Are large-language models enough to generate questionnaires? A use-case with clinical trials

Anonymous ACL submission

Abstract

We propose DivAC-LLM, a hybrid pipeline that automatically generates physician-facing questionnaires from clinical trial eligibility criteria. The pipeline integrates rule-based systems, shallow machine learning models, fine-tuned biomedical transformers, and prompt-engineered Large Language Models (LLMs). Unlike purely generative approaches, our system explicitly predicts the number of questions derived from each criterion and injects structured cues into LLM prompts to improve control and accuracy. We validate the approach on two datasets: CTQG, a corpus we created from cancer trials, and QG-SQuAD, a modified version of SQuAD for question generation. Experimental results show that combining symbolic and neural components reduces hallucinations and improves semantic alignment. The results highlight that LLMs benefit from lightweight structure and explicit intermediate reasoning when generating interpretable clinical questionnaires.

1 Introduction

Since the mid-20th century, clinical trials have been the gold standard for evaluating the safety and efficacy of medical interventions (Weed, 1968). Randomized Controlled Trials (RCTs) have guided drug approvals, treatment guidelines, and healthcare policies (Meinert, 2012). However, a bottleneck in this process is eligibility pre-screening (Brøgger-Mikkelsen et al., 2020). Although inclusion and exclusion criteria are designed to identify suitable patient subgroups, they are typically expressed in unstructured free text. As a result, these criteria are often difficult to interpret, even for clinicians, and pose significant challenges to automatic parsing and operationalization (Raghavan et al., 2014).

With the widespread adoption of Electronic Health Records (EHRs) in the 2000s (Blumenthal and Tavenner, 2010), and the public availability of over 450,000 trials on ClinicalTrials.gov[1], the vision of automated patient-trial matching became increasingly plausible. However, the gap between free-text criteria and structured patient data remains unresolved (Kaskovich et al., 2023).

One intermediate solution is the use of structured questionnaires automatically derived from eligibility criteria. These questionnaires, intended for physicians, can be completed semi-automatically using EHR data (Figure 1). They offer interpretability, partial standardization, and actionable guidance—enabling healthcare providers to assess trial eligibility in a more streamlined and reproducible manner.

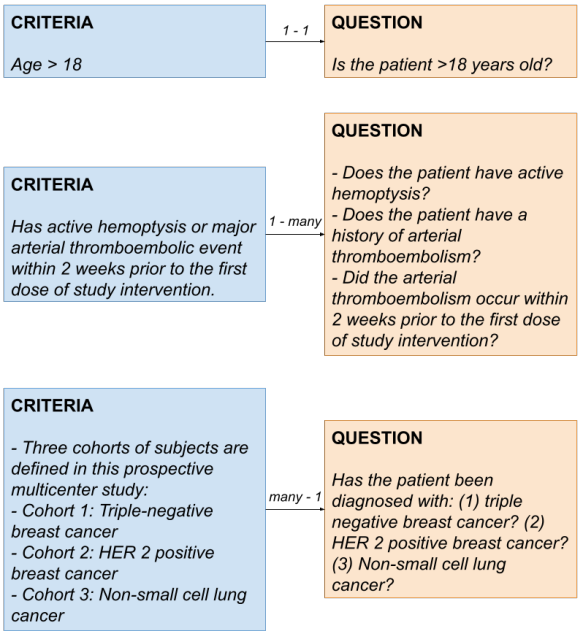

Figure 1: Examples of question generation.

Prior literature has focused on query generation rather than on physician-oriented questionnaire generation (Park et al., 2024a). However, recent advances in LLMs enable prompt-based rephrasing

[1] https://ClinicalTrials.gov

of eligibility criteria into questions, although LLMs alone remain insufficient (Smith et al., 2023). Our hypothesis is that LLMs need to be explicitly informed about the criteria values and the number of questions that a criterion generates, to improve their ability to create a questionnaire from a clinical trial.

To explore this idea, we propose a hybrid pipeline that identifies criteria in trial descriptions, extracts the clinical attributes, and predicts whether a criterion should yield one or multiple questions before prompting an LLM. This produces questionnaires where clinicians fill in patient information rather than interpreting raw text.

We also create a dataset, called CTQG (Clincal Trial Question Generation), extracted from lung cancer clinical trials from ClinicalTrials.gov written exclusively in English, built to train models to translate criteria into questions.

We evaluate our approach on an annotated corpus of real-world clinical trial criteria. In order to compare it with existing methods, we also apply our approach to the SQuAD dataset (Rajpurkar et al., 2018).

The contributions [2] of this paper are:

- An overview of related work on patient prescreening and automated processing of clinical trial criteria.
- A reproducible and operational pipeline for transforming clinical trial criteria into physician-facing questionnaires.
- A set of guidelines for the creation of the CTQG dataset and the public release of the QG-SQuAD dataset.

2 Related Work

Regarding clinical trial eligibility criteria processing, early efforts focused on rule-based extraction pipelines such as EliIE (Kang et al., 2017) and Criteria2Query (Yuan et al., 2019), as well as annotated corpora like Chia (Kury et al., 2020), which transform unstructured criteria into structured representations aligned with database queries.

Text mining from clinical trial documents has also enabled alignment between eligibility criteria and structured patient data (Weng et al., 2011). This includes entity normalization, logical form parsing and OMOP-compatible cohort definitions (Ziletti and D'Ambrosi, 2025). The Leaf Clinical Trials corpus (Dobbins et al., 2022) has provided a large-scale semantic benchmark for such tasks. More recently, LeafAI (Dobbins et al., 2023) introduced neural query generators that achieve retrieval performance comparable to experienced programmers.

Some approaches convert criteria into interactive questions to assist in patient recruitment. DQueST (Liu et al., 2019) generates dynamic "Yes/No" questions to match patients with trials. However, it lacks domain-specific clinical phrasing and does not integrate EHR signals.

LLMs have opened new possibilities for clinical trial matching. (Wornow et al., 2025) apply zero-shot prompting to evaluate trial-patient compatibility and generate natural language justifications. TrialGPT (Jin et al., 2024) fine-tunes generative models for trial matching across thousands of real-world protocols, while FollowupBench (Gatto et al., 2025) addresses clinical question generation from patient-provider dialogues. However, these systems aim for classification or summarization, not the generation of interpretable, structured questions. Besides, they often suffer from factual hallucinations, constraint omission, and semantic overgeneralization (García-Barragán et al., 2025; Pal et al., 2022b).

Hybrid architectures have emerged to address these shortcomings by combining symbolic, statistical, and neural methods. (Li et al., 2024) propose a post-processing framework to reduce hallucinations in medical QA. (Yi et al., 2024) incorporate entity-based pre-filtering before LLM reasoning in trial-patient matching. Finally, (Shi and Bucher, 2025) combine prompt engineering with rule-based parsing for better generalization across eligibility criteria. It is the closest work to ours.

Recent works unleashed the power of LLMs to enhance patient-trial matching. Criteria2Query3 (Park et al., 2024b) used GPT-4 to transform the text from ClinicalTrials.gov into SQL queries. Finally, the PRISM framework (Gupta et al., 2024) fine-tuned an LLM to both simplify eligibilty criteria and realize patient prescreening, but the question generation module does not consider that a criteria can become several questions. However, it is the closest work to ours.

Despite recent progress, no work has tackled structured question generation from clinical trial criteria for physician-facing tools. There is no stan-

[2] The clinical trial codes and the final prompts will be available with the final version of this paper

dard protocol or template to transform criteria into questions: identical meanings can yield distinct formulations across trials.

3 Methodology

3.1 Data

3.1.1 CTQG dataset

The main dataset used in this work has been designed based on a set of clinical trials obtained from the ClinicalTrials.gov platform. This is an online database, administered by the US National Institutes of Health (NIH), which contains information on more than 440,000 clinical studies from 220 countries covering a wide range of diseases and conditions.

These criteria were then refined with medical expert input to establish an explicit correspondence between each criterion and the questions required to operationalize it. An oncologist generated and curated 729 questions using ChatGPT-4o, ensuring normalization so that identical criteria across trials map to the same question. The main prompting setup is provided in Appendix A (Section A.1.1). Criteria were presented in blocks to obtain initial question drafts, which were edited and standardized. CTQG is designed as a high-precision expert-normalized benchmark, prioritizing semantic consistency over scale. The details of the questions creation is available at appendix B.

