Generative Models for Automatic Medical Decision Rule Extraction from Text

Yuxin He[1,3] **and Buzhou Tang**[*1,2] **and Xiaoling Wang**[4]
[1]Department of Computer Science, Harbin Institute of Technology (Shenzhen)
[2]Peng Cheng Laboratory
[3]The Hong University of Science and Technology (Guangzhou)
[4]East China Normal University
21S051047@stu.hit.edu.cn
tangbuzhou@gmail.com

Abstract

Medical decision rules play a key role in many clinical decision support systems (CDSS). However, these rules are conventionally constructed by medical experts, which is expensive and hard to scale up. In this study, we explore the automatic extraction of medical decision rules from text, leading to a solution to construct large-scale medical decision rules. We adopt a formulation of medical decision rules as binary trees consisting of condition/decision nodes. Such trees are referred to as medical decision trees and we introduce several generative models to extract them from text. The proposed models inherit the merit of two categories of successful natural language generation frameworks, i.e., sequence-to-sequence generation and autoregressive generation. To unleash the potential of pretrained language models, we design three styles of linearization (natural language, augmented natural language and JSON code), acting as the target sequence for our models. Our final system achieves 67% tree accuracy on a comprehensive Chinese benchmark, outperforming state-of-the-art baseline by 12%. The result demonstrates the effectiveness of generative models on explicitly modeling structural decision-making roadmaps, and shows great potential to boost the development of CDSS and explainable AI. Our code will be open-source upon acceptance.

1 Introduction

Currently, the development of clinical decision support systems (CDSS) relies heavily on manual enumeration of medical decision rules (Matsumura et al., 1986; Grosan et al., 2011; Shortliffe and Sepúlveda, 2018). Although this paradigm brings CDSS interpretability and reliability, its request of extensive labor poses a challenge on scaling, given the huge amount of potential medical decision rules (Tsumoto, 1998). And the fact that some medical decision rules get occasionally updated make the challenge even worse. This motivates researchers to explore the automation of medical decision rules construction. Inspired by the fact that human doctors acquire medical decision rules from textbooks and clinical guidelines, a recent study proposes to imitate this process via deep learning methods (Zhu et al., 2022).

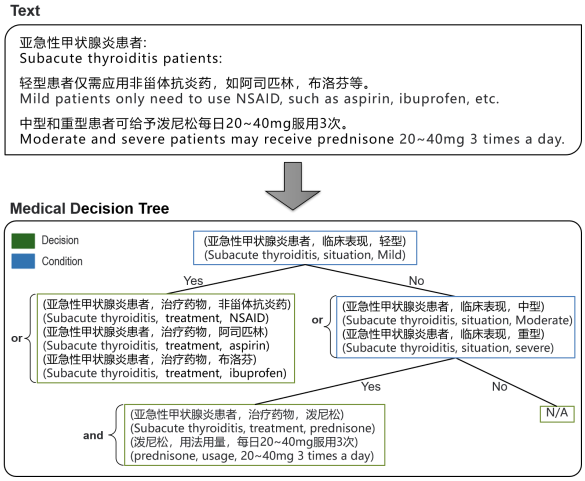


Figure 1: An example (translated from Chinese) of extracting tree-form medical decision rules from clinical guidelines and textbooks.

There exist two typical formulations of medical decision rules: first-order predicate logic formulas (Matsumura et al., 1986; Tsumoto, 1998) and medical decision trees (Zhu et al., 2022), where the latter is an extension of the former. Formally, a medical decision tree is a binary tree consisting of condition nodes and decision nodes. Each node is a relation triple or multiple relation triples combined by logical operators ("OR", "AND"). The decision nodes are leaf nodes of the tree, whereas the condition nodes are internal nodes. And the transition from one node to another represents judgment or decision-making. A first-order predicate logic formula in conjunctive normal form can be viewed as a special case of a medical decision tree where

[*]Corresponding Author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7034–7048
November 12-16, 2024

there is only one condition node and one decision node. Hence, we adopt the tree-form formulation in this paper.

Different from traditional information extraction tasks, e.g., name entity recognition (Tan et al., 2021; He and Tang, 2022), relation triple extraction (Yan et al., 2021; He and Tang, 2023) and event extraction (Yang et al., 2021; He et al., 2023), where the target output is a set of unitary/dual/multivariate tuples, the target output of medical decision tree extraction is a logically combined complex of relation triples. The logical coherence exhibited by such complexes mimics that of human language. This motivates us to adopt generative approaches for medical decision tree extraction, so as to better model the intrinsic logical connection among the relation triples inside a medical decision tree.

Reflecting on the exciting success within the field of natural language generation, we can observe that two paradigms (sequence-to-sequence, autoregressive generation) along with the idea of pretraining play the crucial roles. In this work, we try to replicate the success of sequence-to-sequence/autoregressive generation on the task of medical decision tree extraction.

In order to maximally elicit the potential of pretrained generative language models, three designs of medical decision tree linearization are trialed: 1) natural language (NL) style of linearization, where the relation triples are verbalized and naturally assembled with conjunctions; 2) augmented natural language (AugNL) style of linearization, where each relation triple is represented as an augmented token, sharing equal status with natural language tokens; 3) JSON style of linearization, the most widely used data interchange format that represents data objects as key–value pairs. The linearized medical decision trees act as the target sequences during training, and are generated then parsed into tree structure during inference.

The proposed sequence-to-sequence models employ an encoder-decoder architecture with a pair of pretrained language encoder and decoder, as well as a query-based entity-relation extractor. Under this paradigm, relation triple extraction is treated as a sub-task and the models fulfill it via the entity-relation extractor. Whereas the proposed autoregressive models are instantiated from decoder-only large language models (LLMs). In this discipline, relation triple extraction is treated as an auxiliary task for multi-task learning without introducing extra parameters.

Benchmarking on Text2DT (Zhu et al., 2022), a comprehensive Chinese dataset, we find that generative models are much more capable of extracting medical decision tree than state-of-the-art (SOTA) discriminative models. Our experiments also show that a carefully designed sequence-to-sequence model (Section 2.2) is competitive to a LLM-based autoregressive model (Section 2.3) that is 10+ times larger.

Our contributions are summarized as follows:

- We propose several generative models under the sequence-to-sequence/autoregressive paradigms to better capture the intrinsic logical connection among the relation triples within a medical decision tree and extract the tree from text accurately.
- We design 3 styles of tree linearization to represent each medical decision tree as a sequence that is suitable to be generated by different pretrained generative language models.
- Experimental results demonstrate that our method outperforms SOTA discriminative method by 12% tree accuracy, 9% path F1 score on the only available public benchmark, Text2DT. In-depth analysis also uncovers the pros and cons of different generative medical decision tree extraction models.

