Text2MDT: Extracting Decision Trees from Medical Texts Using Large Language Models

Anonymous ACL submission

Abstract

Knowledge of the medical decision process, which can be modeled as medical decision trees (MDTs), is critical to building clinical decision support systems. However, the current MDT construction methods rely heavily on time-consuming and laborious manual annotation. In this work, we propose a novel task, Text2MDT, to explore the automatic extraction of MDTs from medical texts such as medical guidelines and textbooks. We normalized the form of the MDT and created an annotated Text2MDT dataset in Chinese with the participation of medical experts. We investigate two different methods for the Text2MDT tasks: (a) an end-to-end framework that only relies on a GPT style large language models (LLM) instruction tuning to generate all the node information and tree structures. (b) The pipeline framework decomposes the Text2MDT task into three subtasks. Experiments on our Text2MDT dataset demonstrate that (a) the end-to-end method based on LLMs (7B parameters or larger) shows promising results and successfully outperforms the pipeline methods. (b) The chain-of-thought (COT) prompting method (Wei et al., 2022) can improve the performance of the fine-tuned LLMs on the Text2MDT test set. (c) the lightweight pipelined method based on encoder-based pre-trained models also performs well with LLMs with model complexity two magnitudes smaller.[1].

1 Introduction

As a typical application of artificial intelligence in the medical field, clinical decision support systems (CDSS) have been widely concerned by researchers (Tsumoto, 1998; Fotiadis et al., 2006; Machado et al., 2017). CDSS can suggest experienced doctors of all the options and problems to be considered when making decisions, help inexperienced medical students to learn clinical knowledge, or give medical advice to patients without medical background (IoannisVourgidis et al., 2018). The core of building a CDSS is the knowledge of medical decision processes, which are rules that link given conditions to medical decisions (Abraham, 2005) and are usually modeled as medical decision trees (MDTs). However, existing methods for constructing MDTs rely on manual tree construction by medical experts (Saibene et al., 2021), which is time-consuming, laborious, and cannot absorb the latest research timely. All these hinder the construction, dissemination, and maintenance of large-scale CDSS (Nohria, 2015). There is an unmet need to explore automated pipelines to precisely extract MDTs from vast and rapidly growing medical knowledge sources.

It is computationally challenging to automatically extract MDTs for the following reasons: 1) the current MDT lacks a normalized and structured form, leading to ambiguity in understanding medical decision knowledge and therefore hinders automated knowledge extraction; 2) the NLP community lacks a benchmark dataset for training and validating MDT extraction tasks; and constructing such data is challenging in that annotating medical decision trees requires in-depth domain knowledge; 3) existing methods for medical information extraction are not directly applicable for MDT extraction.

In this work, we formally define Text-to-MDT (Text2MDT), the task of automatic extraction of MDTs from medical texts. As shown in Figure 1, the knowledge of a medical decision process embedded in the medical text can be modeled as a binary decision tree. In this work, we construct the first Text2MDT benchmark dataset with the help of well-trained annotators and medical experts.

With the constructed Text2MDT benchmark, we systematically evaluate different pre-trained model-based methods. The first cohort of methods we

[1]Our Text2MDT dataset and the source codes are open-sourced, and we will make the dataset and the source codes openly available upon acceptance.

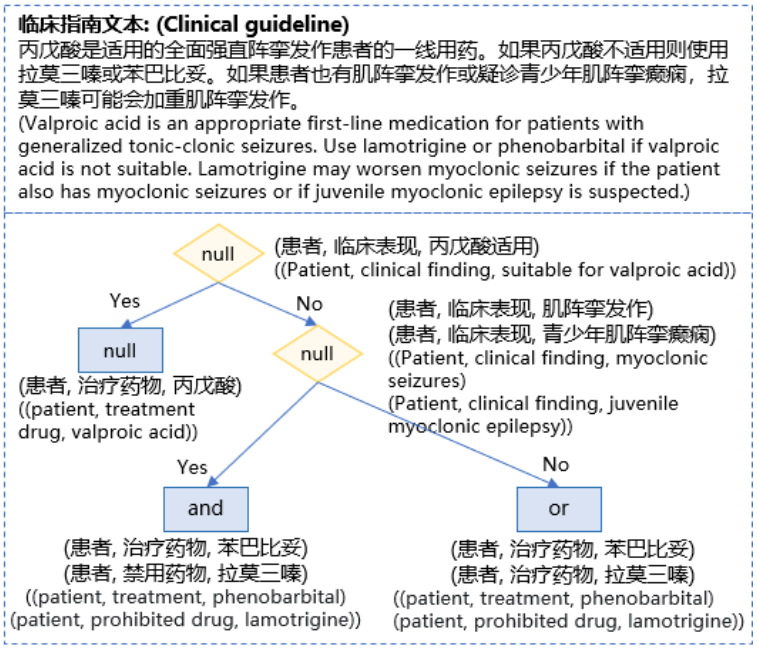

Figure 1: An example of a medical decision tree contained in a medical text from an epilepsy clinical guideline. English translations are provided in brackets.

consider is from the pipeline framework, in which the Text2MDT task is decomposed into three subtasks: triplet extraction, node grouping, and tree assembling. The second cohort of methods are all end-to-end (end2end) methods utilizing pretrained generative LMs, especially the current large language models. Notably, the chain-of-thought (Wei et al., 2022) (COT) style reasoning is also utilized and demonstrated to be beneficial. Experiments on our Text2MDT benchmark show promising results.

In summary, the main contributions of this work are:

- We propose a well-defined novel task, Text2MDT, to extract MDTs from medical text automatically. We construct the first Text2MDT benchmark dataset with the help of medical experts.
- Both the pipeline and end2end models are investigated, including encoder-based methods and LLM fine-tuning methods. The experiments show that LLMs can perform strongly on our Text2MDT benchmark. However, the encoder-based models can also perform well under the pipeline framework.
- The Text2MDT dataset and source codes will be openly available to facilitate future research.

2 Related Work

Due to limited length, we put the Related Work for medical natural language processing and medical information extraction in the Appendix A.

2.1 Text2Tree modeling

There is a rich history of NLP tasks that aim to extract tree structures from a given text. The most fundamental task in NLP is syntax analysis, which aims to express the syntactic structure of a sentence into a syntactic tree (Zhang, 2020). Parsing often relies on a specific grammar, which is used to refine the output structures of syntax and semantics. Two of the most popular grammars are constituent parsing and dependency parsing. Text2Tree is also seen in many application scenarios. Math word problems (MWPs) (Zhang et al., 2022c; Zhao et al., 2023) extract mathematical expressions from the unstructured texts and try to improve the neural networks' capabilities in math problem solving by asking the model to understand the tree structure. Semantic parsing (Kamath and Das, 2018), which transforms unstructured text into an SQL query, has promising application potential in areas like dialogue systems, search engines, and business intelligence. Our Text2MDT task is novel compared to the literature in the following sense: (a) Text2MDT focuses on extracting medical decision trees from unstructured medical texts. (b) our task has a differ-

ent granularity from the existing Text2Tree tasks since each node in our task consists of one or more triplets. (c) the tree structure, or the links among different nodes, have different meanings from the existing Text2Tree tasks.

Regarding the model architectures for the existing Text2Tree methods, we have seen a trend from idiosyncratic models to more unified model architectures. The field of syntactic analysis has seen many different model architectures, such as recursive neural network (Socher et al., 2011), CRF (Sutton and McCallum, 2010), transition-based models like (Fernandez Astudillo et al., 2020; Zhang et al., 2016), graph-based models (Pei et al., 2015). With the rise of pre-trained encoder models (Devlin et al., 2019), a series of works apply pre-trained models like BERT to enhance the performances on the Text2Tree tasks. For example, (Dozat and Manning, 2017) proposes to install a Biaffine module on top of a pre-trained BERT for the dependency parsing task. This method models the relations among token pairs as a table-filling task and decodes the tree structures of the entire input sequence in one forward pass. With the advances of generative language models, many works apply the pretrained sequence-to-sequence (Seq2Seq) models or GPT style models to Text2Tree tasks (Wang et al., 2018; Zhong et al., 2017). Since the generative models generate sequences that ignore the constraints of the tree, a series of approaches (Xie and Sun, 2019; Yu et al., 2018) are devoted to adding constraints for tree-structured decoders by utilizing the structural information or syntactic rules. In this work, we contribute to the existing literature by systematically evaluating the encoder-based and generation-based methods, especially the open-sourced or commercial LLMs.

3 The Text2MDT Task

3.1 Task formulation

As shown in Figure 1, the Text2MDT task focuses on extracting the medical decision trees from a given text containing the medical decision process from medical guidelines or textbooks. We denote a medical text with n_{text} words as $X = [x_1, x_2, ..., x_{n_{text}}]$, the goal of Text2MDT is to generate the pre-order sequence of n_{node} nodes in the MDT $T = [N_1, N_2, ..., N_{n_{node}}]$. The pre-order sequence of the nodes in the MDT can uniquely represent this tree.