Some prompts we crafted for ChatGPT are available at Appendix A. No explicit rules were defined to sort criteria according to how many questions they generate.

However, the correspondence between criterion and question is not strictly one-to-one, but can be also one-to-many or many-to-one (see Figure 1). Data have been processed representing this information.

3.1.2 SQuAD dataset

The second dataset employed in this study is the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2018). Released in 2016, SQuAD 1.1 was designed as a realistic benchmark for machine-reading comprehension. It consists of Wikipedia passages paired with human-written questions whose answers correspond to contiguous spans in the text, totalling over 100,000 question–answer pairs.

For this work, SQuAD was transformed to mirror the structure of the CTQG dataset, yielding 101,888 entries. Each passage was split into sentences, their character offsets were recorded, and each question was linked to the sentence containing its answer span (Algorithm 1). We used the NLTK toolkit (Bird and Loper, 2004) for sentence segmentation.

Algorithm 1 Map questions to sentence intervals

Require: Paragraphs $\mathcal{P}$; contexts $\mathcal{C}_p$ per paragraph; for each context c: QA pairs $\mathcal{A}_c = \{(a_{\text{start}}^{(k)}, q^{(k)})\}_k$

for all $p \in \mathcal{P}$ **do**
  for all $c \in \mathcal{C}_p$ **do**
    $[s_1, \ldots, s_L] \leftarrow \textsc{SplitIntoSentences}(c)$
    $o \leftarrow 0$;
    for $i = 1$ to L **do**
      $\ell_i \leftarrow \textsc{IndexOf}(c, s_i, o)$; $r_i \leftarrow \ell_i + |s_i|$; $o \leftarrow r_i$
    end for
    for all $(a_{\text{start}}, q) \in \mathcal{A}_c$ **do**
      find j s.t. $\ell_j \le a_{\text{start}} < r_j$ ▷ binary search or scan
      map $q \to (s_j, [\ell_j, r_j))$
    end for
  end for
end for

Furthermore, correspondences were identified between each original text segment (or "Criterion") and the number of questions generated. The distribution of the number of questions associated with each criterion is as follows: 40,961 (40.2%) with zero questions, 40,467 (39.72%) with one question, 14,536 (14.27%) with two questions, 3,985 (3.91%) with three questions, 1,251 (1.23%) with four questions, and 688 (0.67%) with five questions. This distribution highlights that, unlike CTQG, this dataset includes many cases in which no questions are generated. QG-SQuAD reflects a positional task: most sentences contain zero or one answerable segment, and the number of questions depends on span alignment. In other words, QG-SQuAD reflects positional span alignment, not conceptual decomposition. As a result, QG-SQuAD offers scale and robustness tests but evaluate another skill: local span-driven generation rather than domain-level reasoning.

Both processed datasets were partitioned into training, validation, and test subsets, whose sizes are given in Table 1.

3.1.3 Comparative of datasets

Throughout this section, both datasets have been compared from different perspectives, both globally and with respect to the train, test, and validation partitions. These measures are presented in Table 1.

As can be seen, there is homogeneity within the partitions of each dataset, although certain differences can indeed be detected between them. For example, SQuAD texts are longer in terms of the num-

Dataset	Partition	Size	Vocab. size	Entropy	Characters				Words				Sentences			
					Min	Max	Avg	StD	Min	Max	Avg	StD	Min	Max	Avg	StD
CTQG	*Train*	473	1,879	9.19	5	602	126.62	99.94	1	76	18.77	14.91	1	4	1.15	0.43
	Test	148	925	8.68	9	742	125.40	111.04	1	111	18.90	16.86	1	4	1.15	0.46
	Validation	108	818	8.63	13	573	130.55	125.63	3	93	19.43	19.02	1	4	1.15	0.49
	Total	729	2,244	9.25	5	742	126.95	106.24	1	111	18.90	15.96	1	4	1.15	0.44
SQuAD	*Train*	81,252	91,287	11.27	1	2,458	150.93	78.67	0	341	24.37	12.54	0	12	1.04	0.25
	Test	10,407	26,644	10.95	2	781	150.39	79.04	1	127	24.29	12.55	0	4	1.04	0.23
	Validation	10,229	26,877	10.93	1	1,412	155.68	85.23	0	230	25.13	13.58	0	11	1.05	0.31
	Total	101,888	103,675	11.30	1	2,458	151.35	79.40	0	341	24.44	12.65	0	12	1.04	0.26

Table 1: Statistical summary of the datasets used (CTQG, QG-SQuAD) with respect to the different partitions (train, test, validation, total), including values for size, vocabulary size, Shannon entropy, and the minimum, maximum, mean, and standard deviation of the number of characters, words and sentences. Note that for the SQuAD dataset, a sample of the total was taken for the development of this research.

ber of characters and words, although they contain fewer sentences than CTQG. Similarly, SQuAD shows a broader vocabulary as well as higher entropy, which makes sense given the size differences between the two datasets.

3.2 DivAC-LLM pipeline

To tackle the challenges mentioned in Section 2, our pipeline is divided into two tasks: predicting the number of questions to generate per criterion and the generation itself with criteria extraction. Figure 2 illustrates our pipeline and the next sections explain the different modules.

3.2.1 Question division prediction

This task can either be seen as a regression (approximate the real number of questions to be generated from a criterion) or a classification (categorize the number of questions to be generated). In both cases, it is a supervised learning approach, as we know how many questions were generated for each criterion. To ease the task, we decided to treat it as a ternary classification problem: 0 (no question to be generated), 1 (one question to be generated) and 2 (several questions to be generated).

As it is a predictive task, LLMs are not as strong as most classic fine-tuning approaches (Trust and Minghim, 2024). In addition, since criteria are a short sentence (18.9 words on average), simple solutions can be effective, thus reducing computing time and energy consumptions. We created a voting algorithm in three steps, giving more weight to precision than recall, arguing that it is best that the system ignores how many questions have to be generated rather than guessing a number:

1. We extracted dozens of features from the criterion with SpaCy (Honnibal et al., 2020) (whose list is available in Table 15 at Appendix E) and established correlation patterns between the feature values and the three classes in order to implement a rule-based system. To complete it, we implemented a Random Forest algorithm and we predicted the remaining classes that had not been labeled by the rule-based system, that has great precision but low recall.

2. We fine-tuned two pre-trained transformers specialized in the bio-medical domain (BioBERT (Lee et al., 2019) and BioMedLM (Bolton et al., 2024)) to predict the right class for each criterion.

3. We crafted a voting algorithm with several conditions, where in case of disagreement between the rule-based + Random Forest and the different fine-tuned models, the priorities could be given to a different predictor, or to the "I don't know" class. The description of the different strategies is given in Table 16 in Appendix E.

We chose the voting configuration that maximizes precision with acceptable recall.

The training process is illustrated in Algorithm 2 and the voting system in Algorithm 3, both in Appendix C.

3.2.2 Question generation and criteria management

Two approaches are state-of-the-art, as seen in Section 2: supervised approaches with transformers fine-tuning and zero-shot learning with LLMs. They are end-to-end methods that take as input a criterion in natural language and transform it into questions. The fine-tuning approach can be considered as a machine translation task (Raffel et al., 2020), where the criterion is the source language and the questions the target. The advantage of this

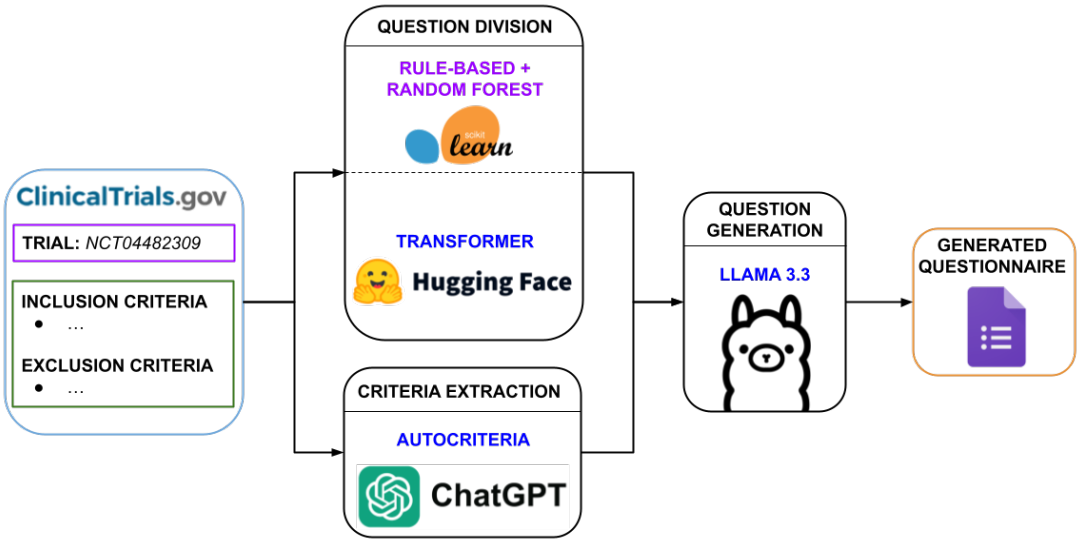

Figure 2: Automatic questionnaire generation pipeline.

technique is that the model learns from real question examples and thus is accurate when "translating". However, in order to work, it requires a parallel corpus of a minimum size (Koehn and Knowles, 2017) in order to be effective. On the other side, the LLM approach requires a few or even no question examples in order to work, due to its huge language model and generative capacities, but the issues are in the risks of hallucinations and variability in its generation.