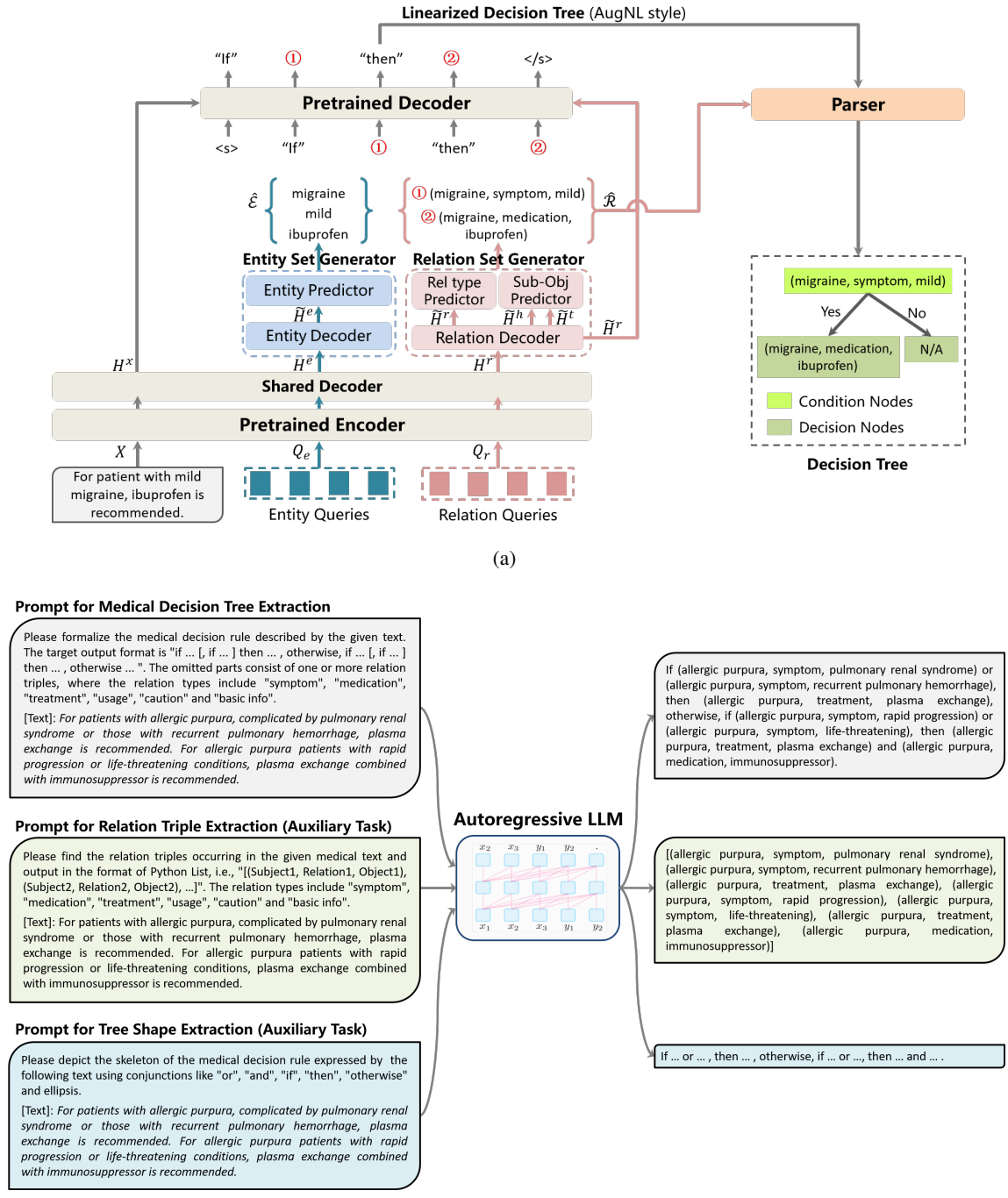
2 Methodology

2.1 Medical Decision Tree Linearization

To linearize medical decision trees into NL or AugNL style sequences as target output for training, we traverse each tree in pre-order, insert transition conjunctions (“if”, “else”, “then”, “otherwise”) between nodes according to the node position, and join the relation triples within each node with logical conjunctions (“or”, “and”). This procedure is depicted in Algorithm 1. The specific differences between NL and AugNL styles are explained in Section 2.2.4. The JSON-style linearization is more straightforward, see Appendix D for the details. Since CPT (Shao et al., 2021), so far the best Chinese language encoder-decoder is pretrained on text corpora and unable to generate code, we only try the JSON-style linearization on autoregressive LLMs (ChatGPT and ChatGLM).

2.2 Sequence-to-sequence Models

Figure 2(a) shows the overall framework of our sequence-to-sequence models, which work in 4



(a)

(b)

Figure 2: An overview of our generative medical decision tree extraction models. (a) A sequence-to-sequence model that extracts relation triples within input text and translates the text along with the extracted relation triples into a linearized medical decision tree. (b) An autoregressive model that follows task instructions to generate a linearized medical decision tree conditioned on input text. See Figure 4 for the original Chinese-language prompts.

steps: 1) encodes the input text and entity/relation queries with a pretrained language encoder; 2) generates the entity/relation set with a query-based entity-relation extractor; 3) generates the linearized decision tree with a pretrained language decoder, conditioned on the text encoding, relation representation and extracted relation set; 4) parse the linearized decision tree. Detailed designs are introduced as follows.

2.2.1 Query-based Entity-relation Extraction

The query-based entity-relation joint extractor is the one proposed by He and Tang (2023), which consists of a shared decoder, an entity decoder, a relation decoder, an entity predictor, a relation type predictor and a subject-object predictor. It also

owns a series of learnable entity queries Q_e and relation queries Q_r (each query is a vector), which are concatenated with input text X. Pretrained language encoder and shared decoder transform the concatenation into text encoding H^x along with contextual entity/relation representation H^e/H^r. Entity decoder and relation decoder further update H^e into $\widetilde{H}^e$, update H^r into $\widetilde{H}^r, \widetilde{H}^h, \widetilde{H}^t$ via linear transform and attention mechanism.

The predicted sets of entities $\hat{\mathcal{E}}$ and relations $\hat{\mathcal{R}}$ are finally computed based on $\widetilde{H}^e, \widetilde{H}^r, \widetilde{H}^h, \widetilde{H}^t$. Only $\hat{\mathcal{R}}$ is utilized by downstream modules while $\hat{\mathcal{E}}$ is not. See Appendix A or the work by He and Tang (2023) to learn more about this module.

2.2.2 Relational Context

Since a medical decision tree is essentially a combination of relation triples, leveraging the predicted relation set as an additional decoding context may help the pretrained language decoder keep aware of which triples are already included in the generated sequence and which ones are not. It can address the problem of low triple coverage in the predicted decision tree. Motivated by this idea, three designs of relational context are attempted: 1) Relation query context (RQC), the representation vectors $\widetilde{H}^r$ of relation queries corresponding to all extracted relation triples; 2) Relation-centric textual context (RTC), a cross-attention-based context, where text encoding H^x acts as key and value, relation query vectors $\widetilde{H}^r$ corresponding to all extracted relation triples act as query; 3) Harmonized relation context (HRC), the fusion of RQC and RTC through gating mechanism.