Node structure Nodes in a MDT consist of three parts: role, triplets, and logical relationship between triplets. We denote a node by

$$\begin{aligned} \text{Node} &= \{\text{Role, Triplets, Logical\_Rel}\}, \\ \text{Role} &= \diamond \text{ or } \square, \\ \text{Triplets} &= (t_1, t_2, ..., t_{n_{tri}}), \\ \text{Logical\_Rel} &= \text{and, or, null,} \end{aligned} \tag{1}$$

where: (a) Role denotes the role of the node. Role $= \diamond$ means that the node is a condition node describing certain statuses of patients (presented as diamond-shaped nodes in Figure 1), while Role $= \square$ means that the node is a decision node demonstrating how to treat the patients given certain conditions. (b) Triplets $= (t_1, t_2, ..., t_{n_{tri}})$ denotes the collection of n_{tri} triplets extracted from the given text, where each triplet $t = (sub, rel, obj)$ consists of a subject sub, a relation rel, and a object obj. These triplets are used to describe medical contents, either a patient' medical condition or status, or a medical decision representing the medical procedure to treat the patients. (c) Logical_Rel denotes the logical relationship (and/or/null relation) among the Triplets in a node. Note that the logical relation is null if and only if the number of triplets n_{tri} in the node is less or equal to 1.

Tree structure. A medical decision tree represents the structured process for physicians' decision-making. As depicted in Figure 1, medical professionals need to identify the condition of patients and make the appropriate decisions. Sometimes, medical conditions are complex, so one may have to differentiate many levels of conditions before one can make a valid medical decision. Therefore, we define an MDT as a binary tree consisting of condition and decision nodes, where non-leaf nodes are called conditional nodes, and leaf nodes are decision nodes. For the condition node, when the conditional judgment result is "Yes" ("No"), it will go to the left (right) branch for the following condition judgment or decision. Note that each condition node has left and right child nodes. If the subsequent operation that needs to be done after the result of the condition judgment is "Yes" ("No") is not reflected in the text, a decision node without triplets is used as the left (right) child node. After this operation, a decision tree can be represented by a preorder sequence of its nodes.

Figure 1 shows a concrete example of MDT. In the example, the medical decision process embedded in the medical text above can be modeled by the MDT below: 1) Firstly, the condition "whether

Tree_Depth	Amount	Proportion
2	402	26.80%
3	906	60.40%
4	192	12.80%

Table 1: Statistics of the medical decision tree in Text2MDT dataset.

Relation_Name	Amount	Proportion
clinical_feature	4122	42.51%
therapeutic_drug	2730	28.15%
medical_option	1683	17.36%
usage_or_dosage	666	6.87%
forbidden_drug	249	2.57%
basic_information	246	2.54%

Table 2: Statistics of the triplet relations in Text2MDT dataset.

valproic acid is applicable for patients with generalized tonic-clonic seizures" is determined, and if the result is "Yes," i.e., valproic acid is applicable, then go to the left branch and make the corresponding decision, i.e., valproic acid is used for treatment; 2) if the result is "No," that is, valproic acid is not applicable, next go to the right branch and make another conditional judgment, i.e., the condition "whether the patient has myoclonic seizures or suspected juvenile myoclonic epilepsy" is determined, and go to different branches according to the result.

3.2 Dataset construction

We construct our dataset using two types of resources: (a) clinical guidelines published by authoritative medical institutions about 30 clinical departments from 2011 to 2023; (b) undergraduate clinical medical textbooks published by People's Health Publishing House[2]. The Text2MDT dataset is annotated first by 15 medical school students pursuing master's degrees. Then, a panel of 5 experts will review each sample's annotation. The detailed annotation procedures are described in Appendix B.

3.3 Data Statistics

Table 1 reports the statistics of the tree depth in the Text2MDT dataset. There are 1500 text-tree pairs in the Text2MDT dataset with tree depths equal to 2 to 4. The average number of nodes per tree is 3.76, and the average number of triplets per tree is 6.46. There are 5688 nodes in the dataset. In terms of the nodes' role labels, the dataset includes 2802 decision nodes, 2886 conditional nodes. In terms of the nodes' logical relation labels, the dataset includes 1428 "or" nodes, 1101 "and" nodes, and 3159 "null" nodes. Table 2 reports the statistics of the types of triplet relations in the Text2MDT dataset. Our Text2MDT dataset has six types of relationships with an in-balanced distribution.

3.4 Manual evaluation of quality and usefulness

To evaluate the quality and usefulness of the annotated medical decision tree and whether it can help make medical decisions, we invited ten medical practitioners (with more than two years' work experience in hospitals) and ten people without medical background to complete the following two evaluation tasks: 1) We observed the participants' performance (accuracy and time spent) in answering medical decision problems of similar difficulty under two settings (with medical texts or decision trees as a reference). 2) We asked participants to evaluate the ability of the medical decision trees to represent the medical decision process (completeness, readability, helpfulness).

Most of the participants could answer the decision-making questions more accurately or faster with the help of the MDTs and thought that our annotated MDTs are more readable and helpful for understanding the knowledge of the medical decision process while providing a comprehensive representation of decision knowledge in medical texts. This demonstrates the quality of our annotations and the strength of the decision tree in terms of expressive power. The detailed results of the evaluations are provided in Appendix C.

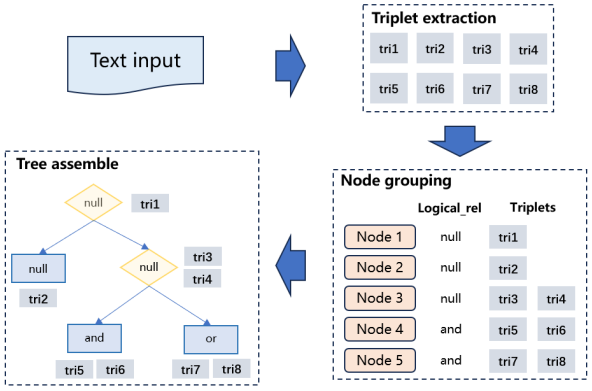

Figure 2: Overview of our pipeline framework. Text2MDT consists of 3 subtasks: triplet extraction, node grouping and tree assembling.

[2] http://www.pph166.com/.

3.5 Evaluation Metrics

In order to evaluate how different models perform on the Text2MDT task, we now define the following evaluation metrics:

- For triplet extraction, we follow (Zhu et al., 2023) to adopt the triplet-level precision (Prec), recall (Rec) and F1 scores as evaluation metrics.
- For node grouping, we define a Levenshtein ratio (Navarro, 2001) style score, NG_LR, for this subtask.
- For the tree assembling subtask and also the whole Text2MDT task, we define three metrics: (a) the accuracy of decision tree extraction (Tree_Acc); (b) the F1 score of decision paths (DP_F1); (c) Lenvenshtein ratio of the decision tree (Tree_LR).

The formal definitions of the above metrics are detailed in Appendix E.

4 Methods of modeling Text2MDT

In this section, we will elaborate on our proposed methods for modeling the task of Text2MDT. First, we will present each module of the pipeline framework for Text2MDT. Then, we will discuss the end-to-end framework.

4.1 Pipelined framework

Figure 2 demonstrates the pipeline for Text2MDT, which consists of three steps: triple extraction, node grouping, and tree assembling.

Triplet Extraction The first step is to extract all the triplets representing either decisions or conditions from medical texts with a unified triplet extraction model TEModel():

$$\{t_1, ..., t_{n_{tri}}\} = \text{TEModel}\left([x_1, \ldots\ldots, x_{n_{text}}]\right), \quad (2)$$

where $t_i = (s_i, r_i, o_i)$ is the i-th triplet in the text, representing a part of a decision or a condition. s_i and o_i are two entity spans from the given text, and r_i is a relation between the two entities and is one of the relation types presented in Table 2.

Node grouping Given the medical text $X = [x_1, \ldots\ldots, x_{n_{text}}]$ and the triplets $\{t_1, ..., t_{n_{tri}}\}$ extracted from this text, we now need to group these triplets into different groups, i.e., nodes, with Logical_Rel $\in$ (and, or, null) (a triple constitutes a group if it has the null relation with other triples). These groups will be the main components of nodes of the MDT.

Tree assembling To assemble the nodes into a medical decision tree, one has to assign a role (condition or decision) to each node and determine whether a pair of nodes is connected. Considering the node's role as the node's named entity label and whether a pair of nodes are connected in the decision tree as a directional relation, the tree assembling task can also be regarded as a joint task of entity type classification and relation extraction.