Nevertheless, as the LLMs approach is based on prompt engineering and instructions crafting (Liu et al., 2023), it is possible to control the output by adding to the prompt information about the criteria. In other words, the LLMs approach can be stronger if used with a previous step where the criteria values are extracted in a NER fashion and passed explicitly to the prompt. We decided to combine two LLMs:

1. One is the state-of-the-art in criteria extraction called AutoCriteria (Datta et al., 2024), a framework based on the GPT-4 model to extract granular eligibility criteria information from clinical trial documents.

2. Another is an instance of Llama 3.3 (Grattafiori et al., 2024), an open-source LLM, easier to fine-tune than commercial ones, that we used in order to generate more accurate questions for each criterion with less variability among clinical trials with a few-shot learning approach where we feed the generic LLM with examples of criteria-questions pairs. The prompts that have been used are presented in Appendix A.1.

The final result is a three-module system (question division, AutoCriteria and LLM) that we called DivAC-LLM, optimized for questionnaire generation from clinical trials.

4 Experimental results and evaluation

4.1 Experimental setup

Regarding the baselines, we used, depending on the dataset:

1. **CTQG:** three state-of-the-art pre-trained transformers, two specialized in the medical domain (BioBART (Yuan et al., 2022), BERT2BERT (Moll et al., 2025)), one generic (Flan-t5 (Smilga and Alabiad, 2024)), ChatGPT-4o and Llama 3.3 against the full DivAC-LLM.

2. **QG-SQuAD:** two state-of-the-art generic pre-trained transformers (Prophetnet (Qi et al., 2020), Pegasus-Xsum (Zhang et al., 2019)), ChatGPT-4o and Llama3.3 against DivAC-LLM.

For the experiments, the hardware, training time and costs are available in Tables 18 and 19 in Appendix E. When fine-tuning transformers models for question generation, we performed a grid search, where the ranges are available at Table 21 in Appendix E. The best hyperparameters for each dataset are available at Table 17 in Appendix E.

Dataset & class	Precision	Recall	F-score
CTQG 0	0.74	0.33	0.45
CTQG 1	**0.84**	**0.71**	**0.77**
CTQG 2	0.33	0.43	0.38
QG-SQuAD 0	**0.69**	0.41	0.52
QG-SQuAD 1	0.38	0.41	0.40
QG-SQuAD 2	0.45	**0.68**	**0.54**

Table 2: Results of question division prediction with the best voting strategy for both CTQG and QG-SQuAD.

4.2 Results evaluation

For question division, we realized a classic automatic evaluation. For question generation, we conducted two distinct evaluations, a human evaluation based on (Nema and Khapra, 2018) and an automatic evaluation based on the QGEval metrics benchmark (Fu et al., 2024).

4.2.1 Question division evaluation

As it is a module that can be activated or deactivated, we evaluate separately the question division prediction on both CTQG and QG-SQuAD. We show in Table 2 the results with the best voting strategy. While the question division module underperforms on QG-SQuAD, due to this module reliance on linguistic heuristics rather than domain-informed labeling, the module achieves better results on CTQG, in the precision of classes 0 and 1. In the Section 4.2.3 we observe the impact of the module for questionnaire generation.

4.2.2 Human evaluation

We compared the results of the benchmarked models with ours (DivAC-LLM) using a human evaluation. We used three metrics with three annotators (3 Spanish males between 30 and 40, 2 NLP engineers, 1 domain expert) to evaluate generated questions quality: *fluency* (how fluent is the generated question), *adequacy* (how close are the reference and the generated question) and *understandability* (how understandable is the generated question) (Callison-Burch et al., 2006). Annotators followed a shared guideline defining adequacy, fluency and understandability, with ties allowed. For the three of them, a scale from 1 to 5 is asked for, where the higher the value, the better the text is with respect to such metric. These evaluations have been made individually, and no information about the model has been provided (blinded revision).

We merged the mean of each annotator for each model and each metric and summarized it in the Table 3. The results show that for fluency and understandability, the metrics that depend on the quality of the generation, the gold standard and the DivAC-LLM have similar scores, whereas DivAC-LLM matches or slightly exceeds reference adequacy, reflecting normalization consistency rather than superior clinical judgment.

Additionally, two bio-medical experts (1 male, 1 female both Spanish and between 30 and 40) were in charge of detecting which questions were generated and which were the original reference questions. In Table 4 we can observe that they are unable to detect which of the questions belong to the reference standard and which to Llama 3.3, proving how hard it is to detect generated content for a non-computing linguist. Finally, these two annotators were asked to choose between DivAC-LLM and their own ground truth. In 69.8% of the cases, they chose DivAC-LLM.

Three annotators (two computational linguists and one biomedical expert) rated all generated questions for adequacy, understanding, and fluency. Table 5 reports standard inter-annotator agreement (Fleiss' κ, Krippendorff's α, Scott's π) (Artstein and Poesio, 2008) together with a Wilcoxon dominance score (W) (Dror et al., 2018). The W score summarises all pairwise one-sided Wilcoxon tests as the number of significant wins minus losses for each model, providing a single comparative indicator that is not affected by chance-correction artifacts.

Agreement values are heterogeneous across systems: encoder-decoder baselines show moderate agreement, whereas LLMs display low or negative $\kappa/\alpha/\pi$, reflecting the instability of chance-corrected measures under skewed or near-ceiling rating distributions. The Wilcoxon scores offer a clearer picture: DivAC-LLM and ChatGPT obtain the highest W values across all three dimensions, while Pegasus underperforms with negative scores. Mid-range models (BioBART, BERT2BERT, FLAN-T5, ChatGPT-lite, Llama3.3) show small positive or negative values, indicating mixed performance.

4.2.3 QGEval evaluation

To validate DivAC-LLM, instead of relying on traditional machine translation metrics such as BLEU (Papineni et al., 2001), ROUGE (Lin, 2004), or BERT-score (Zhang et al., 2020), we used the QGEval benchmark. This open source framework brings together more than a dozen evaluation met-

Model	Annotator 1			Annotator 2			Annotator 3			Mean		
	Adeq.	Underst.	Flu.	Adeq.	Underst.	Flu.	Adeq.	Underst.	Flu.	Adeq.	Underst.	Flu.
biobart	4.01	4.48	4.47	3.87	4.47	4.47	3.81	4.25	4.14	3.90	4.40	4.36
BERT2BERT	3.44	4.18	4.15	3.32	4.17	4.21	3.40	3.74	3.72	3.38	4.03	4.03
FLAN-T5	3.83	4.77	4.68	3.60	4.70	4.70	3.93	4.50	3.88	3.79	4.66	4.42
pegasus-xsum	2.86	3.46	3.41	2.76	3.45	3.46	2.50	2.99	2.40	2.71	3.30	3.09
ChatGPT	4.28	4.99	4.99	3.83	**5.00**	**5.00**	4.67	4.95	4.88	4.26	4.98	4.96
ChatGPT-lite	4.59	4.99	4.97	4.18	**5.00**	**5.00**	4.91	**5.00**	4.95	4.56	**5.00**	**4.97**
Llama3.3	4.28	**5.00**	**5.00**	3.99	**5.00**	**5.00**	4.58	4.68	4.60	4.28	4.89	4.87
DivAC-LLM	**4.71**	4.98	4.92	**4.87**	4.94	**5.00**	**4.93**	4.98	4.94	**4.84**	4.97	4.95
human	4.54	4.98	4.94	4.83	4.97	4.98	4.74	4.99	**5.00**	4.70	4.98	**4.97**

Table 3: Human evaluation scores for adequacy, understanding, and fluency from three annotators (Annotator 1-3) and their mean.

