used GPT4-o, an enhanced version of the GPT4 model released by OpenAI [25], accessed through the Python API endpoint. [3] Because it is a proprietary model, details about its technical specifications, architecture, parameters, and the like have not been disclosed to the public.

4. Methodology

Questions Generation. After the data pre-processing and cleaning, we asked GPT4-o to generate questions for each ICC article, treated as a simple text passage. We adapted the number of questions to be asked to the model on the basis of the length of the input article in terms of sentences. To do so, we used the tokenizer of the Spacy Python library [4] to split the articles into sentences. As the Spacy tokenizer is not trained to operate on texts from specific domains, such as the legal one, we customized the standard tokenizer by integrating a long list of abbreviations obtained by expanding those in [26]. In that way, the tokenizer can recognize frequent acronyms patterns like *c.c.* or *art.* and have a better understanding of the sentence boundaries. To meaningfully relate the number of generated questions to the article length, we applied a simple heuristic by which we asked the model to generate a number of questions equal to the number of sentences compsing the article. To avoid excessively noisy generations, we set 8 as the maximum number of questions for the longest articles, if those exceed the length of 8 sentences.

More formally, we take all the articles in the ICC to be a collection of passages P and for any passage $p \in P$ we asked the model M to generate a set of passage-related queries $\mathbf{q}^p = \{q_i^p ... q_n^p\}$ where $n = min(len(p), 8)$ and the length is computed in terms of number of sentences. Then we obtain the total number of queries for all the passages QP, from the union of all the sets of generated queries as $QP = \bigcup_{p \in P} \mathbf{q}^p$.

Figure 1 shows the prompt used to generate the questions.

```
###ISTRUZIONI###:
Sei un esperto in materia di giurisprudenza. Formula {N} domande possibili
a partire dal seguente Testo. Le domande devono strettamente riguardare il
contenuto del testo e null'altro. Restituisci esclusivamente le domande e
null'altro. Numera ogni domanda formulata.

###Testo###
{INPUT TEXT}
```

Figure 1: Prompt used to generate questions

Automatic Questions Evaluation. In a second step, we provided the model with each article paired with the questions it had generated initially and asked it to evaluate whether the answer to each question could be found within the corresponding textual passage. The model was instructed to produce a binary output to facilitate efficient parsing in subsequent evaluation stages. Specifically, the model assigned one of two labels to each

[2] https://it.wikisource.org/wiki/Codice_civile
[3] https://platform.openai.com/docs/overview.
[4] https://spacy.io/

question–passage pair: "*SI*" for a positive match, indicating the answer is present, and "*NO*" for a negative match, indicating it is absent. The question, passage, and instructions were formatted into the prompt illustrated in Figure 2. Therefore, given a pair consisting of a passage $p \in P$, a related question $q^p \in \mathbf{q}^p$ generated in the previous step, and a general template prompt t shown in Figure 2, we built a prompt t^{pq} for each passage-question pair. The model M had to determine if p contains the necessary information to answer q^p, which basically translates into the model performing a binary classification task over the prompt t^{pq}, as shown in 1.

$$M(t^{pq}) = \begin{cases} SI, & \text{if } p \text{ answers } q \\ NO, & \text{otherwise} \end{cases} \quad (1)$$

```
###ISTRUZIONI###
Sei un esperto in giurisprudenza. Di seguito ti verranno mostrati un testo
e una domanda. Il tuo compito è stabilire se la risposta alla domanda è
contenuta nel testo. Puoi utilizzare solo i seguenti due OUTPUT validi:
["SI", "NO"]. L'OUTPUT è "SI" se la risposta alla domanda è contenuta nel
testo. L'OUTPUT è "NO" se la risposta alla domanda non è contenuta nel
testo. Per poter dire "SI" la risposta alla domanda deve essere strettamente e
chiaramente nel testo. Restituisci solamente "SI" o "NO" e null'altro.

###TESTO###
{text}

###DOMANDA###
{query}
```

Figure 2: Prompt used to evaluate questions

We replicate the automatic evaluation on a random subset used to perform the manual evaluation (see below), this time using a 2-shot prompt technique, in which we provided the model with one correct and one incorrect example.