To inject the relational context into the model, we concatenate text encoding H^x with the relational context in the sequence dimension and together they serve as the decoding context for the pretrained language decoder:

$$\boldsymbol{h}^d_{t-1} = \text{Decoder}(\hat{y}_{<t}|[H^x; \mathcal{C}]) \quad (1)$$

$$\mathcal{C} \in \{\text{RQC}, \text{RTC}, \text{HRC}\} \quad (2)$$

$$P(\hat{y}_t) = \text{LMHead}(\boldsymbol{h}^d_{t-1}) \in \mathbb{R}^{|V|} \quad (3)$$

$$\hat{y}_t = \text{DecodeSearch}(P(\hat{y}_t), \hat{y}_{<t}, \hat{\mathcal{R}}) \quad (4)$$

where ';' means concatenation, $\hat{y}_{<t}$ is the generated tokens by time step t, $\boldsymbol{h}^d_{t-1}$ is the undated hidden state of current time step, LMHead is a classifier that first convert current hidden state into vector of size $|V|$ that apply SoftMax to obtain predicted probability distribution $P(\hat{y}_t)$ over the vocabulary, DecodeSearch is the decode search strategy (we use constrained search in this paper, see Section 2.2.3). $\hat{y}_t$ is the token generated for current time step and will be concatenated with $\hat{y}_{<t}$ to restart the process, until the terminal token </s> is generated.

2.2.3 Constrained Decoding

In order to utilize apriori decision tree linearization grammar (as shown in Algorithm 1) to constrain the candidate space of generated target sequence with the set of extracted relations, we employ a specially designed constrained decoding (CD) strategy during generative inference.

Specifically, the strategy restricts the candidate token vocabulary at each generation step based on the generated sequence prefix using a trie. The construction of the trie takes into account the following scenarios: 1) if the sequence prefix is "if", the candidates include the first token of all head entities; 2) if the sequence prefix is "else", the candidate token is only "then"; 3) if the sequence prefix is "then", the candidates include "," and the first token of each head entity; 4) if the sequence prefix is ",", the candidate token is only "if"; 5) if the sequence prefix is the first half of an entity/relation name, the candidates are the first token of the second half of the entity/relation name; 6) if the sequence prefix is a complete head entity, the candidates are the first token of all relation names with that entity as the head; 7) if the sequence prefix is a complete relation name, the candidates include the first token of all tail entities; 8) if the sequence prefix is a complete tail entity, the candidates include "then", "otherwise", and "</s>".

2.2.4 AugNL-style Linearization

Augmenting natural language (Mialon et al., 2023) with tokens of other modalities (e.g., vision(Zhu et al., 2023; Liu et al., 2023) and knowledge graph(Pan et al., 2023)) can not only provide complementary context but also greatly enhance the expression ability. Distinguish from NL style of linearization (Paolini et al., 2021; Lu et al., 2022), where relation triples have to get verbalized before being placed in the target sequence, in AugNL style of linearization relation triples are considered as basic tokens of high-level abstract semantics and get naturally embedded in the target sequence, which decreases the average length of linearized relation triples by 10+ times.

The technical difference between sequence-to-sequence models with NL-style linearization and AugNL-style linearization lies in the decoding

mechanism. Models with AugNL-style linearization employ a pointer-based copy mechanism, where the relational part of generated sequence is made up of pointers to extracted relation triples and the conjunction part of generated sequence is made up of pointers to predefined structure tokens (i.e., "or", "and", "if", "then", "otherwise", ",", "</s>"):

$$P(\hat{y}_t) = \text{Softmax}(\boldsymbol{h}^d_{t-1} \odot [\text{Emb}(\hat{\mathcal{R}}); \text{Emb}(\text{StructureTokens})]) \quad (5)$$

For the embeddings of extracted relation triples $\text{Emb}(\hat{\mathcal{R}})$, we reuse the three designs of relational context representation but name them differently as relation query embeddings (RQE), relation-centric textual embeddings (RTE) and harmonized relation embeddings (HRE) to clarify the different usage.

2.3 Autoregressive Models

In contrast to sequence-to-sequence models, our autoregressive models inherit from decoder-only LLMs, as shown in Figure 2(b). When properly prompted with examples, a LLM can handle simple tasks without supervision, which is known as the ability of in-context learning (ICL). After supervised fine-tuned (SFT), a LLM will get better at modeling the desired output of complex tasks.

We explore the ICL as well as SFT settings. For the first setting, two LLMs, ChatGPT (gpt-3.5-turbo) and ChatGLM are employed, and the NL, JSON styles of linearization are tried (note that AugNL style is inapplicable here). For the SFT setting, we only consider ChatGLM (for reproductivity concern) and the NL style linearization (since the ICL results suggest this style of linearization is more suitable for ChatGLM, see Section 3.2).

2.3.1 Few-shot In-context Learning

In the in-context learning (ICL) setting, autoregressive models are prompted with task instruction for medical decision tree extraction and few-shot demonstration. Specifically, the prompt for autoregressive models with NL-style linearization under the ICL setting is similar to the one in Figure 2(b), except that it contains 5 examples of expected input-output (randomly sampled from the training set). The prompt template is shown in Appendix D.

2.3.2 Multi-task Joint Fine-tuning

Different from unsupervised in-context learning, supervised fine-tuning helps a LLM master complex tasks through end-to-end training on a diverse set of instruction-response pairs. In this work, we propose a multi-task joint fine-tuning method for our autoregressive models, where medical decision tree extraction is the main task, relation triple extraction and tree shape extraction serve as the auxiliary tasks. And a novel progressively-dynamic sampling strategy helps the model gradually acquire easy-to-hard structural extraction abilities.

Prompts for these tasks are illustrated in Figure 2(b). The target output of medical decision tree extraction is just the NL-style linearized tree. The target output of relation triple extraction is all mentioned relation triples in list format (ordered by textual position). The target output of tree shape extraction is the skeleton of a tree, made up of conjunctions and ellipses. Our progressively-dynamic sampling strategy is inspired by curriculum learning (Wang et al., 2021). With the increase of training step, the sampling rate of each task changes according to the assumed task difficulty: for relation triple extraction, the sampling rate goes from 0.8 to 0 linearly; for tree shape extraction, the sampling rate goes from 0.7 to 1 linearly; for the main task, the sampling rate stays as 1.