Model	Percentage (%)
FLAN-T5	6.4
ChatGPT	11.1
BERT2BERT	13.4
bioBART	13.4
ChatGPT lite	14.6
Llama3.3	**20.5**
Manual labeling	**20.5**

Table 4: Percentage of times each model was selected as the best system.

rics, ranging from the aforementioned ones to others such as QRelScore (Wang et al., 2022) (which combines word-level matching with sentence-level generation) and UniEval (Li et al., 2025) (a set of fine-grained evaluation tags designed to better correlate with human judgments). A detailed description of these metrics is provided in Tables 13 and 14 in Appendix D. Table 6 shows the results of the metrics for the CTQG dataset. Our method outperforms all the other baselines in both lexical and semantic metrics. We also made an ablation study by removing the question division and then AutoCriteria modules. It is interesting to notice that a fine-tuned Llama 3.3 works best with both modules and then with none, so we hypothesize there could be some dependency between them. To confirm the importance of question division for this task, we also compared our method against an oracle, where the LLM knows the real number of questions to be generated. The oracle outperforms all the other configurations but the difference is not as big with DivAC-LLM, showing that our module is well trained.

Regarding QG-SQuAD, as there are no criteria, we combined the question division module with different LLMs and compared the results with fine-tuned transformers. Table 7 shows the results for the QG-SQuAD dataset. In this case, the question division module improves LLMs models, but the fine-tuned models outperform the LLMs for ROUGE and, more surprisingly, for BERTscore, proving that the question division is critical for clinical trials, but not generalizable for all question generation tasks.

5 Discussion

Our results support three observations. First, conditioning prompts with predicted question counts and extracted criterion values improves CTQG generation across lexical and semantic metrics, and better matches expert phrasing than bare LLM prompting. Second, separating division, extraction, and generation yields more stable outputs than end-to-end prompting, especially for multi-clause criteria. Third, precision-oriented division reduces multi-question splits and hallucinations.

The comparison with QG-SQuAD clarifies scope. Because its "criteria" are generic sentences and alignment is position-based, division accuracy has limited semantic impact, and fine-tuned seq2seq baselines remain competitive on ROUGE and BERTScore. In contrast, CTQG benefits from explicit domain cues (entities, thresholds, exception lists), which allow rule- and feature-based models to anchor generation.

Human evaluation mirrors these trends: fluency and understandability vary little across strong systems, while adequacy is more sensitive to prompt conditioning and explicit metadata. Inter-annotator agreement is moderate, consistent with prior subjective NLG evaluation. While the questionnaire format improves clarity, errors may still affect eligibility decisions. DivAC-LLM mitigates this risk via rule-guided prompting and explicit metadata, but remains a decision-support tool requiring clinical supervision.

Finally, the pipeline ablation suggests that LLMs

Model	Adequacy				Understanding				Fluency			
	Fleiss κ	Kripp. α	Scott's π	W	Fleiss κ	Kripp. α	Scott's π	W	Fleiss κ	Kripp. α	Scott's π	W
biobart	0.326	0.318	0.316	**-1**	0.503	0.498	0.496	**+1**	0.471	0.466	0.463	**+1**
BERT2BERT	0.333	0.321	0.318	**-1**	0.457	0.447	0.445	**+1**	0.465	0.455	0.453	**+1**
FLAN-T5	0.255	0.245	0.242	**-1**	0.384	0.381	0.378	**+1**	0.156	0.134	0.130	**+1**
pegasus-xsum	0.308	0.281	0.278	**-7**	0.514	0.488	0.486	**-7**	0.355	0.323	0.321	**-7**
ChatGPT	0.199	0.163	0.160	**+4**	0.163	0.162	0.159	**+4**	0.055	0.045	0.041	**+5**
ChatGPT-lite	0.092	0.043	0.039	**-1**	1.000	1.000	1.000	**+1**	0.192	0.191	0.188	**+1**
Llama3.3	0.225	0.202	0.199	**-1**	1.000	-0.017	-0.021	**+1**	1.000	-0.022	-0.026	**+1**
human	0.220	0.214	0.211	**NA**	0.149	0.150	0.147	**NA**	0.071	0.068	0.065	**NA**
DivAC-LLM	0.015	-0.299	-0.304	**+6**	-0.001	-0.434	-0.439	**+4**	1.000	-0.434	-0.439	**+5**

Table 5: Inter-annotator agreement (Fleiss' κ, Krippendorff's α, Scott's π) and Wilcoxon signed-rank dominance score (W). W is the number of significant wins minus losses for each model against all others for that dimension.

metrics	QRelScore	BERTScore	ROUGE-L	Q-BLEU4
bioBART	0.363	**0.772**	**0.721**	0.462
BERT2BERT	0.393	0.696	0.438	**0.523**
Flan-t5	0.658	0.710	0.471	0.443
ChatGPT	0.658	0.710	0.471	0.443
Llama3.3	**0.670**	0.709	0.475	0.456
Autocriteria + Llama3.3	0.745	0.891	0.399	0.395
Question division + Llama3.3	0.742	0.893	0.408	0.410
Fine tuned Llama3.3	0.710	0.944	0.694	0.618
DivAC-LLM	**0.746**	**0.945**	**0.758**	**0.682**
DivAC-LLM oracle mode (no autocriteria)	0.747	**0.956**	0.750	**0.696**
divAC-LLM oracle mode	**0.752**	0.953	0.722	0.653

Table 6: Automatic evaluation results on CTQG with respect to different models and combinations compared with DivAC-LLM.

Model	BERTScore	BLEU	ROUGE
Gemini	0.868	0.149	2.27
Gemini + div	0.855	0.209	2.98
ChatGPT	0.868	0.153	2.19
ChatGPT + div	0.874	**0.210**	3.07
Llama3.3	0.869	0.177	2.37
Llama3.3 + div	0.884	0.170	3.17
ProphetNet-large	0.888	0.143	**3.41**
Pegasus-XSum	**0.898**	0.173	3.21

Table 7: Automatic evaluation results on QG-SQuAD comparing fine-tuned models and LLMs with question division module.

are strong surface realizers but weak controllers without explicit inputs. Lightweight predictors provide control signals that keep generation within clinically useful bounds.

6 Future work

In future work, we will replace the static voting mechanism with a learned calibration layer to improve per-criterion decision reliability, following recent work on expectation consistency calibration for neural networks (Clarté et al., 2023). We also plan to enrich criterion extraction with structured representations and apply constrained decoding to enforce template-level consistency, extending recent findings in controlled text generation (Hokamp and Liu, 2017). Finally, we will integrate clinician feedback in a human-in-the-loop workflow so that their edits continuously fine-tune both division and generation modules, aligning with current advances in active learning for foundation models (Wan et al., 2023).

7 Conclusion

We introduced DivAC-LLM, a hybrid pipeline for generating physician-facing questionnaires from clinical trial criteria. By combining rule-based signals and classical models with LLM prompting, and by explicitly injecting extracted values, the system improves automatic metrics on CTQG and yields human judgments comparable to strong baselines for fluency and understandability while improving adequacy. The findings indicate that lightweight structure and prompt conditioning compensate for limitations of end-to-end prompting in clinical settings where multi-clause logic and exceptions are common.

Limitations

Certain limitations can be identified in this approach:

- **Dataset scope.** The dataset is limited to English-language lung cancer trials, which narrows both linguistic and clinical variability. Broader multilingual evaluation is needed to ensure robustness across registries and pathologies (Neumann et al., 2024).

- **Human evaluation scale.** Expert assessment was conducted by only three annotators within one specialty, potentially biasing adequacy and fluency judgments. Larger collaborative annotation efforts such as MedMCQA illustrate the benefits of scaling expert input (Pal et al., 2022a).

- **Model transparency.** Although DivAC-LLM improves interpretability through explicit intermediate steps, it still depends on opaque LLM components. Reliability and calibration remain open challenges before any clinical deployment.

- **External validation.** No downstream testing has yet been performed on real patient screening tasks, leaving practical effectiveness and safety unverified.