Manual Evaluation. We randomly selected a sample of the generated questions and asked human judges to evaluate whether the answer to the question could be found inside the textual passage (article). Specifically, we randomised the data on two levels. Firstly, we shuffled the whole set of pairs composed of generated questions and reference texts. Secondly, we split the shuffled dataset into subsets of 100 samples each and randomly chose subsets to be annotated by human judges.

We distributed one randomly-selected subset per annotator with no overlap of annotators on the same sets. In that way, we have been able to divide the workload for annotators, asking a single person to annotate samples of 100 items. We estimated that around one hour is required to annotate a sample of that size. All the annotators had an education level of a master's degree or above. They were personally instructed by one of the authors and presented with a Google form providing further instructions and the question-passage pairs to evaluate. The Google Forms have been automatically generated using the TypeScript extension from Google Sheets[5]. We have been able to collect manual annotations for 12 random samples of 100 entries each, for a total of 1200 question-passage pairs. Each question-passage pair to be evaluated has been presented to the annotators as as shown in Figure 3. In this way, the human annotators had to perform the same binary classification task as the model, as illustrated in the previous paragraph, so that 1 can be turned into 2, where H indicate the human performing the task.

$$H(t^{pq}) = \begin{cases} SI, & \text{if } p \text{ answers } q \\ NO, & \text{otherwise} \end{cases} \quad (2)$$

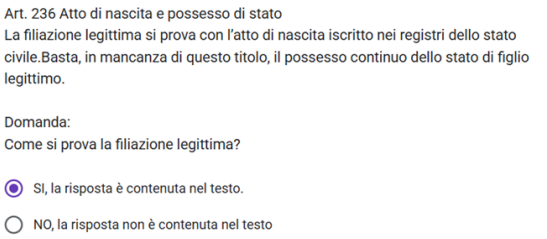

Figure 3: Example question as shown to the human annotators in the google form.

Evaluation cross-comparison. As a last step, we compared the manual and automatic evaluations on the portion we sampled for human annotators. In addition with the 0-shot evaluation already conducted on the whole dataset, we also performed a 2-shot automatic evaluation on the random subsets to have a more comprehensive picture of model's possible performance. Firstly, we simply compared the outputs of the model's evaluation and human evaluation, counting the respective values, that is, how many positive and negative judgments have been provided by each method. Secondly, we treated the human annotation as a gold standard and used it to assess model performance by computing standard machine learning classification metrics such as Precision, Recall and F1, thus having a more nuanced and faithful picture of the relation between human and model evaluations. The primary objective of this step is to evaluate the extent to which the model's judgments, align with human judgments, across all prompts in the randomly selected subsets, considering both zero-shot and two-shot settings.

5. Results

5.1. Generation

The results statistics for the first experiment, that is the generation step, are shown in the Table 1:

[5]This task has been performed with the aid of an LLM.

Book	Input Articles	Generated Questions	Generation Rate
1	392	1115	2.84
2	345	874	2.53
3	359	949	2.64
4	888	2116	2.38
5	623	2132	3.42
6	320	890	2.78
ICC (all)	2927	8076	2.7

Table 1
Statistics of the generated questions across ICC books.

As shown, the model demonstrates strong proficiency in generating questions for each article in terms of quantity, with an average of approximately 3 questions per article, ranging from 2.38 to 3.42 across books. Given a total of 2,927 input articles, the model generated 8,076 questions, effectively doubling or tripling the length of each book.

5.2. Automatic Self-Evaluation

Next, we examine the results of the auto evaluation performed by the model itself and regarding the quality of the generated questions with respect to the input reference text. Figure 4 shows the distribution of the positive and negative values assigned by the model to each pair of generated questions and reference article text. The values are respectively represented by the labels *SI* and *NO* as required by the prompt shown in the previous section in Figure 2, and their distribution is computed per ICC book. In this phase, the model assigned the positive label *SI* to a total of 5369 question-passage pairs, while judging 2692 pairs as negative, which were labelled with *NO*. Additionally, the model failed to provide a legitimate answer (*SI* or *NO*), thus failing to follow the instructions written in the prompt in 15 cases. Overall, the model judged as relevant to the reference article 66% of the questions, thus interpreting as correct only 2/3 of its own generations.