Note that the Text2MDT task is complex. However, we decompose it into the three subtasks, making it more tractable for relatively traditional encoder-based models like BERT (Devlin et al., 2019). We now present the methods for the subtasks.

Encoder-based pipeline framework The above three subtasks can be addressed by different variants of the Biaffine model (Yu et al., 2020a). For example, triplet extraction is addressed by many recent works like CASREL (Wei et al., 2020), TPLinker (Wang et al., 2020) or UNIRE (Wang et al., 2021), and the above models all utilize a Biaffine-style module on top of a pretrained encoder. For completeness, we present the details on using the Biaffine-based models to deal with the above three subtasks in the Appendix D.

LLM-based pipeline framework We can formulate each subtask of the Text2MDT into a prompt-response generation task. In Appendix H, we present the prompt template and response format for each subtask in the pipeline framework. Note that for the generative LMs like LlaMA-2 to excel at the three tasks, we need to construct the designated datasets for each subtask so that LMs can be finetuned. The details of constructing each subtask's dataset are presented in the Appendix G.

4.2 End-to-end framework

For the end2end framework, due to the complexity of this task, it is challenging for the encoder-based models to deal with the Text2MDT task in an end2end fashion. Thus, we mainly utilize the generative LMs for the end-to-end framework. Since this task is complex, it is natural that the idea of chain-of-thought (COT) (Wei et al., 2022) could benefit our task. In this task, we constructed a series of different COT-style prompts and responses (with prompt and response templates in Appendix H).

Thus, for the end2end framework, we consider the following variations:

direct generation (Gen), in which an LM is asked to generate the final MDT information given the text inputs directly.

COT-Gen-1, which decomposes the Text2MDT task precisely as the pipeline framework and asks the LM first to generate the extracted triplets, then node grouping, and then tree assembly, in a single generation run before generating the end-of-sentence token.

COT-Gen-2 decomposes the task into a more fine-grained subtask. It asks the model to generate entities, triplets, node assignments, node roles, and finally the entire tree.

COT-Gen-3 asks the LM to extract triplets and then generate the whole MDT.

COT-Gen-4 decomposes the triplet extraction subtask by asking the LM to extract entities, then generate the triplets, and finally generate the whole MDT.

5 Experiments

5.1 Implementation Details

Our code was implemented with Pytorch[3] and Huggingface Transformers[4].

For generative LMs, we consider a collection of well-known language models of different sizes. (a) GPT-2 Chinese[5]. (b) Randeng-T5-784M[6]. (c) BLOOMZ-7.1B-mt[7]. (d) ChatGLM-6B-2. (e) ChatMed[8], which is adapted from the LlaMA-7B backbone. (f) Chinese-LLaMA-2 7B/13B[9], which are the Chinese version of LlaMA-2 models (Touvron et al., 2023) from Meta. (g) Ziya-13B-medical[10] is also further pre-trained with the LlaMA-2 models. (h) Baichuan-2 7B/13B models(Yang et al., 2023), which are one of the most recent open-sourced Chinese LLMs, and have achieved excellent performances in many evaluation benchmarks like (Li et al., 2023a). Unless stated otherwise, we will use Baichuan-2 7B as the default generative LM backbone. For generative LMs with parameters fewer than 500 million, we fine-tune all the model parameters. For larger models, we will fine-tune with LoRA (Hu et al., 2021) with rank 24. The LoRA parameters are fine-tuned with a learning rate 1e-4, batch size 16, and warm-up steps of 50. The rest of the hyper-parameters are kept the same with the Transformers package.

For each method, we validate the model performance on the dev set and choose the checkpoint with the best dev performance to predict on the test set. Each experiment is run with different random seeds five times, and the average scores are reported.

The implementation details of the encoder based models are put in Appendix F.

5.2 Datasets

We construct train/dev/test splits for (a) the end2end framework, both in the structural and prompt-response formats. (b) the pipeline framework, where each subtask requires a designated dataset. We put the detailed explanation of constructing the datasets for each subtask to Appendix G, and the prompt-response templates to Appendix H.

5.3 Competing Methods

Encoder-based pipeline framework We now present the competing methods for the encoder-based pipeline framework:

For the triplet extraction subtask, we consider the following methods: (a) UNIRE (Wang et al., 2021); (b) TPLinker (Wang et al., 2020); (c) CasRel (Wei et al., 2020); (d) Sep-Biaffine, which uses a Biaffine model (Yu et al., 2020a) to conduct entity recognition, and another one for relation classification between entity pairs.

For the node grouping subtask, we consider the following methods: (a) the NG-Biaffine method and (b) the NG-TableFill method described in Appendix D.

For the tree assembling subtask, we consider the following methods: (a) TreeAssemble-Biaffine method and (b) TreeAssemb-TableFill described in Appendix D.

To complete the whole task under the pipeline framework, one has to include three models for the three subtasks. We denote the complete pipeline method as Enc-Pipe. Enc-Pipe first uses Sep-Biaffine for triplet extraction, then uses the NG-Biaffine for node grouping, and finally applies

[3] https://pytorch.org/.

[4] https://github.com/huggingface/transformers.

[5] https://huggingface.co/uer/gpt2-chinese-cluecorpussmall

[6] https://huggingface.co/IDEA-CCNL/Randeng-T5-784M-MultiTask-Chinese

[7] https://huggingface.co/bigscience/bloomz-7b1-mt

[8] https://github.com/michael-wzhu/ChatMed

[9] https://github.com/michael-wzhu/Chinese-LlaMA2

[10] https://huggingface.co/shibing624/ziya-llama-13b-medical-lora

Subtask	Triplet extract			Node Grouping	Tree assembling		
Metric	Prec	Rec	F1	NG_LR	Tree_Acc	DP_F1	Tree_LR
Encoder-based methods							
UNIRE	0.913	0.881	0.896				
TPinker	0.909	0.878	0.893				
CasRel	0.882	0.891	0.886				
Sep-Biaffine	0.893	0.897	0.895				
NG-Biaffine				0.962			
NG-TableFilling				0.961			
TreeAssemble-Biaffine					0.735	0.841	0.937
TreeAssemble-TableFilling					0.741	0.838	0.933
Generation-based methods							
Gen	0.901	0.894	0.897	0.965	0.745	0.848	0.943
COT-Gen	0.898	0.904	0.901	0.968	0.748	0.852	0.947
GPT-4 + ICL	0.783	0.815	0.798	0.916	0.672	0.786	0.893

Table 3: Results for each subtask of the pipeline framework, and the overall result of the Text2MDT task when applying the framework. The average results in five different runs are reported. The best results are in bold.

TreeAssemble-Biaffine for the tree assembling subtask.

Generation-based pipeline framework For each step of the generation-based pipeline framework, we consider the COT style generation (COT-Gen) for each subtask. We denote the whole pipeline based on generative LMs as CGen-Pipe, which utilizes the COT-Gen method for each subtask.

To demonstrate the need for fine-tuning for our task, we also compare the method of in-context learning with the currently most powerful commercial LLM, GPT-4 (OpenAI, 2023). For each subtask, we give five demonstration samples randomly selected from the training set to GPT-4 and ask it to make predictions on the samples of the test set. We will denote this method as GPT-4 + ICL.

End2end framework Following Section 4, we consider the following end2end methods: (a) Gen; (b) four variations of COT-style generation, (b1) COT-Gen-1; (b2) COT-Gen-2; (b3) COT-Gen-3; (b4) COT-Gen-4. We also consider GPT-4 + ICL (with five demonstration samples) for the end-to-end generation of medical decision trees.

5.4 Main experimental results

5.4.1 Performances on each subtask

The results of each subtask are reported in Table 3. We can see that: (a) Despite being heavy in model sizes, the Baichuan-2 7B model performs better than the encoder-based models on all the subtasks. The clear advantage of generative models is a unified task format and a unified model architecture. (b) COT-Gen helps the LLMs to achieve better performances on all three sub-tasks in LLM fine-tuning, consistent with the observations of (Zhu et al., 2023). (c) We can see that GPT-4 + ICL can not perform satisfactorily on the three subtasks without fine-tuning.

Method	Tree_Acc	DP_F1	Tree_ER
Pipeline methods			
Enc-Pipe	0.450	0.612	0.884
CGen-Pipe	0.470	0.631	0.897
End2end methods			
Gen	0.440	0.619	0.885
COT-Gen-1	0.470	0.628	0.894
COT-Gen-2	0.450	0.623	0.889
COT-Gen-3	**0.490**	**0.632**	**0.898**
COT-Gen-4	0.450	0.626	0.892
GPT-4 + ICL	0.312	0.529	0.776

Table 4: Overall results of the pipeline framework and the end2end methods. The average results in five different runs are reported. The best results are in bold.