- **Energy consumption** Although the efforts were optimized in order to avoid a huge energy consumption, the training and fine-tuning periods were long enough to be costly and energy-greedy, which has to be improved in order to be more sustainable.

- **Why not fine-tuning an LLM to do everything?** Given the size of the dataset and its high-quality expert-based annotation, a few-shot end-to-end supervised fine-tuning of a small Llama could have been used to enrich the benchmark. However, as we have very short inputs (and outputs), we prioritize precision over recall (the "I always reply something" from the LLM can decrease precision). Furthermore, the energy consumption would be even bigger, so we decided to skip this experiment.

References

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

David Blumenthal and Marilyn Tavenner. 2010. The "meaningful use" regulation for electronic health records. *N. Engl. J. Med.*, 363(6):501–504.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. 2024. Biomedlm: A 2.7b parameter language model trained on biomedical text. *Preprint*, arXiv:2403.18421.

Mette Brøgger-Mikkelsen, Zarqa Ali, John R Zibert, Anders Daniel Andersen, and Simon Francis Thomsen. 2020. Online patient recruitment in clinical trials: Systematic review and meta-analysis. *J. Med. Internet Res.*, 22(11):e22179.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

François Clarté, Yann Chevaleyre, Guillaume Contardo, and Massimiliano Tommasi. 2023. Expectation consistency for calibration of neural networks. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 216 of *Proceedings of Machine Learning Research*, pages 5431–5450.

Surabhi Datta, Kyeryoung Lee, Hunki Paek, Frank J Manion, Nneka Ofoegbu, Jingcheng Du, Ying Li, Liang-Chin Huang, Jingqi Wang, Bin Lin, Hua Xu, and Xiaoyan Wang. 2024. AutoCriteria: a generalizable clinical trial eligibility criteria extraction system powered by large language models. *J. Am. Med. Inform. Assoc.*, 31(2):375–385.

Nicholas J Dobbins, Bin Han, Weipeng Zhou, Kristine F Lan, H Nina Kim, Robert Harrington, Özlem Uzuner, and Meliha Yetisgen. 2023. LeafAI: query generator for clinical cohort discovery rivaling a human programmer. *J. Am. Med. Inform. Assoc.*, 30(12):1954–1964.

Nicholas J. Dobbins, Tony Mullen, Özlem Uzuner, and Meliha Yetisgen. 2022. The leaf clinical trials corpus: a new resource for query generation from clinical trial eligibility criteria. *Scientific Data*, 9(1):490.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing.

In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Weiping Fu, Bifan Wei, Jianxiang Hu, Zhongmin Cai, and Jun Liu. 2024. QGEval: Benchmarking multi-dimensional evaluation for question generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11783–11803, Miami, Florida, USA. Association for Computational Linguistics.

Álvaro García-Barragán, Ahmad Sakor, Maria-Esther Vidal, Ernestina Menasalvas, Juan Cristobal Sanchez Gonzalez, Mariano Provencio, and Víctor Robles. 2025. Nssc: a neuro-symbolic ai system for enhancing accuracy of named entity recognition and linking from oncologic clinical notes. *Medical & Biological Engineering & Computing*, 63(3):749–772.

Joseph Gatto, Parker Seegmiller, Timothy Burdick, Inas S. Khayal, Sarah DeLozier, and Sarah M. Preum. 2025. Follow-up question generation for enhanced patient-provider conversations. *Preprint*, arXiv:2503.17509.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shashi Gupta, Aditya Basu, Mauro Nievas, Jerrin Thomas, Nathan Wolfrath, Adhitya Ramamurthi, Bradley Taylor, Anai N Kothari, Regina Schwind, Therica M Miller, Sorena Nadaf-Rahrov, Yanshan Wang, and Hrituraj Singh. 2024. PRISM: Patient records interpretation for semantic clinical trial matching system using large language models. *NPJ Digit. Med.*, 7(1):305.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Qiao Jin, Zifeng Wang, Charalampos S. Floudas, Fangyuan Chen, Changlin Gong, Dara Bracken-Clarke, Elisabetta Xue, Yifan Yang, Jimeng Sun, and Zhiyong Lu. 2024. Matching patients to clinical trials with large language models. *Nature Communications*, 15(1):9074.

Tian Kang, Shaodian Zhang, Youlan Tang, Gregory W. Hruby, Alexander Rusanov, and Noémie Elhadad. 2017. EliIE: An open-source information extraction system for clinical trial eligibility criteria. *J. Am. Med. Inform. Assoc.*, 24(6):1062–1071.

Samuel Kaskovich, Kirk D Wyatt, Tomasz Oliwa, Luca Graglia, Brian Furner, Jooho Lee, Anoop Mayampurath, and Samuel L Volchenboum. 2023. Automated matching of patients to clinical trials: A patient-centric natural language processing approach for pediatric leukemia. *JCO Clin. Cancer Inform.*, 7:e2300009.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Fabrício Kury, Alex Butler, Chi Yuan, Li-heng Fu, Yingcheng Sun, Hao Liu, Ida Sim, Simona Carini, and Chunhua Weng. 2020. Chia, a large annotated corpus of clinical trial eligibility criteria. *Scientific Data*, 7(1):281.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Songda Li, Yunqi Zhang, Chunyuan Deng, Yake Niu, and Hui Zhao. 2024. Better late than never: Model-agnostic hallucination post-processing framework towards clinical text summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 995–1011, Bangkok, Thailand. Association for Computational Linguistics.

Yi Li, Haonan Wang, Qixiang Zhang, Boyu Xiao, Chenchang Hu, Hualiang Wang, and Xiaomeng Li. 2025. Unieval: Unified holistic evaluation for unified multimodal understanding and generation. *Preprint*, arXiv:2505.10483.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. *Annu Meet Assoc Comput Linguistics*, pages 74–81.

Cong Liu, Chi Yuan, Alex M Butler, Richard D Carvajal, Ziran Ryan Li, Casey N Ta, and Chunhua Weng. 2019. DQueST: dynamic questionnaire for search of clinical trials. *J. Am. Med. Inform. Assoc.*, 26(11):1333–1343.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Curtis L Meinert. 2012. *ClinicalTrials: design, conduct and analysis*, volume 39. OUP USA.

Johannes Moll, Louisa Fay, Asfandyar Azhar, Sophie Ostmeier, Tim Lueth, Sergios Gatidis, Curtis Langlotz, and Jean-Benoit Delbrouck. 2025. Structuring

radiology reports: Challenging llms with lightweight models. *arXiv preprint arXiv:2506.00200*.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Mark Neumann, Emily Alsentzer, Alistair Johnson, and Abhinav Singh. 2024. Medalign: Benchmarking foundation models for multilingual clinical text understanding. *Journal of Biomedical Informatics*, 152:104512.

Ankit Pal, Logesh Umapathi, and Malaikannan Sankarasubbu. 2022a. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. *NeurIPS Datasets and Benchmarks Track*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022b. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Jimyung Park, Yilu Fang, Casey Ta, Gongbo Zhang, Betina Idnay, Fangyi Chen, David Feng, Rebecca Shyu, Emily R. Gordon, Matthew Spotnitz, and Chunhua Weng. 2024a. Criteria2query 3.0: Leveraging generative large language models for clinical trial eligibility query generation. *Journal of Biomedical Informatics*, 154:104649.

Jimyung Park, Yilu Fang, Casey Ta, Gongbo Zhang, Betina Idnay, Fangyi Chen, David Feng, Rebecca Shyu, Emily R Gordon, Matthew Spotnitz, and Chunhua Weng. 2024b. Criteria2Query 3.0: Leveraging generative large language models for clinical trial eligibility query generation. *J. Biomed. Inform.*, 154(104649):104649.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Preethi Raghavan, James L Chen, Eric Fosler-Lussier, and Albert M Lai. 2014. How essential are unstructured clinical narratives and information fusion to clinical trial recruitment? *AMIA Summits Transl. Sci. Proc.*, 2014:218–223.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jianlin Shi and Brian T. Bucher. 2025. Initial investigation of llm-assisted development of rule-based clinical nlp system. *Preprint*, arXiv:2506.16628.

Veronika Smilga and Hazem Alabiad. 2024. TüDuo at SemEval-2024 task 2: Flan-t5 and data augmentation for biomedical NLI. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 737–744, Mexico City, Mexico. Association for Computational Linguistics.