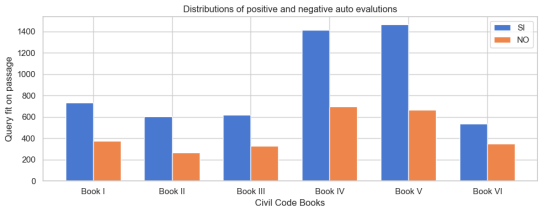

Figure 4: Distribution of labels assigned by the model in the self-evaluation step.

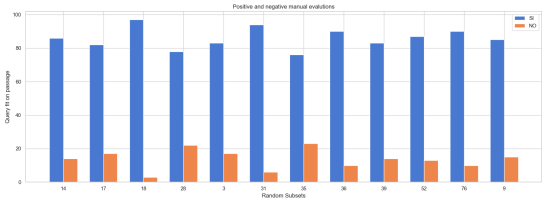

Figure 5: Distribution of labels assigned by humans on the selected random subsets.

Eval Mode	Pos. (SI)	Neg. (NO)	Pos. ratio
HUMAN	1036	164	**0.86**
MODEL-0SHOT	792	408	0.66
MODEL-2SHOT	982	218	0.82

Table 2
Distribution of questions considered as correct (SI) and incorrect (NO) in the aggregated random subsets across evaluation modalities.

5.3. Manual Evaluation

As introduced in the previous section, we randomly selected a subset of the generated questions and asked human evaluators to judge if a question would be good for a given reference passage, thus eliciting the same type of binary judgment obtained by prompting GPT4-o. We did so for 12 sub-sets of data each containing 100 items, for a total of 1200 items. As can be seen from Figure 5, human annotators assigned far more positive labels than negative, as the model itself already did in the zero-shot settings, but with an even greater gap between the two classes, for a total of 1036 (86%) positive labels against 164 (14%) negative ones. The manual evaluation on the random sample seems to point out that the majority of questions generated by the model are, on average, correct with respect to the related text passage.

5.4. Cross Evaluation

We ran a cross-analyisis between HUMAN and MODEL evaluations. As for the latter, we use the zero-shot evaluations previously performed on the whole generated dataset, as well as a new set of 2-shots evaluations elicited for the random subsets assigned to humans. In that way, we could compare HUMAN evaluations against two type of model evaluations, namely MODEL-0SHOT and MODEL-2SHOT. As shown in Table 2, human evaluations assigned the most positive labels (86%), closely followed by the MODEL-2SHOT (82%), while MODEL-0SHOT evaluations lag behind both (66%). In fact, when the model is prompted with no example provided, its evaluations display a gap of around 18-20% compared to the other two modalities. It should be stressed that in that case *positive* and *negative* do not necessarily correspond to correct and incorrect,

	Average	P	R	F1
H@M-0SHOT	**Macro**	0.62	0.72	0.62
	Weigthed	0.85	0.72	0.76
H@M-2SHOT	**Macro**	0.70	0.75	0.72
	Weigthed	0.87	0.85	**0.86**

Table 3
Classification metrics between human (H) evaluations and model (M) evaluations at 0- and 2-shots respectively.

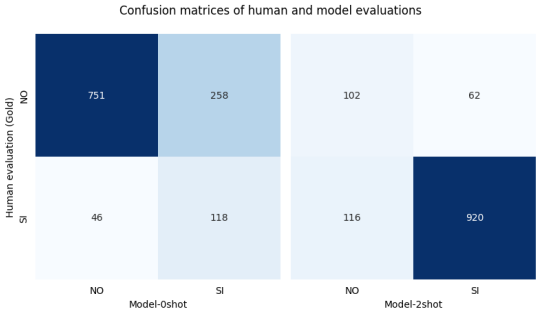

Figure 6: Confusion matrices between HUMAN evaluations and MODEL-0SHOT and MODEL-2SHOT respectively.

but to how an evaluator, human or artificial, has considered the input pair. So, at this stage the comparison between human annotators and the model is more on the dimension of the propensity to assign positive values to the analysed pairs rather than on judging correct responses.