2.4 Data augmentation and model ensemble

The SOTA baseline, PromptRE (Jiang et al., 2022), leverages R-Drop (Wu et al., 2021) for data augmentation, and assembles predictions of relation triples after each round of relation extraction. However, their practices are inapplicable to generative models. For a fair comparison, we devise a general data augmentation method and model ensemble method for the task. To obtain augmented samples, we randomly replace entities within the train data with their synonyms. For model ensemble, our system first vote on the tree structures predicted by multiple models and then vote on the content (logical operator and relation triples) of each node. *Note that, our top models outperform SOTA baselines even without these tricks (see Section 3.2).*

3 Experiments

3.1 Data and Evaluation Metrics

We conduct experiments on the only available medical decision tree extraction dataset, Text2DT, which is from a shared task of the 8th China Health Information Processing Conference (Zhu et al., 2022) and get included in the CBLUE 3.0 benchmark (Zhang et al., 2022). Built on a rich corpus of Chinese medical textbooks and guidelines, it covers diagnosis and treatment knowledge of around 200

Paradigm	Method	Triple F1(%)	Node F1(%)	Path F1(%)	Tree Acc(%)
Discriminative	BERT-Biaffine (2022)†	90.19	74.80	52.71	37.00
	PromptRE (2022)†‡	94.39	85.31	69.27	55.00
Sequence-to-sequence	**CPT (NL)**	92.67±0.20	83.54±0.26	66.27±0.51	51.00±0.89
	CPT (NL)†	92.96±0.33	83.68±0.41	66.55±0.64	52.50±1.01
	CPT (NL)†‡	94.08	86.45	70.63	59.00
	CPT (AugNL)	93.21±0.19	85.06±0.32	68.13±0.55	55.50±1.06
	CPT (AugNL)†	94.18±0.29	86.97±0.26	69.47±0.58	58.00±0.99
	CPT (AugNL)†‡	<u>95.04</u>	88.43	**78.26**	<u>66.00</u>
Autoregressive ICL	ChatGPT (JSON)	73.12±0.42	63.56±0.57	44.61±0.73	28.00±1.22
	ChatGPT (NL)	70.60±0.61	58.59±0.74	35.08±0.98	22.00±1.30
	ChatGLM (JSON)	54.56±0.45	42.86±0.52	23.25±0.66	9.00±1.07
	ChatGLM (NL)	58.67±0.70	49.52±0.83	27.11±0.93	17.00±1.36
Autoregressive SFT	**ChatGLM (NL)**	92.26±0.37	87.70±0.42	71.51±0.67	60.00±0.98
	ChatGLM (NL)†	91.60±0.34	87.59±0.39	72.41±0.60	61.50±0.93
	ChatGLM (NL)†‡	93.92	<u>90.00</u>	77.05	<u>66.00</u>
Final Ensemble†‡		**95.43**	**90.48**	<u>77.91</u>	**67.00**

Table 1: Main Results. † or ‡ mean using data augmentation or model ensemble respectively. The version of ChatGPT is gpt-3.5-turbo. Final Ensemble is the ensemble of CPT (AugNL)† and ChatGLM (NL)†. The highest scores are in bold and the second-highest scores are underlined. Standard errors are included when applicable.

CD	RQC	RTC	HRC	Triple F1(%)	Node F1(%)	Path F1(%)	Tree Acc(%)
				89.43	79.68	60.10	45.75
✓				92.63	82.35	63.45	48.25
✓	✓			92.88	81.65	61.31	47.00
✓		✓		**92.67**	**83.54**	**66.27**	**51.00**
✓			✓	<u>92.83</u>	<u>83.23</u>	<u>64.87</u>	<u>50.25</u>

Table 2: Results of ablation experiments on sequence-to-sequence models with NL-style linearization (without data augmentation and model assemble). "CD", "RQC", "RTC" and "HRC" are abbreviations of Constrained Decoding, Relation Query Context, Relation-centric Textual Context and Harmonized Relation Context respectively.

diseases. Six categories of relation are annotated in the dataset, including "symptom", "medication", "treatment", "usage", "caution" and "basic info". All annotations are verified by clinical experts to ensure clinical validity. Dataset statistics is provided in Appendix F.

The performance of different medical decision tree extraction methods is evaluated using the following metrics: 1) **Triple F1 Score**: for each triple in the extracted decision tree, it is considered correct only if it is identical to a triple in the ground-truth decision tree; 2) **Node F1 Score**: for each node in the extracted decision tree, it is considered correct only if it is identical to a node in the ground-truth decision tree; 3) **Path F1 Score**: for each path (from the root node to a leaf node) in the extracted decision tree, it is considered correct only if all nodes within are identical to those of a path in the ground-truth decision tree; 4) **Tree Accuracy**: an extracted decision tree is considered correct only if its structure and all contained nodes are identical to those of the ground-truth decision tree.

We compare our models with SOTA medical decision tree extraction methods, BERT-Biaffine and PromptRE (see Section 4.3 for an introduction). All results without ensemble are averaged over 5 runs and reported with standard errors. Otherwise, the results are recorded for the ensemble of 5 models under different random seeds and it is inapplicable to compute the standard errors. Please refer to Appendix E for more details on implementation.

3.2 Main Results

The overall performance of different models on Text2DT is shown in Table 1. In comparison of different paradigms, sequence-to-sequence and autoregressive models (under the SFT setting) exhibit top-2 capacity on the task, achieving tree accuracy of 55.5% and 60% respectively without data augmentation and model ensemble, outperforming

RQC	RTC	HRC	RQE	RTE	HRE	Triple F1(%)	Node F1(%)	Path F1(%)	Tree Acc(%)
			✓			92.09	82.62	65.21	49.50
✓			✓			92.86	83.97	62.03	52.50
	✓		✓			93.12	84.69	67.51	54.50
		✓	✓			93.00	84.32	66.98	54.00
				✓		92.27	82.36	64.94	48.75
					✓	92.74	83.45	66.49	51.00
	✓				✓	**93.21**	**85.06**	**68.13**	**55.50**

Table 3: Results of ablation experiments on sequence-to-sequence models with AugNL-style linearization (without data augmentation and model assemble). "RQE", "RTE" and "HRE" are abbreviations of Relation Query Embeddings, Relation-centric Textual Embeddings and Harmonized Relation Embeddings respectively.

RE	TS	PDS	Triple F1	Path F1	Tree Acc
			87.44	66.55	53.00
✓			89.65	67.98	57.00
	✓		90.10	68.35	57.00
✓	✓		90.44	70.83	59.50
✓	✓	✓	**92.26**	**71.51**	**60.00**

Table 4: Ablation results of autoregressive models under the SFT setting (without data augmentation and model assemble). "RE", "TS" mean the auxiliary Relation Triple Extraction and Tree Shape Extraction tasks respectively. "PDS" stands for the progressively-dynamic sampling strategy. We omit the "%" marks here.

the SOTA discriminative method by a large margin. After applying data augmentation and model ensemble, both models reach 66% tree accuracy, higher than the current SOTA by 11%. The tree accuracy further increases to 67% when combining these two families of models.

The evaluation results of sequence-to-sequence models suggest AugNL-style linearization is remarkably better than NL style for sequence-to-sequence generation, boosting the tree accuracy by 4.5%, 5.5% and 6% respectively under 3 different settings of data augmentation and model ensemble.

The evaluation results of autoregressive models in the ICL setting demonstrate the superiority of ChatGPT over ChatGLM on generating JSON code and Chinese language. However, the gap between ChatGLM and ChatGPT is much smaller on Chinese language generation than on JSON code generation. The results also show that barely relying on LLMs and ICL is insufficient to solve the task of medical decision tree extraction. Although ChatGPT reaches 28% tree accuracy when prompted to generate JSON-style linearized decision tree, it is still far from satisfaction.

3.3 Ablation Study

We conduct extensive ablation experiments on the proposed generative models to verify the contributions of different components and determine the optimal design choice among alternative component designs. The results are shown in Tables 2-4.