Andrew L Smith, Felix Greaves, and Trishan Panch. 2023. Hallucination or confabulation? neuroanatomy as metaphor in large language models. *PLOS Digit. Health*, 2(11):e0000388.

Paul Trust and Rosane Minghim. 2024. A study on text classification in the age of large language models. *Machine Learning and Knowledge Extraction*, 6(4):2688–2721.

Shuo Wan, Tianyang Gao, Yichi Zhang, and Zhen Wang. 2023. A survey of deep active learning for foundation models. *Intelligent Computing*, 2(2):0058.

Xiaoqiang Wang, Bang Liu, Siliang Tang, and Lingfei Wu. 2022. QRelScore: Better evaluating generated questions with deeper understanding of context-aware relevance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 562–581, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

L L Weed. 1968. Medical records that guide and teach. *N. Engl. J. Med.*, 278(11):593–600.

Chunhua Weng, Xiaoying Wu, Zhihui Luo, Mary Regina Boland, Dimitri Theodoratos, and Stephen B Johnson. 2011. EliXR: an approach to eligibility criteria extraction and representation. *J. Am. Med. Inform. Assoc.*, 18 Suppl 1(Supplement 1):i116–24.

Michael Wornow, Alejandro Lozano, Dev Dash, Jenelle Jindal, Kenneth W Mahaffey, and Nigam H Shah. 2025. Zero-shot clinical trial patient matching with llms. *NEJM AI*, 2(1):AIcs2400360.

Xiaoyang Yi, Yuru Bao, Jian Zhang, Yifang Qin, and Faxin Lin. 2024. Integrating structural semantic knowledge for enhanced information extraction pretraining. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*,

pages 2156–2171, Miami, Florida, USA. Association for Computational Linguistics.

Chi Yuan, Patrick B. Ryan, Casey Ta, Yixuan Guo, Ziran Li, Jill Hardin, Rupa Makadia, Peng Jin, Ning Shang, Tian Kang, and Chunhua Weng. 2019. Criteria2Query: a natural language interface to clinical databases for cohort definition. *J. Am. Med. Inform. Assoc.*, 26(4):294–305.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022. BioBART: Pretraining and evaluation of a biomedical generative language model. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 97–109, Dublin, Ireland. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint*, arXiv:1912.08777.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Angelo Ziletti and Leonardo D'Ambrosi. 2025. Generating patient cohorts from electronic health records using two-step retrieval-augmented text-to-sql generation. *Preprint*, arXiv:2502.21107.

A Appendix

A.1 Prompts

A.1.1 CTQG initial dataset generation

I'll give you (one by one) a set of clinical trial inclusion/exclusion criteria and I need you to turn them into a question, as if in a form. e.g. for criteria "patient is > 18" the question would be "is the patient > 18?"

A.1.2 Question generation for Clinical Trials

You are a clinical assistant specialized in communicating eligibility criteria for clinical trials. Your task is to transform inclusion or exclusion criteria into understandable questions directed at patients in English, while maintaining both clinical accuracy and clarity of language.
Additionally, you must briefly reason out loud before generating each question, explaining step by step how you break down the criterion, what terms you identify, and how you decide to formulate the questions. This reasoning must be clear, logical, and concise, and should be included just before each group of questions.
Objective: Convert each criterion into one or more clear and understandable questions in English, without losing medical rigor.
Important instructions (read carefully and follow all of them):

1. *Analyze each criterion carefully: if the same criterion contains multiple clinical conditions, diseases, values, histories, or options, you must generate a separate question for each of those elements. Do not combine multiple conditions into a single question.*
2. *Before generating the questions, write a brief explanation of your reasoning in English (chain of thought), identifying the relevant parts of the clinical criterion and justifying how you are going to split them.*
3. *Always use the following format for your answers:*
   - *"Reasoning: [your step-by-step explanation in English]"*
   - *"New question N: [generated question]"*
   - *If there is more than one question for that criterion: "New question N: [question 1] || [question 2] || [question 3] ..."*
4. *Keep the original abbreviations (e.g., VIH, BMI, COPD). Do not explain or expand them.*
5. *Do not rewrite or interpret the criteria. Only transform them into questions.*
6. *Always generate the questions in English.*

Reasoning example:
Criterion: Age ≥ 18 years old Reasoning: The criterion refers to age, with a threshold of 18 years. It is a single condition related to minimum age. New question 1: Is the patient ≥18 years old?
Other examples without reasoning:
Example 1:
Criterion: Age ≥ 18 years old
Question: Is the patient ≥18 years old?
Example 2:

...

A.1.3 Question generation for SQuAD

You are a documentalist, expert in converting sentences into questions. Your task is to extract from the original text understandable questions in English which answer is included in the same text, while maintaining both accuracy and clarity of language.
Additionally, you must briefly reason out loud before generating each question, explaining step by step how you break down the text, what terms you identify, and how you decide to formulate the questions. This reasoning must be clear, logical, and concise, and should be included just before each group of questions.
Objective: Convert each sentence into one or more clear and understandable questions in English, without losing rigor.
Important instructions (read carefully and follow all of them):

1. *Analyze each text carefully: if the same text contains multiple relevant questions, you must generate them. If you consider that there is no relevant question to be generated, simply write "Unknown"*
2. *Before generating the questions, write a brief explanation of your reasoning in English (chain of thought), identifying the relevant parts of the text and justifying how you are going to split them.*
3. *Always use the following format for your answers:*
   - *"Reasoning: [your step-by-step explanation in English]"*
   - *"New question N: [generated question]"*
   - *If there is more than one question for that criterion: "New question N: [question 1] || [question 2] || [question 3] ..."*
4. *Keep the original abbreviations (e.g., VIH, BMI, COPD). Do not explain or expand them.*

5. *Do not rewrite or interpret the text. Only transform them into questions.*

6. *Always generate the questions in English.*

Remember: if necessary, you should generate various questions separated by "?", and avoid including new lines in the questions and between them if many are generated
Example 1:
Criterion: Cultural reactions have varied from ridicule to admiration; many common stereotypes exist regarding redheads and they are often portrayed as fiery-tempered.
Question: What type of temper are people with red hair considered to have?
Example 2:

...

B Annotation guidelines for criterion-to-question mapping

Unit of annotation The primary unit is an *atomic clinical condition* expressed in an eligibility criterion (e.g., age constraint, ECOG status, prior treatment, specific comorbidity, organ function, pregnancy/contraception). A single textual criterion may yield one or multiple questions if it contains several independent conditions or clinically relevant exceptions.

B.1 Inputs and annotator setup

- **Source documents:** full inclusion/exclusion sections of oncology trial protocols.
- **Annotator:** one medically literate annotator, working alone (no adjudication, no inter-annotator agreement).
- **LLM assistance:** ChatGPT is used to draft questions from criteria; the annotator then manually edits the output to follow the rules in this section.
- **Effort:** typical time per question is a few seconds, with a single prompt per batch of criteria (no iterative prompt engineering per item).

B.2 Question generation rules

B.2.1 General phrasing

1. **Patient-centred formulation:** questions are formulated about the patient, not about abstract "participants". Typical patterns:
   - "Is the patient . . . ?"
   - "Does the patient have . . . ?"
   - "Has the patient . . . ?"
2. **Binary questions by default:** questions are yes/no. The logical polarity (inclusion vs. exclusion) is not encoded in the wording. For example, both "patients with X are excluded" and "patients must have X" yield questions of the form "Does the patient have X?"
3. **Preserve clinical content and terminology:** clinical constructs (e.g., ECOG, TNM, PD-L1, drug names, numeric thresholds) are preserved verbatim. Acronyms and units are not simplified or explained.
4. **Abstract away protocol management:** protocol-management phrasing ("must be", "is required", "prior to enrolment") is removed when it does not alter the clinical condition. Time constraints that affect eligibility are kept (e.g., "within N days before treatment").

B.2.2 Numbers, thresholds and comparators

5. **Preserve numeric operators and units:** inequalities ($\geq$, $\leq$, $>$, $<$) are kept as symbols in the question text. Units (g/dL, $\times$ULN, mL/min, mmHg, months, weeks, years, etc.) are preserved. Example pattern:

   > Criterion: "Age $\geq$ X years" $\rightarrow$ Question: "Is the patient $\geq X$ years old?"