Therefore, we then analysed how the model evaluations performed against the human ones, using the latter as the gold standard, in order to have a more meaningful comparison between HUMAN and MODEL evaluations. As previously stated (see above Section 4), the evaluation task can be formalised as a binary classification task. Therefore, we computed classical machine learing metrics such as Precision, Recall and F1 between human and model annotations. Again, we did so for model's evaluations elicited in 0-shot and 2-shot settings. Results are shown in Table 3.

As expected, given the previous comparisons, the F1 score obtained between HUMAN and MODEL-0SHOT is modest (76%). This is a confirmation of the tendency of the model to underestimate the correctness of the generated questions when prompted with no example whatsoever. This led the model to mislabel lots of items, favouring negative labels, hence leading to a problem of false negatives, as already guessable in previous analysis. While the percentage of false positives assigned by the model is much lower.

On the other hand, the F1 improved of 10 points (86%) when MODEL-2SHOT evaluations are used, substantially levelling the false negatives problem emerged in the 0-shot evaluation. In other words, as it is further summed-up in the confusion matrices shown below in Figure 6, much of the discrepancy between the two evaluation settings depends on the GPT4-o underestimating the goodness of its own generations when the evaluation is led with no examples provided, failing to correctly match a huge number of pairs in which the question and reference article text were positively related. On the contrary, with just one correct and one incorrect examples, the model evaluations align with humans one significantly better.

6. Discussion

We have performed a series of experiments to assess the ability of GPT4-o to generate pertinent legal questions in relation to articles of the Italian Civil Code. We first prompted the LLM to generate the questions, then asked the model itself to judge their goodness, adopting a binary labelling schema. In parallel, we sampled a subset of the generated questions and asked humans to judge their quality with respect to the reference text they were generated from, using the same schema adopted for the model. Next, we compared the kind of evaluation, the automatic made by the model, and the manual performed by human annotators.

Overall, we saw that, as expected, GPT4-o has been generally able to produce an adequate number of questions for each article, as it was stated by our heuristic, which would allow the seamless creation of a dataset to train models for the Legal Information Retrieval task, which may then be integrated into Search Engines or RAG applications. In fact, given the starting set of input texts, we have been able to triple its size in terms of generated questions.

The model's self-evaluation phase seemed to reveal an underestimation of the goodness of the questions by the model itself when it is prompted to perform the task in 0-shots settings. The model judged only 66% of the questions as pertinent to their respective reference text when no example is provieded, initially leading us to think that while it is very good at generating, it underperforms when it comes to evaluating, even though the evaluation concerns its own generated texts. On the other hand, the model has been able to close the gap with human judges in positively evaluating question-passage pairs from a difference of 20% to only 4% when provided with a correct and an incorrect example. While the 0-shot settings underlined a substantial problem of false negatives, this has been substantially reduced in the 2-shot settings. The results show that an SOTA LLM can be

seamlessly used to generate legal content-related questions. It can hardly compete with humans in the 0-shot evaluation of the quality of the same questions with respect to their reference passage, but can better mimic human performance when provided with a negligible number of examples. Overall, all the above hints suggest that using LLMs to cope with the shortage of annotated resources to train machine learning models in the legal domain is an asset worth putting into practice. As stated in previous sections, we used the LLM as a generator to produce questions and as a judge to evaluate the goodness of its own generations. While the LLM-as-a-judge paradigm provides an easy and efficient way to evaluate model responses, its value is not limited to that. Indeed, we can readapt model evaluations and consider them as annotations, with no need to discard incorrect questions, which can be used as negative labels of the generated dataset.

7. Limitations and Future Directions

Some limitations of the present work need to be noted.

First of all, we used a proprietary model. While this choice is apt to our purpose and data, using a closed-source closed-access model implies not being able to precisely define the engine being used, which can undergo updates or modifications without notification. That may hinder the reproducibility and stability of the results across time.