5.4.2 Performances on whole task

We now evaluate the performance of different methods on the whole task. From Table 4, we can see that (a) the CGen-Pipe achieves better performances than the Enc-Pipe method, which is natural since COT-Gen performs better than the encoder-based models on all three subtasks. (b) Interestingly, the pipeline method CGen-Pipe performs better than the Gen method but not better than COT-Gen-3. Intuitively, the pipeline method CGen-Pipe suffers from error propagation from different steps in the pipeline. (c) The COT style generation methods perform better than the direct generation method, which is intuitively sound. Our Text2MDT task is a complex information extraction task containing multiple steps. The COT-based

depth	CGen-Pipe		COT-Gen-3	
	Tree_Acc	DP_F1	Tree_Acc	DP_F1
2	0.750	0.833	0.750	0.833
3	0.428	0.607	0.442	0.603
4	0.454	0.648	0.545	0.676

Table 5: Model performance on medical decision trees of different depths.

Backbone	Tree_Acc	DP_F1	Tree_ER
The Enc-Pipe method			
MedBERT	0.450	0.612	0.884
BERT-www-ext	0.440	0.615	0.882
BERT-base Chinese	0.390	0.583	0.867
Erlangshen-ZEN1	0.410	0.596	0.873
The COT-Generation-3 method			
GPT-2 base Chinese	0.030	0.121	0.238
Randeng-T5-784M	0.080	0.253	0.352
BLOOMZ-7.1B-mt	0.330	0.536	0.782
ChatGLM-6B-2	0.380	0.592	0.849
ChatMed	0.420	0.596	0.864
Chinese-LlaMA-2 7B	0.410	0.581	0.868
Chinese-LlaMA-2 13B	0.460	0.623	0.890
Ziya-13B-medical	0.450	0.614	0.886
Baichuan2 7B	**0.490**	**0.632**	**0.898**
Baichuan2 13B	0.490	0.628	0.896

Table 6: The effects of the pre-trained backbones on the Enc-Pipe and COT-Generation-3 methods.

generative methods inject priors on how the models should solve the task. Thus, LLMs can be more informed to use the results of the previously generated contents for future token generation. (d) Intuitively, the generative LMs should benefit more from detailed and fine-grained COT instructions. However, Table 4 shows that COT-Gen-3 performs the best. COT-Gen-3's thought steps have a relatively smaller response length, which is helpful for the LMs to keep track of the generation contents. (e) with in-context learning, GPT-4 performs relatively worse than the fine-tuned open-sourced LLM.

5.5 Discussions and further analysis

Impact of tree depth In table 5, we present the results of CGen-Pipe, and COT-Gen-3 on different MDT depths. We can see that the two methods obtain the same performance metrics on the MDTs with depth 2. The performance difference between the two methods mainly lies in MDTs with higher depth. We can see that the performances on the MDTs with a depth larger than 2 are significantly worse than those on the MDTs with a depth of 2.

Impact of backbone models Table 6 reports the experimental results for different backbone models, and the following observations can be made: (a) for the Enc-Pipe method, the in-domain pre-trained model, MedBERT performs the best among the four pre-trained encoders, showing that further in-domain pretraining is beneficial. This observation aligns with (Zhu, 2021b; Guo et al., 2021a; Zhu et al., 2023b). (b) For the generative LMs, models with small parameter sizes perform unsatisfyingly in our task. Among the open-sourced generative LMs we experiment with, the Baichuan2 models perform the best. Baichuan2's advantage results from its large-scale pretraining and complete instruction alignment pipeline.

Case studies On the test set of the Text2MDT task, COT-Generation-3 achieves the best performance. Figure 16 and 17 (in Appendix J) report two examples where COT-Generation-3 can not predict the same MDTs with the ground truth. In Figure 16, COT-Generation-3 misses the triplet (患者, 治疗药物, 缓解充血药) ((patient, treatment, decongestant)) in the second node, and the triplet (患者, 治疗药物, 退热药) ((patient, therapeutic drug, antipyretic drug)) in the fourth node, during prediction. These errors are mainly from the triplet extraction subtask, the first step of tackling MDTs. In Figure 17, COT-Generation-3 made an error in triplet extraction regarding the basic status of the patients and, as a result, made a mistake in node grouping.

6 Conclusion

In this study, we propose a novel task, Text2MDT, which aims to automatically extract medical decision trees from medical texts that are significant for intelligent medicine. We constructed the first Text2MDT dataset in the NLP community with the participation of medical experts. Since there are no existing neural network-based methods that can directly deal with our novel tasks, we propose two cohorts of methods: (a) the pipeline-based method, which decomposes the Text2MDT task into three subtasks and utilizes the existing methods to complete the subtasks; (b) the end2end method, which is challenging and can not be handled by the encoder-based models. We utilize the recent open-sourced LLMs and chain-of-thought prompting for the end-to-end methods. Experiments show that the LLMs can achieve promising results on the Text2MDT benchmark end-to-end with the help of chain-of-thought prompting.

Limitations

Our work is the first exploration of extracting MDTs from medical texts, and our work is currently applicable to some simple scenarios, specifically: 1) The logic expression of nodes is limited. The triplets between nodes are only "and" and "or," while in more complex scenarios, there should be a combination of multiple logical relationships; 2) The expressiveness of the tree is limited—our decision tree aborts after reaching a decision. The actual scenario should be a process of continuous judgment and decision-making. 3) The length of the text is limited. We only contend with extracting one paragraph of medical text; in fact, much medical knowledge must be based on multiple sections or chapters. We will improve on these shortcomings in our future work.

Ethics Statement

This study, focusing on developing a dataset and methodologies for extracting medical decision trees from medical texts, is conducted carefully, considering ethical principles and potential risks associated with the research.

Firstly, it is essential to note that the dataset utilized in this study is derived from medical textbooks and guidelines and, thus, does not contain any personally identifiable information. However, ethical considerations regarding patient privacy and confidentiality remain paramount despite the absence of direct personal information. We have taken measures to ensure that no sensitive patient data is included in the dataset and that all information extracted is solely for research purposes.

Furthermore, the participation of medical experts in constructing the Text2MDT dataset is essential for ensuring the accuracy and relevance of the data. We have obtained informed consent from all contributors, emphasizing the voluntary nature of their participation and the intended use of the dataset for research purposes.

Moreover, while our study focuses on advancing the field of intelligent medicine through developing novel techniques, we acknowledge the importance of transparency and accountability in AI-driven healthcare applications. As such, we are committed to openly sharing our findings, methodologies, and datasets with the research community, facilitating peer review, reproducibility, and further ethical scrutiny.

In conclusion, this study underscores our commitment to upholding ethical standards in research, particularly in healthcare and artificial intelligence. By proactively addressing potential risks and ethical considerations, we aim to contribute responsibly to advancing medical knowledge and technology.

References

Ajith Abraham. 2005. Rule-based expert systems. *Handbook of measuring system design*.

Merijn Beeksma, Suzan Verberne, Antal van den Bosch, Enny Das, Iris Hendrickx, and Stef Groenewoud. 2019. Predicting life expectancy with a long short-term memory recurrent neural network using electronic medical records. *BMC medical informatics and decision making*, 19(1):1–15.

Yan-fen CHENG, Jia-jun WU, and Fan HE. 2023. Aspect level sentiment analysis based on relation gated graph convolutional network. *Journal of ZheJiang University (Engineering Science)*, 57(3):437–445.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1001–1007, Online. Association for Computational Linguistics.

D. Fotiadis, Y. Goletsis, A. Likas, and A. Papadopoulos. 2006. Clinical decision support systems. *John Wiley & Sons, Inc.*

Xiangxiang Gao, Wei Zhu, Jiasheng Gao, and Congrui Yin. 2023. F-pabee: Flexible-patience-based early exiting for single-label and multi-label text classification tasks. *ArXiv*, abs/2305.11916.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Zhao Guo, Yuan Ni, Keqiang Wang, Wei Zhu, and Guotong Xie. 2021a. Global attention decoder for Chinese spelling error correction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1419–1428, Online. Association for Computational Linguistics.

Zhao Guo, Yuan Ni, Keqiang Wang, Wei Zhu, and Guotong Xie. 2021b. Global attention decoder for chinese spelling error correction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1419–1428.

Udo Hahn and Michel Oleynik. 2020. Medical information extraction in the age of deep learning. *Yearbook of medical informatics*, 29(01):208–220.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Degree: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908.

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*.

Mark Hughes, Irene Li, Spyros Kotoulas, and Toyotaro Suzumura. 2017. Medical text classification using convolutional neural networks. In *Informatics for Health: Connected Citizen-Led Wellness and Population Health*, pages 246–250. IOS Press.

IoannisVourgidis, ShadreckJosephMafuma, PaulWilson, JennyCarter, and GeorginaCosma. 2018. Medical expert systems – a study of trust and acceptance by healthcare stakeholders. *Springer, Cham*.

Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V. Chawla, and Meng Jiang. 2019. The role of: A novel scientific knowledge graph representation and construction model. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1634–1642. ACM.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *ArXiv*, abs/1812.00978.

Mohamed Yassine Landolsi, Lobna Hlaoua, and Lotfi Ben Romdhane. 2023. Information extraction from electronic medical documents: state of the art and future research directions. *Knowledge and Information Systems*, 65(2):463–516.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Bo Li, Dingyao Yu, Wei Ye, Jinglei Zhang, and Shikun Zhang. 2022. Sequence generation with label augmentation for relation extraction. *arXiv preprint arXiv:2212.14266*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023a. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Yang Li, Buyue Qian, Xianli Zhang, and Hui Liu. 2020. Graph neural network-based diagnosis prediction. *Big Data*, 8(5):379–390.

Zihao Li, Mosha Chen, Kangping Yin, Yixuan Tong, Chuanqi Tan, Zhenzhen Lang, and Buzhou Tang. 2023b. Chip2022 shared task overview: Medical causal entity relationship extraction. In *Health Information Processing. Evaluation Track Papers*, pages 51–56, Singapore. Springer Nature Singapore.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Alencar Machado, Vinícius Maran, Iara Augustin, Leandro Krug Wives, and José Palazzo Moreira de Oliveira. 2017. Reactive, proactive, and extensible situation-awareness in ambient assisted living. *Expert Systems with Applications*, 76:21–35.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33:31–88.

Rimpy Nohria. 2015. Article: Medical expert system-a comprehensive review. *International Journal of Computer Applications*, 130(7):44–50. Published by Foundation of Computer Science (FCS), NY, USA.

OpenAI. 2023. GPT-4 Technical Report. *arXiv e-prints*, page arXiv:2303.08774.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2015. An effective neural network model for graph-based dependency parsing. In *Annual Meeting of the Association for Computational Linguistics*.

Aurora Saibene, Michela Assale, and Marta Giltri. 2021. Expert systems: Definitions, advantages and issues in medical field applications. *Expert Systems with Applications*, 177:114900.

Richard Socher, Cliff Chiung-Yu Lin, A. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *International Conference on Machine Learning*.

Haixia Sun, Jin Xiao, Wei Zhu, Yilong He, Sheng Zhang, Xiaowei Xu, Li Hou, Jiao Li, Yuan Ni, and Guotong Xie. 2020. Medical knowledge graph to enhance fraud, waste, and abuse detection on claim data: Model development and performance evaluation. *JMIR Med Inform*, 8(7):e17653.

Charles Sutton and Andrew McCallum. 2010. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4:267–373.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

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

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating math word problem to expression tree. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1064–1069. Association for Computational Linguistics.

Mengqiu Wang and Daniel Cer. 2012. Stanford: Probabilistic edit distance metrics for STS. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 648–654, Montréal, Canada. Association for Computational Linguistics.

Y. Wang, B. Yu, Y. Zhang, T. Liu, and L. Sun. 2020. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. *Proceedings of the 28th International Conference on Computational Linguistics*.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. Unire: A unified label space for entity relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 220–231. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1476–1488. Association for Computational Linguistics.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5299–5305. ijcai.org.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. *ArXiv*, abs/2106.01223.

Ai Ming Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Hai Zhao, Hang Xu, Hao-Lun Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kuncheng Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Pei Guo, Ruiyang Sun, Zhang Tao, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yan-Bin Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *ArXiv*, abs/2309.10305.

Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, and Zhiwei Wang. 2022. A survey of information extraction based on deep learning. *Applied Sciences*, 12(19):9691.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020a. Named entity recognition as dependency parsing. In *Annual Meeting of the Association for Computational Linguistics*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020b. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir R. Radev. 2018. Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1653–1663. Association for Computational Linguistics.

Meishan Zhang. 2020. A survey of syntactic-semantic parsing based on constituent and dependency structures. *Science China Technological Sciences*, 63:1898 – 1920.

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, and Weiming Lu. 2022a. De-bias for generative extraction in unified ner task. In *Annual Meeting of the Association for Computational Linguistics*.

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, and Weiming Lu. 2022b. De-bias for generative extraction in unified NER task. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 808–818, Dublin, Ireland. Association for Computational Linguistics.

Wenqi Zhang, Yongliang Shen, Yanna Ma, Xiaoxia Cheng, Zeqi Tan, Qingpeng Nong, and Weiming Lu. 2022c. Multi-view reasoning: Consistent contrastive learning for math word problem. In *Conference on Empirical Methods in Natural Language Processing*.

Xinpeng Zhang, Ming Tan, Jingfan Zhang, and Wei Zhu. 2023a. Nag-ner: a unified non-autoregressive generation framework for various ner tasks. In *Annual Meeting of the Association for Computational Linguistics*.

Xun Zhang, Yantao Du, Weiwei Sun, and Xiaojun Wan. 2016. Transition-based parsing for deep dependency structures. *Computational Linguistics*, 42(3):353–389.

Yuming Zhang, Xiangxiang Gao, Wei Zhu, and Xiaoling Wang. 2023b. Fastner: Speeding up inferences for named entity recognition tasks. In *International Conference on Advanced Data Mining and Applications*.

Zhexi Zhang, Wei Zhu, Junchi Yan, Peng Gao, and Guowang Xie. 2021. Automatic student network search for knowledge distillation. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2446–2453.

Xu Zhao, Yuxi Xie, Kenji Kawaguchi, Junxian He, and Qizhe Xie. 2023. Automatic model selection with large language models for reasoning. In *Conference on Empirical Methods in Natural Language Processing*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.

Binggui Zhou, Guanghua Yang, Zheng Shi, and Shaodan Ma. 2021. Natural Language Processing for Smart Healthcare. *arXiv e-prints*, page arXiv:2110.15803.

Xiaofeng Zhou, Yuan Ni, Guo Tong Xie, Wei Zhu, Cai Chen, Tianhao Wang, and Zhi-Gang Pan. 2019. Analysis of the health information needs of diabetics in china. *Studies in health technology and informatics*, 264:487–491.

Wei Zhu. 2021a. AutoRC: Improving BERT based relation classification models via architecture search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 33–43, Online. Association for Computational Linguistics.

Wei Zhu. 2021b. MVP-BERT: Multi-vocab pre-training for Chinese BERT. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 260–269, Online. Association for Computational Linguistics.

Wei Zhu. 2021c. Mvp-bert: Multi-vocab pre-training for chinese bert. In *Annual Meeting of the Association for Computational Linguistics*.

Wei Zhu, Yilong He, Ling Chai, Yuanchun Fan, Yuan Ni, Guo Tong Xie, and Xiaoling Wang. 2021a. paht_nlp @ mediqa 2021: Multi-grained query focused multi-answer summarization. In *Workshop on Biomedical Natural Language Processing*.

Wei Zhu, Wenfeng Li, Xiaoling Wang, Wendi Ji, Yuanbin Wu, Jin Chen, Liang Chen, and Buzhou Tang. 2023a. Extracting decision trees from medical texts: An overview of the text2dt track in chip2022. In *Health Information Processing. Evaluation Track Papers*, pages 89–102, Singapore. Springer Nature Singapore.

Wei Zhu, Yuan Ni, Xiaoling Wang, and Guotong Xie. 2021b. Discovering better model architectures for medical query understanding. In *Proceedings of the*

2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, pages 230–237, Online. Association for Computational Linguistics.

Wei Zhu, Xipeng Qiu, Yuan Ni, and Guotong Xie. 2020. AutoRC: Improving BERT Based Relation Classification Models via Architecture Search. *arXiv e-prints*, page arXiv:2009.10680.

Wei Zhu, Peifeng Wang, Xiaoling Wang, Yuan Ni, and Guo Tong Xie. 2023b. Acf: Aligned contrastive finetuning for language and vision tasks. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Wei Zhu and Xiaoling Wang. 2023. Chatmed: A chinese medical large language model. https://github.com/michael-wzhu/ChatMed.

Wei Zhu, Xiaoling Wang, Yuan Ni, and Guotong Xie. 2021c. Autotrans: Automating transformer design via reinforced architecture search. In *Natural Language Processing and Chinese Computing*, pages 169–182, Cham. Springer International Publishing.

Wei Zhu, Xiaoling Wang, Huanran Zheng, Mosha Chen, and Buzhou Tang. 2023. PromptCBLUE: A Chinese Prompt Tuning Benchmark for the Medical Domain. *arXiv e-prints*, page arXiv:2310.14151.

Wei Zhu, Xiaofeng Zhou, Keqiang Wang, Xun Luo, Xiepeng Li, Yuan Ni, and Guotong Xie. 2019. PANLP at MEDIQA 2019: Pre-trained language models, transfer learning and knowledge distillation. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 380–388, Florence, Italy. Association for Computational Linguistics.