6. **Multi-constraint tests:** when a criterion aggregates several lab thresholds or organ function conditions that jointly define a single concept (e.g., "adequate organ function"), a single question is created:

   > "Does the patient meet the criteria for adequate [organ] function (e.g., [list of thresholds])?"

   If thresholds appear elsewhere as independent criteria, they may be annotated separately.

B.2.3 Splitting criteria into multiple questions

7. **One atomic condition $\rightarrow$ one question:** if a criterion includes several independent clauses (e.g., age + performance status + disease stage), these are split into separate questions, one per atomic condition.
8. **Exceptions and complex logic:** for patterns of the form "X except Y and Z":

- At least one question captures the broad condition (e.g., "Does the patient have any previous or current malignancy?").
- Additional questions may be created for clinically important exceptions when they are meaningful variables (e.g., major cardiovascular disease).
- Benign exceptions (e.g., vitiligo, alopecia) are not systematically turned into separate questions; they remain implicit in the logic.

B.2.4 Multi-choice vs. yes/no questions

9. **Yes/No for presence/absence:** conditions that are naturally binary (pregnancy, presence of a comorbidity, history of a given treatment, presence of brain metastases, etc.) are encoded as yes/no questions.

10. **List when categories are enumerated:** if a criterion explicitly enumerates mutually exclusive categories (e.g., tumour types, histologic subtypes, study cohorts), it is encoded as a single multi-choice question:

    > "Which of the following applies to the patient? (1) . . . , (2) . . . , (3) . . . "

    The corresponding answer type is a list. Follow-up criteria that merely continue the list are not considered separate items.

B.2.5 Target population

11. **Question population:** for each question, a separate field encodes the subpopulation for which the question is meaningful, e.g.:

    - "All patients"
    - "Women of childbearing potential"
    - "Male participants"
    - "Only if [brain metastases = yes]"

    Optionally, population restrictions may be repeated in the question text as a leading clause (e.g., "For women of childbearing potential: . . . ").

B.3 Summary of the annotation process

All questions are created by a single medically literate annotator using ChatGPT as a drafting tool, followed by manual post-editing to enforce the guidelines above. Each criterion is processed once (no double annotation, no adjudication), and the order of questions within each trial follows the original ordering of criteria. When a criterion yields multiple questions, these appear immediately after the parent in the internal sheet.

C Question division experimental details

C.1 Data preparation details

- CTQG: there is a large imbalance in classes, where class 1 is over-represented compared to the other two (Table 8).

Class	Description	Proportion
0	criterion generates no question	0.196
1	criterion generates one question	0.647
2	criterion generates several questions	0.157

Table 8: Data class support for CTQG question division prediction.

- QG-SQuAD: there is more balance than in CTQG, but this time there is one class that is under-represented, class 2 (Table 9).

Class	Description	Proportion
0	criterion generates no question	0.405
1	criterion generates one question	0.405
2	criterion generates several questions	0.190

Table 9: Data class support for QG-SQuAD question division prediction.

C.2 Rules sample

The rules we extracted from the CTQG features are available at Table 10 and the ones for QG-SQuAD at Table 11.

Algorithm 2 Question division algorithm

Require: Labeled criterion $\{(x_i, y_i)\}$
1: Extract features $\Phi(x_i)$
2: Fit/keep high-precision rules $\mathcal{R}$
3: Train RandomForest on $(\Phi(x_i), y_i)$
4: Compute embeddings $E(x_i)$
5: Fine-tune transformer models $\{f_m\}$ on $(E(x_i), y_i)$

D Complementary results

Tables 13 and 14 present the complementary results of Tables 6 and 7, respectively, with additional metrics introduced section 4.2.3.

E Complementary information

In this last Appendix, we include all details on the methods we used (Tables 15, 16) or on the experimental set-up (Tables 17, 19, 18, 20).

Rule condition	Class	Precision	Support
raw_word_len $\geq$ 36 AND shannon_char_entropy $\leq$ 4.1787	2	1.00	5
raw_word_len $\geq$ 39 AND num_clauses $\leq$ 4	2	1.00	5
raw_word_len $\leq$ 2 AND num_words $\leq$ 2	0	1.00	5
raw_char_len $\leq$ 179 AND raw_char_2grams_voc $\geq$ 113	2	1.00	5
raw_char_len $\geq$ 263 AND raw_char_2grams_voc $\leq$ 142	2	1.00	5
raw_char_len $\geq$ 216 AND raw_char_3grams_voc $\leq$ 174	2	1.00	5
raw_char_len $\geq$ 216 AND raw_char_4grams_voc $\leq$ 194	2	1.00	6
raw_char_len $\geq$ 273 AND raw_word_2grams_voc_lemma $\leq$ 38	2	1.00	5
raw_char_len $\leq$ 188 AND raw_word_2grams_voc_pos $\geq$ 15	2	1.00	5
raw_char_len $\geq$ 271 AND num_clauses $\leq$ 4	2	1.00	5
raw_char_len $\leq$ 12 AND num_words $\leq$ 2	0	1.00	6
raw_voc_len $\geq$ 28 AND shannon_char_entropy $\leq$ 4.1705	2	1.00	5
raw_voc_len $\leq$ 2 AND num_words $\leq$ 2	0	1.00	5
raw_voc_len $\leq$ 33 AND num_words $\geq$ 47	2	1.00	5

Table 10: Validated rule set for the main classifier. Each rule is defined by a condition, target class, precision, and support.

Algorithm 3 Inference and voting (precision-prioritized

Require: Criterion x, rules $\mathcal{R}$, RF, models $\{f_m\}$, thresholds $\tau_r, \tau_{ft}, \tau_{maj}$

Compute $\Phi(x)$ and $E(x)$
If a rule in $\mathcal{R}$ fires with confidence $\geq \tau_r$, **return** class
Get RF prediction (y_{rf}, c_{rf})
Get FT predictions $\{(y_m, c_m)\}$ and majority y_{maj} with support p_{maj} and min confidence C_{maj}
if $p_{maj} \geq \tau_{maj}$ **and** $C_{maj} \geq \tau_{ft}$ **then return** y_{maj}
end if
if $c_{rf} \geq \tau_{ft}$ **then return** y_{rf}
end if
return IDK

Rule condition	Class	Precision	Support
raw_char_2grams_voc $\leq$ 0 AND shannon_word_entropy $\geq$ 4	0	0.90	10
raw_voc_len $\geq$ 5 AND raw_word_2grams_voc_pos $\leq$ 0	0	0.83	6
raw_char_len $\leq$ 2 AND raw_word_voc_pos $\geq$ 5	1	0.80	5
raw_char_4grams_voc $\geq$ 5 AND raw_word_2grams_voc_pos $\leq$ 0	0	0.80	5
raw_char_4grams_voc $\leq$ 0 AND shannon_word_entropy $\geq$ 3	0	0.78	36
raw_char_len $\geq$ 5 AND raw_word_2grams_voc_pos $\leq$ 0	0	0.75	8
raw_char_2grams_voc $\leq$ 0 AND num_words $\geq$ 4	0	0.75	8
raw_char_len $\leq$ 0 AND shannon_word_entropy $\geq$ 3	0	0.74	42
raw_voc_len $\leq$ 2 AND raw_word_2grams_voc_pos $\geq$ 5	1	0.71	7

Table 11: Validated rule set derived from the SQuAD-style subset.

Hyperparameter	value
Activation function	Softmax
loss function	cross-entropy
dropout	0.1
class weighting	No
weight decay	0.000001
epochs	3
batch size	32
learning rate	0.000005
warm-up	1000 steps

Table 12: Best hyper parameters for transformers fine-tuning for question division prediction with QG-SQuAD.