On the side of question evaluation, we used a simple binary approach aiming at identifying whether a question could be answered with the information provided in the document from which it has been generated. While this is straightforward and seamless to implement, it does not allow a more nuanced assessment of the quality of the questions. Therefore, future work is reserved to refining the evaluation approach to introduce additional criteria to assess the quality of a question other than simple answerability (e.g. fluency, ambiguity and the alike). Also, due to resource constraints, we distributed the random samples for the manual evaluation among annotators, assigning a single sample to each one, without overlapping. This made it impossible to assess the soundness of the annotations by computing annotators' agreement measures. In the future, we plan to widen the number of annotated items as well as the pool of annotators, in order to obtain a stronger and more faithful gold standard.

Lastly, in this work, we focused solely on the Italian Civil Code, from which we derived more than 8000 training inputs. Despite being a robust starting point, we are planning to extend the strategy to other Italian Codes, like the Penal Code, in order to both extend the dataset quantitatively and add greater linguistic and conceptual variation qualitatively.

8. Conclusions

In conclusion, integrating LLMs in the process of creating datasets for LNLP tasks is surely a promising and worthwhile route, as it may have many benefits in terms of costs and time efficiency. Indeed, we estimate that the total cost of generating and evaluating questions with GPT4-o is less than 30 dollars, and the amount of time needed to perform the computational experiments is between 15 and 20 hours. These numbers suggest that the process may be easily scalable without a great waste of resources. Also, we showed how the model needs at least two examples to approach the human performance in evaluation, while substantially lagging behind it when a 0-shot prompt is used. While manual evaluation seems to still be the most faithful way to derive gold standards, we estimated that around one hour is necessary for a human to perform an evaluation on a sample of 100 entries, which may become impractical to extend to larger datasets. In contrast, using an LLM to both generate and judge-annotate synthetic questions seems to be a viable alternative to fully automate the process of generating training data for Legal Information Retrieval, providing huge benefits in terms of money and time resources, while maintaining an acceptable performance rate, up to an unavoidable level of noise.

9. Acknowledgments

We are deeply grateful to the volunteer human annotators who have participated in the experiments.

References

[1] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[3] H. Darji, J. Mitrović, M. Granitzer, Challenges and considerations in annotating legal data: A comprehensive overview, 2024. URL: https://arxiv.org/abs/2407.17503. arXiv:2407.17503.

[4] D. Dua, E. Strubell, S. Singh, P. Verga, To adapt or to annotate: Challenges and interventions for domain adaptation in open-domain question answering, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 14429–14446. URL: https://aclanthology.org/2023.acl-long.807/. doi:10.18653/v1/2023.acl-long.807.

[5] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging llm-as-a-judge with mt-bench and chatbot arena, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 46595–46623. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

[6] L. Long, R. Wang, R. Xiao, J. Zhao, X. Ding, G. Chen, H. Wang, On llms-driven synthetic data generation, curation, and evaluation: A survey, 2024. URL: https://arxiv.org/abs/2406.15126. arXiv:2406.15126.

[7] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, K. Shu, L. Cheng, H. Liu, From generation to judgment: Opportunities and challenges of llm-as-a-judge (2025). URL: https://arxiv.org/abs/2411.16594. arXiv:2411.16594.

[8] L. Long, R. Wang, R. Xiao, J. Zhao, X. Ding, G. Chen, H. Wang, On LLMs-driven synthetic data generation, curation, and evaluation: A survey, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 11065–11082. URL: https://aclanthology.org/2024.findings-acl.658/. doi:10.18653/v1/2024.findings-acl.658.

[9] S. Shashidhar, C. Fourrier, A. Lozovskia, T. Wolf, G. Tur, D. Hakkani-Tür, Yourbench: Easy custom evaluation sets for everyone, 2025. URL: https://arxiv.org/abs/2504.01833. arXiv:2504.01833.

[10] K. Wang, N. Thakur, N. Reimers, I. Gurevych, GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 2345–2360. URL: https://aclanthology.org/2022.naacl-main.168/. doi:10.18653/v1/2022.naacl-main.168.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[12] J. Ma, I. Korotkov, Y. Yang, K. Hall, R. McDonald, Zero-shot neural passage retrieval via domain-targeted synthetic question generation, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 1075–1088. URL: https://aclanthology.org/2021.eacl-main.92/. doi:10.18653/v1/2021.eacl-main.92.