A Appendix: additional related work

A.1 Medical natural language processing

The developments in neural networks and natural language processing has advanced the field of medical natural language processing (MedNLP) (Zhou et al., 2021; Hahn and Oleynik, 2020; Zhu et al., 2021b). In the pre-BERT era, firstly, RNNs like LSTM/GRU are used for processing sequential medical data such as text and speech (Beeksma et al., 2019). Convolutional networks are also used for medical text classificaiton (Hughes et al., 2017). The techniques of Graph neural networks are also explored for diagnose recommendations (Li et al., 2020). In this period, many different model architectures are specially designed for better performances on a specific MedNLP task (Zhu et al., 2021b,c; Zhang et al., 2021). Since BERT (Devlin et al., 2019), the pretrained language models (PLMs) become the deafult solution for MedNLP. In this stage, researchers become less interested in modifying the model architecture, but instead trying to pretrain or further pretrain a PLM from the open domain to the medical domain (Guo et al., 2021b; Zhu, 2021b; Gu et al., 2020). With the wide study of LLMs, the field of MedNLP is also being revolutionized. There are already works on adapting LLM backbones to the medical domain question answering (Zhu and Wang, 2023). And (Zhu et al., 2023) propose PromptCBLUE, a prompt learning based benchmark dataset for examing the LLMs' ability in MedNLP tasks. This work can also serve as a testbed for the current commercial or open-sourced LLMs, since the complexity of our novel task will pose great challenges for them.

A.2 Information extraction from medical texts

Information Extraction (IE) is a research topic of long history that aims to extract structured knowledge or factual information from unstructured texts (Yang et al., 2022). The field of IE includes a wide range of tasks, such as named entity recognition (Das et al., 2022; Landolsi et al., 2023), relation extraction (RE) (Zhu et al., 2020; Li et al., 2022), event extraction (Hsu et al., 2022), aspect-level sentiment analysis (CHENG et al., 2023). Since the raise of pre-trained models like BERT (Devlin et al., 2019), the performances on IE tasks have advanced greatly (Zhu, 2021b). But one has to have different model structures for different fine-grained IE tasks, for instance, the SOTA nested NER models (Zhang et al., 2022a) are different from those of discontinuous NER tasks (Zhang et al., 2022b). Recently, there is a trend that all the IE task should be solved by a unified paradigm, that is, Seq2Seq generation. (Yan et al., 2021) proposes the framework of BartNER which solves all types of NER tasks with a BART model (Lewis et al., 2019). UIE (Lu et al., 2022) takes a step ahead and proposes to use prompts and a unified structural language to deal with many types of IE tasks with a single model checkpoint.

Medical information extraction is an important research field, and it has broad applications like medical search engine, automatic electronic health record analysis, online health consultation, and medical knowledge graph construction (Sun et al., 2020; Guo et al., 2021a; Zhu et al., 2019; Zhou et al., 2019; Zhu et al., 2021b,a; Zhang et al., 2023a). Compared with open-domain IE tasks, the IE tasks are known for their complexity. For exam-

ple, discontinuous or nested entities are common in the medical field. And knowledge in the medical domain may be too complex to be expressed as triplets (Zhu et al., 2023a). For example, (Jiang et al., 2019) introduced the role of "condition" and argued that a fact triplet is established based on some conditional triplets in the biomedical field. In the CMedCausal (Li et al., 2023b) task, a triplet may be the result of a subject conducting certain behaviour, expressing the causal relations. With the rise of LLMs, the research field of IE and medical IE is also under revolution. In this work, we compliment the existing literature by constructing the challenging Text2MDT task, where not only triplets have to be extracted, but also they need to arranged into nodes of a binary tree to express a complex medical decision process.

B Appendix for dataset construction

Resources We choose clinical practice guidelines and clinical medicine textbooks as our data sources. Clinical practice guidelines are systematically developed multidisciplinary clinical guidelines that help clinicians, patients, and other stakeholders make appropriate management, selection, and decisions about specific clinical issues. Clinical medicine textbooks are the primary means medical students acquire medical knowledge and can be used as a reference for clinical decision-making. We collected over 500 clinical guidelines published by authoritative medical institutions and about 30 clinical departments from 2011 to 2021 and over 100 undergraduate clinical medical textbooks published by People's Health Publishing House[11] to build our dataset. We obtain the informed consents from the resources' owners.

Since medical texts are long and contain rich and various medical knowledge, we used section-based filtering and trigger/template-based filtering to locate segments of medical texts that contain the medical decision process based on the analysis of medical texts and the help of specialized doctors. First, we selected the chapters with a high density of medical decision knowledge, such as "Treatment", "Drug Selection" and "Medical Solutions" in the source data. Then, we analyzed and summarized the structure and pattern of the medical decision text construct templates and trigger words for medical decision knowledge. We filtered the text based on the template and triggers to obtain the text fragments containing the knowledge of the medical decision process.

Annotation procedures Our data collection protocols are approved by our institution's ethics review board. And we recruit our annotators from a medical school in Shanghai. Annotators of our dataset include (a) 15 annotators who are master students from medical schools and (b) five medical experts with medical doctoral degrees, more than ten years of clinical experience, and at least two years of experience with medical text data annotation. All the annotators have been instructed with detailed and formal annotation principles for at least two hours, including understanding the medical decision-making process, the judgment of logical relationships, and the annotation specifications of triplets and decision trees. Every three annotators will form a group, and they first independently annotate each text and revise the initial annotation after discussion inside the group. Medical experts will examine their annotations. If the five experts agree on the annotation unanimously, the annotation enters the dataset collection. If not, they will provide feedback on improvement, and the annotation group will revise the annotation until approval.

Furthermore, we calculate Cohen's Kappa (Cohen, 1960) to measure the agreements between each pair of annotators. The Kappa coefficient for triplet annotation is 0.83 before in-group discussion or experts' feedback and 0.94 after. The Kappa coefficient for the whole medical decision tree annotations is 0.65 before in-group discussion or experts' feedback and 0.83 after. The results ensure the annotation consistency of our Text2MDT benchmark.

C Appendix: Manual Evaluation of Annotated MDTs

The detail of our manual evaluation of medical decision trees are as follows:

1. We observed the participants' performance on medical decision problems of similar difficulty under medical texts and MDTs. Specifically, participants will answer three sets of medical decision questions, each group providing texts or decision trees containing the medical knowledge needed to answer the medical decision question. We observe their accuracy and time spent answering the decision question. Each set of questions is randomly selected from the question pool and is guaranteed to be of similar difficulty.

[11] http://www.pph166.com/.

2. We invited participants to rate medical texts and MDTs in terms of readability, completeness, and helpfulness. Specifically, we randomly selected five medical texts and MDTs expressing the same knowledge. We asked participants to score (0-3) them in terms of whether they were clear and easy to understand (readability), whether they were comprehensive and detailed (completeness), and whether they were helpful in understanding or studying medical knowledge (helpfulness).

	A	T	R	C	H
Text	0.64	31.5	2.26	2.70	2.33
DT	0.86	25.4	2.74	2.72	2.62
Text	0.94	21.6	2.50	2.74	2.68
DT	0.94	18.4	2.66	2.62	2.76

Table 7: Results of manual evaluation of annotated MDTs. The results in the first field are for subjects without medical background, and the results in the second field are for medical practitioners. **A** represents the average accuracy of answering the medical decision questions. **T** represents the average seconds spent answering the medical decision questions. **R**, **C**, and **H** represent the readability, completeness, and helpfulness average scores.

The results of the manual evaluation are shown in Table 7. We can draw the following conclusions:

For subjects without medical background, the medical decision tree helped them make more correct decisions in less time compared with the medical text and gained the highest scores for readability, completeness, and helpfulness. Theoretically, the completeness of the medical text should be better than the medical decision tree. Still, due to the poor readability of the medical text, the subjects may not have gained complete access to the knowledge contained in the medical text.

For medical practitioners, the medical decision tree group achieved the same accuracy on the medical decision questions as the medical text group, but the former took less time. The medical decision trees gained the highest readability and helpfulness scores and slightly lower completeness than the medical texts. The results demonstrate that the medical decision tree can help people make treatment decisions faster and better and can model medical decision knowledge clearly and intuitively, which can help readers better understand medical decision knowledge.