Metrics (I)	QRelScore	METEOR	ROUGE-L	QSTS	QBLEU	BARTScore
BERT2BERT	0.34	0.32	0.47	0.69	0.38	-4.65
bioBART	0.43	0.39	0.52	**0.79**	0.41	-4.37
Llama3.3	0.68	0.44	0.47	**0.79**	0.46	-3.55
FLAN-T5	0.66	0.36	0.41	0.72	0.35	-3.83
ChatGPT	0.66	0.44	0.47	0.74	0.44	-3.49
DivAC-LLM	**0.74**	**0.53**	**0.56**	0.77	**0.52**	**-3.16**
Metrics (II)	**BARTScore**	**fluency**	**clarity**	**conciseness**	**relevance**	**consistency**
BERT2BERT	0.77	-5.22	0.88	0.76	0.77	0.78
bioBART	0.81	-4.86	0.88	0.80	0.81	0.82
Llama3.3	0.92	-3.60	0.89	0.92	0.92	0.92
FLAN-T5	0.86	-3.85	0.86	0.85	0.86	0.86
ChatGPT	0.91	-3.69	0.88	0.90	0.91	0.91
DivAC-LLM	**0.93**	**-3.38**	**0.91**	**0.92**	**0.93**	**0.92**
Metrics (III)	**answerability**	**acceptance**	**answer consistency**	**overall**	**mean**	**scaled**
BERT2BERT	0.77	0.78	0.75	0.78	0.11	0.32
bioBART	0.81	0.81	0.78	0.82	0.19	0.46
Llama3.3	0.93	0.91	0.84	0.91	0.36	0.71
FLAN-T5	0.87	0.85	0.80	0.85	0.29	0.58
ChatGPT	0.92	0.90	**0.85**	0.90	0.35	0.69
DivAC-LLM	**0.93**	**0.92**	0.83	**0.91**	**0.42**	**0.83**

Table 13: Extended automatic evaluation results on CTQG with respect to five different models compared with DivAC-LLM.

Metrics (I)	QRelScore	METEOR	ROUGE-L	QSTS	QBLEU	BARTScore
Gemini	0.46	0.23	0.23	**0.62**	0.15	-4.28
Gemini + div	0.35	0.24	0.30	0.58	**0.21**	-4.30
ChatGPT	**0.47**	0.22	0.22	**0.62**	0.15	-4.33
ChatGPT + div	0.33	0.24	0.31	0.59	**0.21**	-4.37
Llama3.3	0.32	0.17	0.24	0.52	0.18	-4.63
Llama3.3 + div	0.28	0.21	0.32	0.54	0.17	-4.47
ProphetNet-large	0.26	**0.28**	**0.34**	0.30	0.14	**-3.75**
Pegasus-XSum	0.37	0.24	0.32	0.36	0.17	-3.79
Metrics (II)	**BARTScore**	**fluency**	**clarity**	**conciseness**	**relevance**	**consistency**
Gemini	**-3.74**	0.81	0.89	0.90	0.90	0.90
Gemini + div	-4.28	0.78	0.82	0.82	0.84	0.82
ChatGPT	-3.77	0.85	**0.91**	0.91	**0.92**	0.91
ChatGPT + div	-4.35	0.77	0.79	0.80	0.82	0.80
Llama3.3	-4.56	**0.87**	0.87	0.87	0.88	0.87
Llama3.3 + div	-4.64	0.86	0.86	0.86	0.88	0.86
ProphetNet-large	-4.19	0.66	0.76	0.77	0.77	0.77
Pegasus-XSum	-4.21	**0.87**	0.90	**0.92**	**0.92**	**0.92**
Metrics (III)	**answerability**	**acceptance**	**answer consistency**	**overall**	**mean**	**scaled**
Gemini	0.90	0.89	0.73	0.86	0.19	0.62
Gemini + div	0.83	0.82	0.72	0.81	0.15	0.52
ChatGPT	0.91	**0.91**	0.75	0.88	0.19	0.65
ChatGPT + div	0.80	0.80	0.73	0.79	0.14	0.50
Llama3.3	0.88	0.88	0.69	0.85	0.11	0.47
Llama3.3 + div	0.86	0.87	0.73	0.85	0.14	0.56
ProphetNet-large	0.77	0.77	0.67	0.74	0.15	0.46
Pegasus-XSum	**0.92**	**0.91**	**0.76**	**0.89**	**0.22**	**0.81**

Table 14: Extended automatic evaluation results on QG-SQuAD comparing fine-tuned models and LLMs with and without question division module.

Feature	Mean value
AND or OR in text	0.52
has semicolon	0.03
has conjonction	0.57
number of clauses	2.49
number of words	18.32
word length	15.73
character length	108.58
vocab. length	13.82
word bigrams	14.44
character bigrams	68.92
word trigrams	13.66
character trigrams	89.26
word quadrigrams	12.72
character quadrigrams	94.41
word lemmatized	13.73
word bigrams_lemma	14.42
word trigrams_lemma	13.65
word quadrigrams_lemma	12.72
word POS	7.39
word bigrams POS	7.00
word trigrams POS	6.13
word quadrigrams POS	5.28
word NER	1.24
word bigrams NER	0.66
word trigrams NER	0.31
word quadrigrams NER	0.14
shannon character entropy	4.01
shannon word entropy	3.45
conditional word entropy	0.12
conditional character entropy	0.08
renyi word entropy	0.12
renyi character entropy	1.69

Table 15: Average feature values per criterion in CTQG.

Voting Strategy	Description / Decision Rule
`pred`	If both the rule-based model (`prediction`) and BioBERT agree on a class different from 2, use that class; otherwise, output 2.
`default1`	If both the rule-based model and BioBERT agree on a class different from 1, use that class; otherwise, output 1.
`majority_transformer`	If the two transformers (BioBERT and BioMedLM) agree, take their class; otherwise, fall back to the rule-based model.
`majority_rules`	If the rule-based model agrees with either transformer, use the rule-based prediction; otherwise, fall back to BioBERT.
`hybrid_priority`	Prioritize class 1 if any transformer predicts 1; prioritize class 2 if rules predict 2; otherwise, if rules and BioBERT agree, keep that; else 0.
`consensus_only`	Output the class only if all three models (rules, BioBERT, BioMedLM) fully agree; otherwise, label as -1 (uncertain).
`strict_negative`	Reject (-1) if all predict -1; if all agree otherwise, output that class; else fall back to BioBERT.
`2of3`	If any two of the three models agree, take that class; otherwise, label as -1 (no consensus).

Table 16: Overview of voting strategies combining rule-based predictions, BioBERT, and BioMedLM outputs.

Hyperparameter	Value
Activation function	Softmax
Loss function	Cross-entropy
Dropout	0.5
Class weighting	No
Weight decay	5e-6
Epochs	8
Batch size	4
Learning rate	1e-5
Warm-up	100 steps

Table 17: Best hyperparameters for transformer fine-tuning (question division, CTQG).

Metric	Value	Assumptions / Notes
Total utterances	92	
Total tokens	24,949	
Avg. tokens / utterance	271.200	Computed as $24{,}949/92$
GPU time	$0.153\,\mathrm{h}$ ($9.000\,\mathrm{min}$)	Inference only
Energy consumption	$0.046\,\mathrm{kWh}$	$0.153\,\mathrm{h} \times 0.300\,\mathrm{kW}$
CO_2 emissions	$9.200\,\mathrm{g}$	EU grid avg. $0.200\,\mathrm{kgCO2/kWh}$
Electricity cost	€0.009	€0.20/kWh (EU, 2024)
Cloud rental	€0.380	€2.50/GPUh (on-demand)

Table 18: Approximate compute and environmental footprint for 92 utterances (24,949 tokens) on one NVIDIA A100 40GB at $300.000\,\mathrm{W}$. Power and emissions factors from NVIDIA (2022) and IEA (2023).

Experiment	Duration	GPU h	Energy (kWh)	Elec. (€)	Cloud (€)	CO_2 (kg)
CTQG	∼2 weeks	350	105	21	875	21
QG-SQuAD	∼1 month	700	210	42	1750	42

Table 19: Runtime and cost estimates for grid search (question generation + division) on a single A100 40GB at $300.000\,\mathrm{W}$. Prices: €0.20/kWh (EU 2024), €2.50/GPUh (on-demand). Emissions: $0.200\,\mathrm{kgCO2/kWh}$.

Parameter	CTQG	QG-SQuAD
Batch size	16	32
Learning rate	0.0001	0.00001
Epochs	8	12
Warm-up steps	50	1500

Table 20: Hyperparameter configurations used in CTQG and QG-SQuAD experiments.

Hyperparameter	CTQG	QG-SQuAD
Epochs	[4, 6, 8, 10, 12]	[6, 8, 10, 12, 14]
Batch size	[16, 32, 64, 128]	[32, 64]
Learning rate	[5e^{-6}, 1e^{-6}, 1e^{-5}, 1e^{-4}, 5e^{-5}]	[5e^{-5}, 1e^{-5}]
Warm-up steps	[10, 50, 100]	[1k, 5k, 10k]

Table 21: Grid search hyperparameter ranges for CTQG and QG-SQuAD experiments.