[13] R. Meng, Y. Liu, S. Yavuz, D. Agarwal, L. Tu, N. Yu, J. Zhang, M. Bhat, Y. Zhou, Augtriever: Unsupervised dense retrieval by scalable data augmentation, arXiv preprint arXiv:2212.08841 (2022).

[14] Z. Tong, C. Qin, C. Fang, K. Yao, X. Chen, J. Zhang, C. Zhu, H. Zhu, From missteps to mastery: Enhancing low-resource dense retrieval through adaptive query generation, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1, KDD '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 1373–1384. URL: https://doi.org/10.1145/3690624.3709225. doi:10.1145/3690624.3709225.

[15] L. Bonifacio, H. Abonizio, M. Fadaee, R. Nogueira, Inpars: Unsupervised dataset generation for information retrieval, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 2387–2392. URL: https://doi.org/10.1145/3477495.3531863. doi:10.1145/3477495.3531863.

[16] J. Saad-Falcon, O. Khattab, K. Santhanam, R. Florian, M. Franz, S. Roukos, A. Sil, M. Sultan, C. Potts, UDAPDR: Unsupervised domain adaptation via LLM prompting and distillation of rerankers, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 11265–11279. URL: https://aclanthology.org/2023.emnlp-main.693/. doi:10.18653/v1/2023.emnlp-main.693.

[17] M. Aldeen, J. Luo, A. Lian, V. Zheng, A. Hong, P. Yetukuri, L. Cheng, Chatgpt vs. human annotators: A comprehensive analysis of chatgpt for text annotation, in: 2023 International Conference on Machine Learning and Applications (ICMLA), 2023, pp. 602–609. doi:10.1109/ICMLA58977.2023.00089.

[18] J. Savelka, Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts, in: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, ICAIL '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 447–451. URL: https://doi.org/10.1145/3594536.3595161. doi:10.1145/3594536.3595161.

[19] X. Wang, H. Kim, S. Rahman, K. Mitra, Z. Miao, Human-llm collaborative annotation through effective verification of llm labels, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, CHI '24, Association for Computing Machinery, New York, NY, USA, 2024. URL: https://doi.org/10.1145/3613904.3641960. doi:10.1145/3613904.3641960.

[20] J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. M. Ni, H.-Y. Shum, J. Guo, Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph (2024). URL: https://arxiv.org/abs/2307.07697. arXiv:2307.07697.

[21] A. Bavaresco, R. Bernardi, L. Bertolazzi, D. Elliott, R. Fernández, A. Gatt, E. Ghaleb, M. Giulianelli, M. Hanna, A. Koller, A. Martins, P. Mondorf, V. Neplenbroek, S. Pezzelle, B. Plank, D. Schlangen, A. Suglia, A. K. Surikuchi, E. Takmaz, A. Testoni, LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 238–255. URL: https://aclanthology.org/2025.acl-short.20/. doi:10.18653/v1/2025.acl-short.20.

[22] C. Spearman, The proof and measurement of association between two things, The American Journal of Psychology 15 (1904) 72–101.

[23] J. Cohen, A coefficient of agreement for nominal scales, Educational and psychological measurement 20 (1960) 37–46.

[24] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, J. Guo, A survey on llm-as-a-judge, 2025. URL: https://arxiv.org/abs/2411.15594. arXiv:2411.15594.

[25] OpenAI, Gpt-4 technical report, 2024. URL: https://arxiv.org/abs/2303.08774. arXiv:2303.08774.

[26] D. Licari, G. Comandè, Italian-legal-bert models for improving natural language processing tasks in the italian legal domain, Computer Law & Security Review 52 (2024) 105908. URL: https://www.sciencedirect.com/science/article/pii/S0267364923001188. doi:https://doi.org/10.1016/j.clsr.2023.105908.

Declaration on Generative AI

During the preparation of this work, the author(s) did not use any generative AI tools or services.