Table 2 presents the results for sequence-to-sequence models with NL-style linearization. By applying constrained decoding, the tree accuracy improves from 45.75% to 48.25%, validating the necessity of constrained decoding. Besides, relation-centric textual context works better than relation query context or harmonized relation context, boosting tree accuracy by 2.75%. This result indicates a higher acceptance of relation-centric textual context by the pretrained decoder, compared to the relation query representations output by the relation set generator. The reason may lie in the semantic space consistency between relation-centric textual context and natural language, making it more conducive to natural language generation.

For sequence-to-sequence models with AugNL-style linearization, the combination of relation-centric textual context and harmonized relation embeddings works better than other alternatives, as shown in Table 3. This is expected, since harmonized relation embeddings are designed to bridge the relational context and textual context.

For autoregressive models under the SFT setting, the auxiliary relation triple extraction and tree shape extraction tasks contribute equally to model performance, leading to 4% absolute tree accuracy increment respectively. When the two auxiliary tasks are applied together, tree accuracy increases from 53% to 59.5%. By incorporating progressively-dynamic sampling, tree accuracy further increases by 0.5% and reaches 60%.

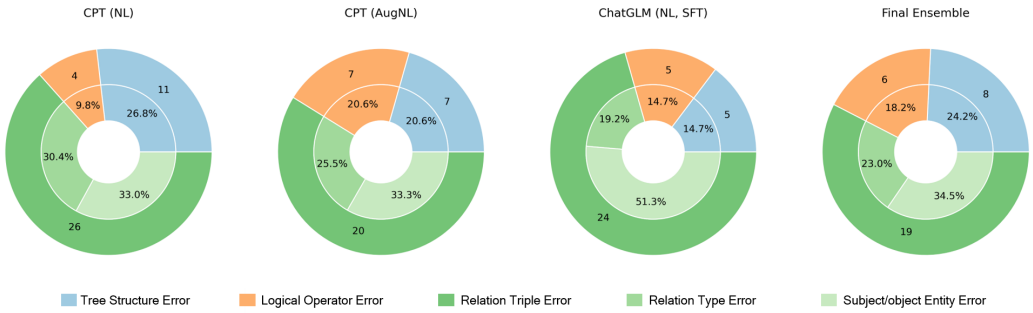


Figure 3: Error distributions (on the test split) of different generative models.

3.4 Error Analysis

To discover the performance bottleneck of this task and facilitate future research, we analyze the errors by our top-performing models. Error distributions on the test split of Text2DT are shown in Figure 3, from which we can observe that: 1) The amount of Logical operator errors is the least, while relation triple errors occur most frequently, especially for generative models with NL-style linearization. 2) Sequence-to-sequence models with NL-style linearization have difficulty in correctly predicting the tree structures. 3) Assembling CPT (AugNL) and ChatGLM (NL) reduces relation triple errors but not the logical operator errors or tree structure errors. 4) Compared to sequence-to-sequence models, autoregressive models produces much more subject/object entity errors, which means they are weak at identifying entity boundaries.

4 Related Work

4.1 Sequence-to-sequence Generation

The idea of sequence-to-sequence was originated from Sutskever et al. (2014), and then dominated neural machine translation (Wu et al., 2016; Zhang et al., 2019) with Transformer (Vaswani et al., 2017). There are many encoder-decoder language models, e.g. T5 (Raffel et al., 2019) and CPT (Shao et al., 2021), that are pretrained with sequence-to-sequence learning tasks. Some works accomplish generating a linearized structure from text in a sequence-to-sequence manner, e.g. neural AMR (Konstas et al., 2017) for AMR parsing, and Text2Event (Lu et al., 2021) for event extraction. Compared to events and AMR graphs, medical decision trees additionally carry logical semantic knowledge, which is non-trivial to capture.

4.2 Autoregressive Generation

The autoregressive generation paradigm employs a single decoder network to generate an output sequence by iteratively predicting the next token conditioned on the current prefix, without the use of an encoder network. Despite its simplicity, this paradigm is shown to generalize better under the zero-shot and few-shot settings (Wang et al., 2022). Besides, it is easier to scale up, leading to LLMs, e.g. GPT-4 (Achiam et al., 2023) and ChatGLM (Du et al., 2022). Many works tackle information extraction tasks by prompting LLMs to autoregressively generate structural content in JSON or other formats (Xu et al., 2023).

4.3 Medical Decision Tree Extraction

Existing medical decision tree extraction methods (Wu, 2022; Jiang et al., 2022) rely on discriminative models. Wu (2022) proposes to combine a BERT-style language model (Cui et al., 2021) with a Biaffine model (Dozat and Manning, 2016) to extract the relation triples, classify the logical connection among triples, and compose the tree. SOTA method, PromptRE (Jiang et al., 2022), formulates medical tree extraction as a multi-round conditional relation extraction task, where each parent node is a condition for extracting relation triples of its left/right child from text. Concurrent with our work, Text2MDT (Zhu et al., 2024) comes up with an expanded version of the Text2DT dataset (**unavailable yet**) and reports some new results.

5 Conclusion

In this study, we present several generative models to extract medical decision trees, which are valuable for CDSS but costly to acquire manually. The proposed models inherit two mainstream text generation paradigms, i.e. sequence-to-sequence generation and autoregressive generation, which bring advantage in modeling both source text and the intrinsic logical connection among tree components. Experiments show that our method wins the SOTA discriminative method by a large margin, es-

tablishing new SOTA with 67% tree accuracy and 78% path F1 score. Besides, an in-depth analysis of error distribution reveals the pros and cons of different models, paving the way for future research on this area. Another direction for future efforts is the evaluation of predicted decision tree's clinical usefulness in real-world scenario, which requires consideration of potential ethical risks and careful experimental designs.

6 Limitations

In this section, we summarize the limitations of our work as follows:

- Although the proposed method is applicable to languages like English, we only experiment on a public Chinese dataset, since there are no other available datasets.
- Entity normalization is not covered in this work, which means the extracted rules are not readily compatible with existing biomedical knowledge bases like UMLS. Future work should include entity normalization a step of post processing, or enhance the formulation and models to support entity normalization.
- We only look into the extraction of medical decision rules in this study, but not decision rules on other knowledge-intensive domains, such as mineral exploration (Duda et al., 1981) and mathematics (Beeson, 1989). However, the proposed method is in fact domain-agnostic and we believe there is no barrier to extend our method to other domains.

Acknowledgments

We thank the reviewers for their insightful comments and valuable suggestions. This study is partially supported by National Key R&D Program of China (2023YFC3502900), National Natural Science Foundation of China (62276082), Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS16/E09/22), Major Key Project of PCL (PCL2021A06), Shenzhen Soft Science Research Program Project (RKX20220705152815035), Shenzhen Science and Technology Research and Development Fund for Sustainable Development Project (GXWD20231128103819001, No.KCXFZ20201221173613036, 20230706140548006) and the Fundamental Research Fund for the Central Universities (HIT.DZJJ.2023117).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Michael J. Beeson. 1989. Logic and computation in mathpert: An expert system for learning mathematics. In *Computers and Mathematics*, pages 202–214, New York, NY. Springer US.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE Transactions on Audio, Speech and Language Processing*.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *ArXiv*, abs/1611.01734.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Richard Duda, John Gaschnig, and Peter Hart. 1981. Model design in the prospector consultant system for mineral exploration. In Bonnie Lynn Webber and Nils J. Nilsson, editors, *Readings in Artificial Intelligence*, pages 334–348. Morgan Kaufmann.