D Details of the encoder based models for the subtasks

D.1 Triplet Extraction

Triplet extraction is widely studied task (Zhu, 2021a; Gao et al., 2023; Zhu, 2021c; Zhu et al., 2021c), and there are many recent works that can be utilized to complete this subtask. One line of work is based on semantic encoders like BERT (Devlin et al., 2019) and a table-filling module (Dozat and Manning, 2016; Zhang et al., 2023b). The representative methods in this direction is: CASREL (Wei et al., 2020), TPLinker (Wang et al., 2020) and UNIRE (Wang et al., 2021). For completeness, we now demonstrate how UNIRE (Wang et al., 2021) applies a biaffine module to complete the entity mention detection and relation classification tasks simultaneously.

With a given sentence input X, a pre-trained encoder like BERT or RoBERTa will encode the semantic information and provide hidden representations for $X^{'}$. Denote the hidden vector corresponding each token x_i as $h_i \in \mathcal{R}^d$. Denote the set of entity types as $\mathcal{K}_e$, and the set of relation types as $\mathcal{K}_r$. UNIRE targets at identifying the label $l_{i,j}$ of each token pair (i, j). That is, if the token pair (i, j) is classified as an entity type $k_e \in \mathcal{K}_e$, we will consider the text span starting from the i-th token and ending at the j-th token as an entity of type k_e. And if the token pair (i, j) is classified as an relation type $k_r \in \mathcal{K}_r$, and token i and j are the starting tokens of two entity mentions, we will consider that these two entities have a relation of type k_r. To complete the two tasks with a single calculation step, the UNIRE construct a biaffine module which maps each token pair (i, j) to a probability distribution of dimension $K = |\mathcal{K}_e| + |\mathcal{K}_r| + 1$: [12]

$$P(l_{i,j}) = \text{Biaffine}\left(h_i, h_j\right), \quad (3)$$

where Biaffine() is given by

$$\text{Biaffine}(h_1, h_2) = h_1^T U h_2 + W\left(h_1 \oplus h_2\right), \quad (4)$$

Since we need to calculate the scores for K categories, U is a $d \times K \times d$ tensor, and W is a $2d \times K$ tensor.[13] Since the above method is analogeous as

[12] Adding 1 for the null type.

[13] Note that in the BERT biaffine NER (Yu et al., 2020b), two feed forward layers are designated to transform the two features passing to the biaffine module. However, we find that dropping the two feed forward layers will not result in significant performance changes.

filling in a $n_{text} \times n_{text}$ sized table, we often refer to the biaffine method as the table-filling method. Denoting the ground truth of $l_{i,j}$ as $y_{i,j}$, then the training objective is the summation of cross-entropy loss at each of

$$\mathcal{L} = -\frac{1}{|n_{text}|^2} \sum_{i=1}^{|n_{text}|} \sum_{j=1}^{|n_{text}|} \log P\left(l_{i,j} = y_{i,j}\right). \tag{5}$$

After the above BERT-based biaffine model is trained, the inference procedure follows UNIRE (Wang et al., 2021).

D.2 Node grouping

Given the medical text $X = [x_1, ..., x_{n_{text}}]$ and the triplets $\{t_1, ..., t_{n_{tri}}\}$ extracted from this text, we now need to group these triplets into different groups, i.e., nodes, with relation $l \in (and, or, null)$ (a triple constitutes a group if it has the $null$ relation with other triples). These groups will be the main components of nodes of the MDT.

Now we will demonstrate the model for this subtask: node-grouping biaffine (NG-Biaffine), which is to adapt the idea of biaffine model to the node grouping task. Note that if a triple belongs to a node with relation $l \in \mathcal{K}_{NG}$ (where $\mathcal{K}_{NG} =$ and, or, null is the set of the logical relations among triplets.), it will have relation l with any other triplet within the group and $null$ relation with other triplets in the other groups. Thus, the key step for node grouping is to determine the relationships among the triplets, which can be conveniently modeled by a table-filling task similar to Equation 3. Denote the augmented text input as $X^{'} = [X, [\text{t}], t_1, ..., [\text{t}], t_{n_{tri}}]$, where [] denotes the text concatenation operation. Note that we add a special token [t] before each triplet. A pre-trained encoder like BERT or RoBERTa will encode the semantic information and provide hidden representations for $X^{'}$, and obtain the semantic representation of triplet t_i by taking the hidden vector corresponding the special token right before t_i (denoted as $h(t_i)$). Then a biaffine module will handle the classification task for each triplet pair (t_i, t_j) by calculating its probability $P(l_{t_i,t_j})$ distribution over all the relation categories.

During inference, we will consider a score based decoding procedure for resolving possible conflicts. For each triplet pair (t_i, t_j), its label l_{t_i,t_j} is obtained by choosing the relation category that receives the highest probability mass. And denote the probability mass of l_{t_i,t_j} as m_{t_i,t_j}. During inference, we first calculate m_{t_i,t_j} and l_{t_i,t_j} for each triplet pair (t_i, t_j) in a single forward pass. And we rank l_{t_i,t_j} by m_{t_i,t_j}. The relation l_{t_i,t_j} that receives the highest m_{t_i,t_j} value will first be established, and any conflicting relation predictions with lower scores will be rejected. Here, a conflict arises when a triplet t_i has the and relation with t_j, but also has the or relation with another triplet $t_{j'}$. Then we will establish the relation prediction with the second highest probability mass that has not been discarded. Repeting the above procedures till all the triplets are included in the established relations, and we will have the complete prediction for node grouping. The logical relation for each node will be the relation type among the triplets inside the node.

Note that we can consider a variant of the NG-Biaffine model, NG-TableFill, which substitute the biaffine module (Equation 3) in the NG-biaffine method to the table-filling module in (Wang et al., 2020) (Equation 1 of (Wang et al., 2020)).

D.3 Tree assembling

Note that in the above procedure, we already has the nodes in the decision tree. To assemble the nodes to a medical decision tree, one has to assign a role (condition or decision) to each node, and determine whether a pair of nodes are connected. Considering the node's role as the node's named entity label, and whether a pair of nodes are connected in the decision tree as a directional relation, the tree assembling task can also be regarded as a joint task of entity type classification and relation extraction.

We now elaborate on the model details for tree assembling. Denote each unclassified node as Node_i (i = 1, 2, ..., n_{node}). We formulate each node as a text sequence by concatenating the logical relation name, role label name, and triplets' text contents, and we augment the text input X to

$$X^{'} = [X, [\text{n}], \text{Node}_i, ...[\text{n}], \text{Node}_{n_{node}}]$$

, where [] denotes the text concatenation operation. Note that we add a special token [n] before each node. After being encoded with a pre-trained text encoder, we can obtain $h(\text{Node}_i)$, the hidden states of the special token [n] right before each node. $h(\text{Node}_i)$ is considered as the semantic representation of Node_i. A simple linear layer can operate as the node type prediciton module, and a

biaffine module will handle the relation classification task for each node pair $(\text{Node}_i, \text{Node}_j)$. During decoding, we employ the strategy described in (Dozat and Manning, 2016) to resolve conflicting predictions. We will refer to the above model as TreeAssemble-Biaffine.

Note that we can consider a variant of the TreeAssemble-Biaffine model, TreeAssemble-TableFill, which substitute the biaffine module (Equation 3) in the TreeAssemble-biaffine method to the table-filling module in (Wang et al., 2020) (Equation 1 of (Wang et al., 2020)).

E Appendix: detailed explanations of the evaluation metrics

E.1 Metrics for the triplet extraction subtask

As described in Section 4, the most fundamental step of Text2MDT is to extract triples from the given text documents. Following (Zhu et al., 2023) and (Zhu, 2021a), we adopt the **triplet precision, recall and F1** scores as evaluation metrics. These metrics of triplet extraction are instance-level strict performance metrics. Here, an instance means a complete piece of information extracted from the given document. In our triplet extraction subtask, an instance consists of a head entity mention, a tail entity mention, and the relation label name between these two entities. And strict means that the model predicts an instance correctly if and only if it correctly predicts the all the components of the instance.