Crina Grosan, Ajith Abraham, Crina Grosan, and Ajith Abraham. 2011. Rule-based expert systems. *Intelligent systems: A modern approach*, pages 149–185.

Yuxin He, Jingyue Hu, and Buzhou Tang. 2023. Revisiting event argument extraction: Can EAE models learn better when being aware of event co-occurrences? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12542–12556, Toronto, Canada. Association for Computational Linguistics.

Yuxin He and Buzhou Tang. 2022. Setgner: General named entity recognition as entity set generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuxin He and Buzhou Tang. 2023. Bispn: Generating entity set and relation set coherently in one pass. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2066–2077.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yiwen Jiang, Hao Yu, and Xingyue Fu. 2022. Medical decision tree extraction: A prompt based dual contrastive learning method. In *China Health Information Processing Conference*, pages 103–116. Springer.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *ArXiv*, abs/2304.08485.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Annual Meeting of the Association for Computational Linguistics*.

Y Matsumura, T Matsunaga, Ryuji Hata, Michio Kimura, and H Matsumura. 1986. Consultation system for diagnoses of headache and facial pain: Rhinos. *Medical Informatics*, 11(2):145–157.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *ArXiv*, abs/2302.07842.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. *ArXiv*, abs/2306.08302.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *ArXiv*, abs/2101.05779.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *ArXiv*, abs/2109.05729.

Edward H Shortliffe and Martin J Sepúlveda. 2018. Clinical decision support in the era of artificial intelligence. *Jama*, 320(21):2199–2200.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *ArXiv*, abs/1409.3215.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. *ArXiv*, abs/2105.08901.

Shusaku Tsumoto. 1998. Automated extraction of medical expert system rules from clinical databases based on rough set theory. *Information sciences*, 112(1-4):67–84.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zihong Wu. 2022. Research on decision tree method of medical text based on information extraction. In *China Health Information Processing Conference*, pages 127–133. Springer.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *Preprint*, arXiv:2312.17617.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 185–197, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Annual Meeting of the Association for Computational Linguistics*.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhifang Sui, Baobao Chang, Hui Zong, Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. 2022. CBLUE: A Chinese biomedical language understanding evaluation benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7888–7915, Dublin, Ireland. Association for Computational Linguistics.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Wei Zhu, Wenfeng Li, Xing Tian, Pengfei Wang, Xiaoling Wang, Jin Chen, Yuanbin Wu, Yuan Ni, and Guotong Xie. 2024. Text2mdt: Extracting medical decision trees from medical texts. *arXiv preprint arXiv:2401.02034*.

Wei Zhu, Wenfeng Li, Xiaoling Wang, Wendi Ji, Yuanbin Wu, Jin Chen, Liang Chen, and Buzhou Tang. 2022. Extracting decision trees from medical texts: An overview of the text2dt track in chip2022. In *China Health Information Processing Conference*, pages 89–102. Springer.

A Details of Query-based Entity-relation Extraction

The input of the query-based entity-relation extractor is the concatenation of embedded text X and a series of learnable entity/relation queries Q_e/Q_r:

$$\tilde{X} = [X; Q_e; Q_r] \in \mathbb{R}^{(L+M_e+M_r)\times d} \quad (6)$$

where d is model dimension, M_e and M_r are the numbers of entity/relation queries (set as the maximum amount of entity/relation in a single text of the corpus).

The bidirectional self-attention of the pretrained encoder is modified into one-way self-attention. Concretely, the upper right $L \times (M_e + M_r)$ submatrix of the attention mask gets filled with negative infinity value so that the entity/relation queries become invisible to the token encodings, while the entity/relation queries can still attend to each other and the token encodings. After multiple one-way self-attention layers and feed-forward layers, the encoder outputs the contextual token encodings as well as the contextual entity/relation queries.

The entity-relation shared decoder consists of one-way self-attention layers that mimics the ones in the encoder, as well as bidirectional self-attention layers that updates the entity/relation queries via modeling the interaction among them. It outputs the updated token representations H^x, entity queries H^e and relation queries H^r.

The entity generator consists of an entity decoder and an entity predictor. It first linearly transforms the token encodings H^x into entity-view:

$$H_e^x = \text{FC}(H^x) \quad (7)$$

where FC means a fully connected linear layer.

The entity decoder, equipped with cross-attention and bidirectional self-attention, receives H_e^x as decoding context and the entity queries H_e as decoder input, and outputs the final representation of entity queries $\tilde{H}^e$:

$$\tilde{H}^e = \text{EntityDecoder}(H^e | H_e^x) \quad (8)$$

For the i-th entity query, the entity predictor predicts the probability distribution of start/end (P_i^{start}/P_i^{end}) and the probability distribution of entity type $P_i^{t^e}$ as follows:

$$S_i^\delta = \text{FC}\big(\text{Relu}(\text{FC}(\tilde{H}_i^e) + \text{FC}(H_e^x))\big) \quad (9)$$

$$P_i^\delta = \text{Softmax}(S_i^\delta),\ \delta \in \{\text{start}, \text{end}\} \quad (10)$$

$$P_i^{t^e} = \text{Softmax}(\text{MLP}(\tilde{H}_i^e)) \quad (11)$$

During inference, the predicted boundary and entity type corresponding to the k-th entity query are calculated as:

$$\text{score}_k(i,j) = P_k^{\text{start}}[i] + P_k^{\text{end}}[j] \quad (12)$$

$$(\hat{start}_k, \hat{end}_k) = \underset{(i,j):\ 0<j-i<L}{\arg\max}\ \text{score}_k(i,j) \quad (13)$$

$$\hat{t_k^e} = \arg\max P_k^{t^e} \quad (14)$$

Note that, entities whose predicted type label is $\varnothing$ will be excluded from the generated entity set.

The relation generator consists of a relation decoder, a subject-object predictor and a relation type predictor. The relation decoder work in the same manner as the entity decoder, except that the relation decoder splits relation queries into head/tail queries before decoding:

$$[H^h; H^t] = \text{FC}(H^r) \tag{15}$$

$$H_r^x = \text{FC}(H^x) \tag{16}$$

$$\tilde{H^h}, \tilde{H^t}, \tilde{H^r} = \text{RelDecoder}(H^h, H^t, H^r | H_r^x) \tag{17}$$

The subject-object predictor then predicts the boundary and entity type of the head/tail entity associated with each relation queries. This process is similar to the entity prediction process. The only difference is that the entities queries becomes the head/tail queries $\tilde{H}^{h/t}$ and the token encodings is now in relation-view H_r^x.

The relation type predictor classifies the category of i-th relation query according to $\tilde{H}_i^r$:

$$P_i^{t^r} = \text{Softmax}(\text{MLP}(\tilde{H}_i^r)) \tag{18}$$

To train the entity-relation extractor, a combination of optimal-assignment-based prediction loss, bipartite consistency loss and entity-relation linking loss are utilized. Please refer to He and Tang (2023) for information about the loss functions.

B Original Chinese-language Prompts

The prompts shown in Figure 2(b) are translated from Chinese into English for ease of reading. Figure 4 illustrates the original prompts.

C NL/AugNL-style Linearization

Algorithm 1 illustrates the concrete procedure of linearizing a medical decision tree into NL/AugNL-style sequence.

D JSON-style Linearization

To linearize a medical decision tree in JSON style, we only need to pack the tree nodes along with their content as nested key-value pairs in pre-order. An example of the utilized JSON template is included in Figure 5.

Algorithm 1 NL/AugNL-style Linearization

Require: $tree$ (a medical decision tree)

$seq \Leftarrow$ “”
while $tree$.preorderNext() **do**
  $node$ = $tree$.preorderNext()
  if isCondition($node$) **then**
    if isLeft($node$) **then**
      seq += “if”
    else
      seq += “else, if”
    end if
  else
    if isLeft($node$) **then**
      seq += “then”
    else
      seq += “otherwise”
    end if
  end if
  if isOrLogic($node$) **then**
    seq += “or”.join($node$.triples())
  else
    seq += “and”.join($node$.triples())
  end if
end while
return seq

E Implementation details

Our sequence-to-sequence models is initialized with CPT-large(Shao et al., 2021), which has 20-layer encoder and 4-layer decoder. The numbers of entity queries and relation queries are set as 30, 25 respectively. We train the models in 2 stages: in the first stage (70 epochs), the pretrained language decoder are frozen and the encoder, entity-relation extractor are optimized with the entity-relation extraction loss; in the second stage (100 epochs), all modules are jointly optimized. The learning rate of the encoder and decoder are set as 3e-5 and 4e-5 respectively. An AdamW(Loshchilov and Hutter, 2017) optimizer with linear warm-up is employed.

For ICL, the autoregressive models are two plug-and-play commercial natural language assistant: 1) ChatGPT (gpt-3.5-turbo version); 2) ChatGLM (chatglm_pro version). We invoke them via API. The default temperature is applied and the number of examples within each prompt is set as 5. The 5 examples are randomly sampled from the training set in a stratified manner (each instance corresponding to one tree structure). For SFT, the autoregressive models are initialized with ChatGLM-6B and

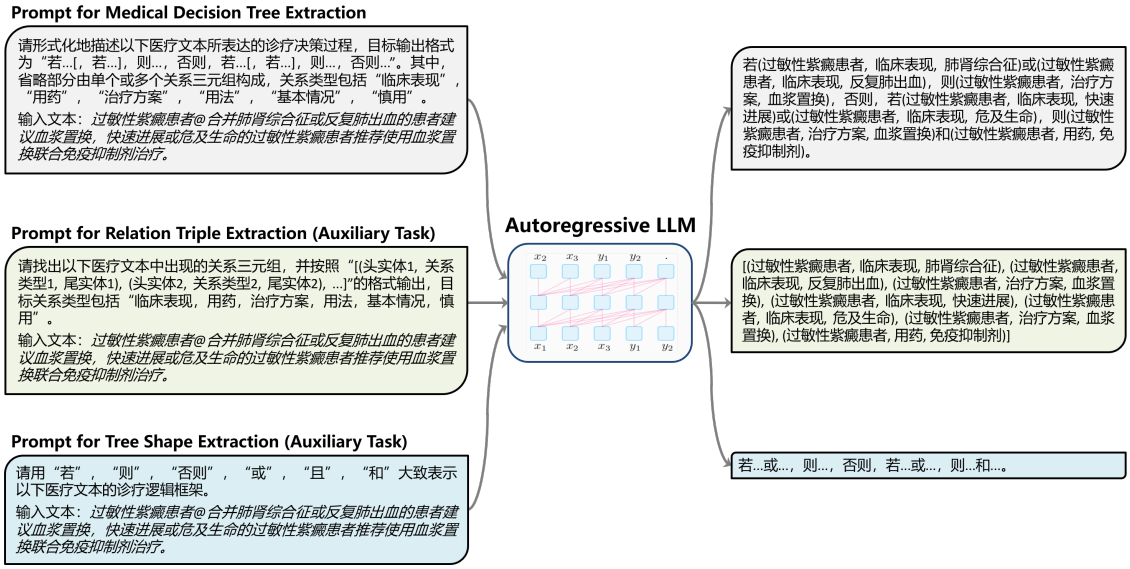


Figure 4: Original Chinese prompts for the main decision tree extraction task (with NL-style linearization) and auxiliary tasks (relation triple extraction, tree shape extraction) introduced in Section 2.3.2.

Item	Count
Texts	500
Train/Dev/Test Splits	300/100/100
Avg. Text Length	66.5
Relation Classes	6
Relations per Text	6.39
Relation Name	**Count (Proportion)**
Symptom	1374 (42.51%)
Medication	910 (28.15%)
Treatment	561 (17.36%)
Usage	222 (6.87%)
Caution	83 (2.57%)
Basic Info	82 (2.54%)
Tree Structure (Pre-order)	**Count (Proportion)**
CDD	134 (26.80%)
CDCDD	253 (50.60%)
CCDDD	47 (9.40%)
CDCDCDD	45 (9.00%)
CCDCDDD	17 (3.40%)
CCDDCDD	2 (0.40%)
CDCDCDCDD	2 (0.40%)

Table 5: Statistics of the Text2DT Dataset (“C”/“D” represents a “condition”/“decision” node)

tuned with LoRA(Hu et al., 2021). The LoRA rank, learning rate, batch size and number of training sets is set as 8, 2e-4, 8 and 2000 respectively.

The number of parameters of our sequence-to-sequence models is less than 1B. Whereas the number of parameters of our autoregressive models based on ChatGLM is 6B. All experiments are conducted on an NVIDIA A100 server, and the computational budget for training each model does not exceed 4 GPU hours.

F Dataset Statistics

Detailed statistics of the Text2DT dataset are listed in Table 5.

G Performance on Generating Trees of Different Depths

There are 7 types of tree structures in the dataset and the depth of annotated decision trees ranges from 2 to 5, as illustrated in Table 5. To analyze the difference between generative models on extracting trees of different complexity, we split the test set according to tree depth and evaluate the model performance on each split respectively. The results are illustrated in Figure 6, from which we can draw the following conclusions:

- Deeper trees are more difficult to be correctly generated than shallower ones.
- For sequence-to-sequence models, the performance gap between NL and AngNL styles of linearization lies on extracting deeper trees.
- In the ICL setting, ChatGPT with JSON-style linearization gains most of its points from trees of depth 2. Under other circumstances, both ChatGPT and ChatGLM perform quite poorly, regardless of the linearization style.

利用医学指南文本生成诊疗决策树：
任务说明：(1)根据给定医学指南文本，创建一个二叉树，包含条件节点和决策节点，用以简洁地展示指南内容，同时捕捉核心实体和关系；(2)条件节点用于判断，根据结果指向左侧或右侧子节点进行下一步决策。(3)每个节点输出为dict，包含三个字段：(3a)"role"表示节点类型，可以是条件节点("C")或决策节点("D")；(3b)"triples"是一个三元组列表，描述诊疗知识或临床信息，包含"临床表现", "治疗药物", "用法用量", "治疗方案", "禁用药物", "基本情况"六类关系；(3c)"logical_rel"表示多个三元组之间的逻辑关系（取值为and, or, null，当只有一个三元组时逻辑关系为 null）。(4)最终生成的诊疗决策树按广度优先策略排列为一个列表，并保证构成合法的二叉树。
你只能按照JSON格式回复。请勿使用任何其他格式。

下面是一些例子：
文本：风湿热患者@对于舞蹈病患者，首选丙戊酸，丙戊酸无效或严重舞蹈病如瘫痪的患者,应用卡马西平治疗。
诊疗决策树：

```
[
    {
        "role": "C",
        "triples": [["风湿热患者", "临床表现", "舞蹈病"]],
        "logical_rel": "null"
    },
    {
        "role": "C",
        "triples": [["风湿热患者", "临床表现", "丙戊酸无效"], ["风湿热患者", "临床表现", "严重舞蹈病"], ["风湿热患者", "临床表现", "瘫痪"]],
        "logical_rel": "or"
    },
    {
        "role": "D",
        "triples": [["风湿热患者", "治疗药物", "卡马西平"]],
        "logical_rel": "null"
    },
    {
        "role": "D",
        "triples": [["风湿热患者", "治疗药物", "丙戊酸"]],
        "logical_rel": "and"
    },
    {
        "role": "D",
        "triples": [],
        "logical_rel": "null"
    }
]
```

.....

现在生成以下文本对应的诊疗决策树：
文本：过敏性紫癜患者@合并肺肾综合征或反复肺出血的患者建议血浆置换，快速进展或危及生命的过敏性紫癜患者推荐使用血浆置换联合免疫抑制剂治疗。
诊疗决策树：

Figure 5: The prompt for generating the JSON-style linearized medical decision tree (utilized by autoregreesive large language models under the ICL setting). It includes 5 instances for demonstration, which are randomly sampled from the training set in a stratified manner (each instance corresponding to one tree structure).

- Supervised fine-tuned ChatGLM outperforms sequence-to-sequence models with AngNL linearization on generating trees of depth 4, but is sub-optimal on generating trees of depth 2 or 3.

- Assembling CPT (AugNL) and ChatGLM (NL, SFT) leads to the most balanced performance on extracting trees of different depths.

H Diversity of Trees Generated by Different models and Its Influence

The performance gains after ensemble vary with different paradigms of models, as observed in Table 1. We suspect this is due to the difference in the "diversity" of trees generated by different models. To verify that, we measures the similarity between medical decision trees using edit distance. The edit distance for medical decision trees is the minimum number of tree edit operations (i.e., inserting or deleting a node, changing a node role, inserting or deleting a triplet and modifying a logical operator) required to transform one tree into another. For a group of trees, the average edit distance between each pair of trees is denoted as the "diversity". Figure 7 shows the diversity of trees generated by various models. It is observed that the diversity of trees by sequence-to-sequence models is much stronger than that of autoregressive models, and that the diversity is the strongest when integrating

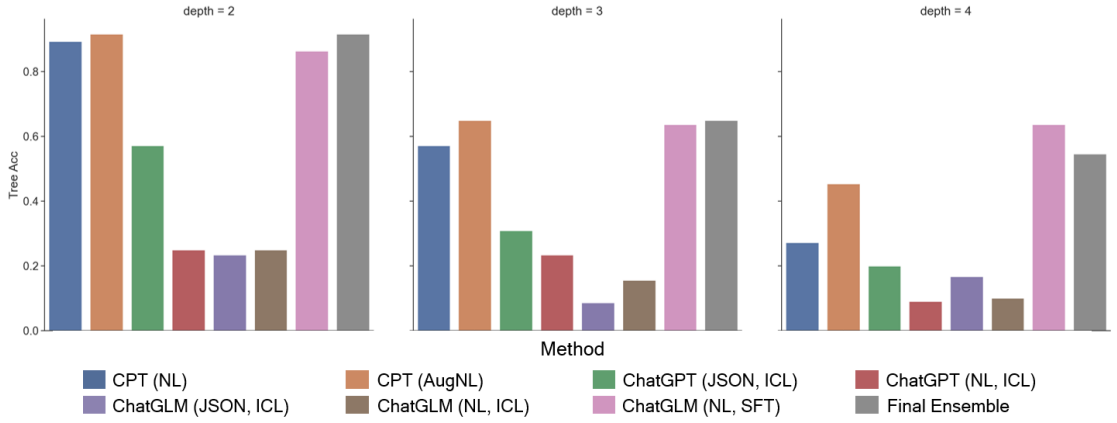


Figure 6: Comparison of different generative models on extracting trees of different depths. Results here are recorded for 5-model ensembles and it is inapplicable to include error bars. Trees of depth=5 only exist in the training data but not the evaluation data, so there is not result for depth=5.

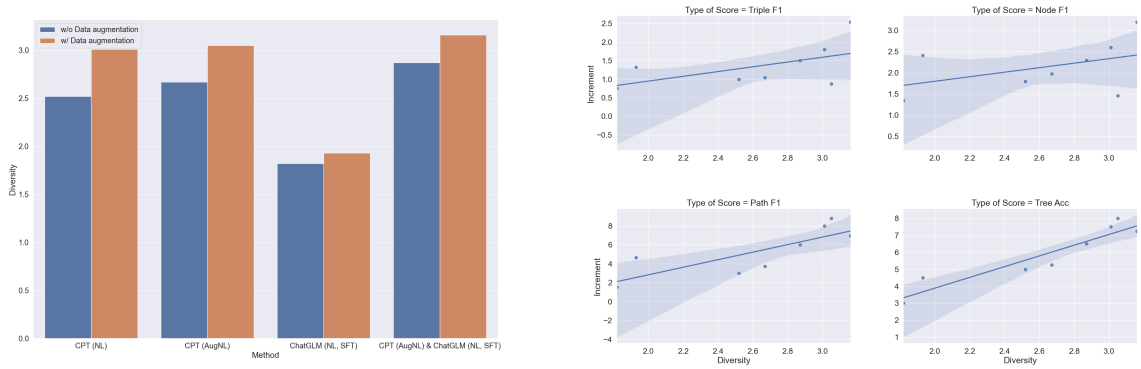


Figure 7: Diversity of trees generated by different models and its correlation with the performance gain after ensemble.

these two paradigms of models.

In Figure 7, a scatter plot with a (least square) fitted line depicts the correlation between tree diversity and performance increment after ensemble. It certifies that the tree diversity has a weak positive correlation with the increment of Triple/Node F1 after ensemble, and a strong positive correlation with the increment of Path F1 and Tree Acc after ensemble.