E.2 Metrics for the node grouping subtask

Following (Wang and Cer, 2012), we now define an edit distance based metric to evaluate how models perform in the node assignment task. According to Equation 1, one can express a predicted node N^{pred} to a tuple.

$$N^{pred} = (\text{Role}^{pred}, t_1^{pred}, ..., t_{n_{tri}}^{pred}, \text{Logical\_Rel}^{pred}). \tag{6}$$

Note that we treat each triplet in the same level with the node role label and the logical relation label. And denote a node in the ground truth as

$$N^{gt} = (\text{Role}^{gt}, t_1^{gt}, ..., t_{n_{tri}}^{gt}, \text{Logical\_Rel}^{gt}). \tag{7}$$

Treating each element in the N^{pred} and N^{gt} tuples as indivisible, one can calculate the edit distance between N^{pred} and N^{gt}. In this scenario, the editing operations include inserting and deleting elements, and each operation has a cost of 1. Now we concatenate all the nodes in the node grouping prediction into a single tuple NG_Tup^{pred}. Since we does not require the model to assign orders to each node in the node grouping step, we consider all the permutation m of nodes in the ground truth MDT^{gt}, and we concatenate the nodes in each permutation (denoted as $\text{NG\_Tup}^{gt,m}$). And the edit distance between the whole node assignment prediction and the ground truth node assignment is defined as the minimum edit distance between the predicted node grouping and a permutation of the ground truth node grouping:

$$\begin{aligned}&\text{NG\_ED}(\text{NG\_Tup}^{pred}, \text{MDT}^{gt}) \\ =&\min_{m \in \text{Permute}(\text{MDT}^{gt})} \text{ED}(\text{NG\_Tup}^{pred}, \text{NG\_Tup}^{gt,m}),\end{aligned} \tag{8}$$

where $\text{ED}(x, y)$ denotes the edit distance between tuple x and tuple y. Since the edit distance score NG_ED is an un-normalized metric, it is in-suitable for model comparisons. Thus, we now define the Levenshtein ratio (Navarro, 2001) (denoted as NG_LR) for the node grouping subtask:

$$\begin{aligned}&\text{NG\_LR}(\text{NG\_Tup}^{pred}, \text{MDT}^{gt}) \\ =&\frac{\text{NG\_ED}(\text{NG\_Tup}^{pred}, \text{MDT}^{gt})}{\max(\text{len}(\text{NG\_Tup}^{pred}), \text{len}(\text{NG\_Tup}^{gt,m^*})}\end{aligned} \tag{9}$$

where len denotes the tuple length, and m^* is the MDT^{gt}'s permutation that obtains the lowest edit distance with the prediction:

$$m^* = \arg\min_{m \in \text{Permute}(\text{MDT}^{gt})} \text{ED}(\text{NG\_Tup}^{pred}, \text{NG\_Tup}^{gt,m}). \tag{10}$$

E.3 Metrics for the tree assembling subtask

To properly evaluate a model's performance in constructing medical decision trees from text, we adopt the following three evaluation metrics:

- The accuracy of decision tree extraction (Tree_Acc). For this metric, the instance is the entire medical decision tree consisting of a series of nodes connected as a binary tree of a certain structure, and each node contains three components, logical relation, role and

triplets. A decision tree predicted by a model is correct when it is precisely the same as the ground truth. Thus, this metric is a very strict metric.

- F1 score of decision paths (DP_F1). We define a decision path in a medical decision tree as a path from the root node to a leaf node. Thus, in DPF1, an instance is a decision path, and a model correctly predicts a decision path if and only if it correctly predicts all the nodes in the path and how they are connected.

- Lenvenshtein ratio of the decision tree (Tree_LR). Similar to the definition of edit ratio defined for the node grouping task, we can arrange the contents of all nodes in the predicted or ground-truth tree into a single tuple in the in the order of depth-first search (denoted as Tree_Tup^{pred} and Tree_Tup^{gt}, respectively), and treat each triple, node role label, node logical relation as indivisible elements. Thus Tree_LR is defined by

$$\begin{aligned}&\text{Tree\_LR}(\text{Tree\_Tup}^{pred}, \text{Tree\_Tup}^{gt})\\ =&\frac{\text{ED}(\text{Tree\_Tup}^{pred}, \text{Tree\_Tup}^{gt})}{\max(\text{len}(\text{Tree\_Tup}^{pred}), \text{len}(\text{Tree\_Tup}^{gt}))}.\end{aligned} \tag{11}$$

F Appendix for implementation details of the encoder based methods

For pretrained encoder based methods, we use the pre-trained Chinese medical BERT (denoted as MedBERT) by (Guo et al., 2021a) as the default backbone model. For ablation studies, we also consider the widely used BERT-wwm-ext[14], Google BERT-base Chinese (Devlin et al., 2019), and Erlangshen-ZEN1[15]. For the decoding module such as the biaffine module (Dozat and Manning, 2016) and (Wang et al., 2021), we will use the original authors' default configurations. We will fine-tune all the model parameters. Batch size is set to 8, warm-up steps is set to 50, the number of training epochs is set to 50, the learning rate is set to 2e-5 with a linear schedule, and the optimizer is AdamW (Loshchilov and Hutter, 2017). The other hyper-parameters like gradient clipping, Adam epsilon are kept the same with the Transformers repository.

[14] https://huggingface.co/hfl/chinese-bert-wwm-ext.

[15] https://huggingface.co/IDEA-CCNL/Erlangshen-ZEN1-224M-Chinese.

G Dataset details for model training

The original Text2MDT has a 1200:150:150 train/dev/test split. Since we are experimenting with different methods from the pipeline and end2end frameworks, we now need to construct different variations of the Text2MDT datasets.

G.1 Datasets for the pipeline framework

Since the pipeline framework has three subtasks, thus, we need to construct a different dataset for each subtask so that we can train an encoder-based model:

- Text2MDT-TE, the Text2MDT triplet extraction dataset, where the input is the medical text, and the target is the list of triplets in the structured format like JSON. This dataset has a 1200:150:150 train/dev/test split.

- Text2MDT-NG, the Text2MDT node grouping dataset, where the input is the medical text and the list of triplets in text sequence concatenated together, and the output is the list of nodes in the structured format like JSON and each node contains a list of triplets and a logical relation label. For the Text2MDT-NG training set, we augment the original Text2MDT four times by shuffling the orders of triplets. Thus, this dataset has a 4800:150:150 train/dev/test split.

- Text2MDT-TA, the Text2MDT tree assembling dataset, where the input is the medical text and the list nodes in text sequence concatenated together, and the output is the list of MDT nodes in the structured format like JSON and each node contains a list of triplets, a logical relation label and a role label. For the Text2MDT-TA training set, we augment the original Text2MDT four times by shuffling the orders of nodes in the input. Thus, this dataset has a 4800:150:150 train/dev/test split.

For each of the above datasets, we will construct a prompt-based dataset for the generative LM methods, with the prompt and response templates in the the Appendix.

G.2 Datasets for the end2end framework

For each end2end method, we will construct the end2end dataset with the prompt and response templates in the the Appendix. So that each end2end dataset has a 1200:150:150 train/dev/test split.

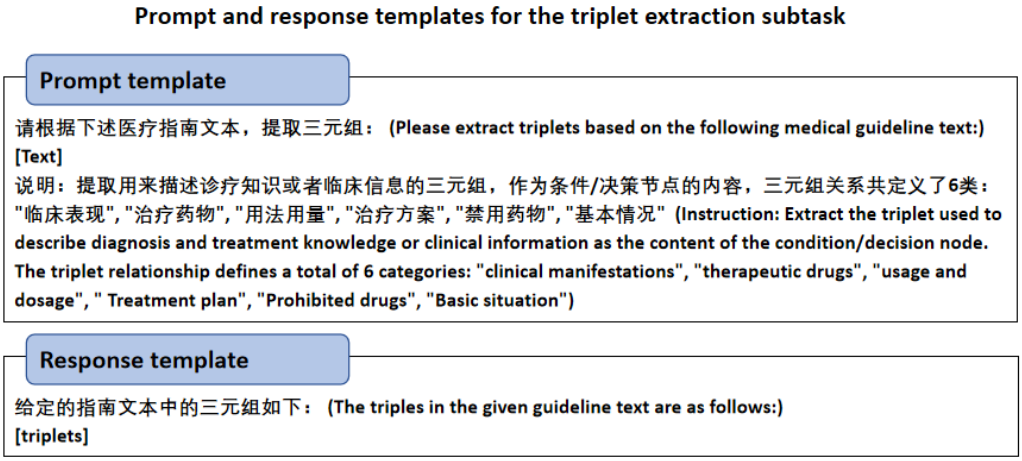

Figure 3: Prompt and response templates for the triplet extraction subtask.

H Prompt templates and response formats for the pipeline framework

H.1 The triplet extraction subtask

In the triplet extraction task asks a language model to predict a series of triplets from the given text. A triplet includes the head entity mention, tail entity mention, and the relation between them. We present the prompt and response template in Figure 3, in which the special token [Text] denotes the input text, and [triplets] denotes a list of triplets. An example pair of prompt and target response is also presented in Figure 4.

With the idea of COT (Wei et al., 2022), the prompt will ask the LLMs to first identify the relations in the given text, and then generate the triplets one by one. We present the COT prompt and response template in Figure 5, in which the COT templates below, [relations] denotes the list of relation names. An example pair of COT prompt and target response is also presented in Figure 6.

H.2 The node grouping subtask

In the node grouping task, we asks a language model to predict which triplets form a node, and which logical relation the node has. Figure 7 presents the prompt and response templates, in which the special token [Text] denotes the input text, and [triplets] denotes the list of extracted triplets, and [node] denotes the contents of the node. An example pair of prompt and target response is also presented in Figure 8.

H.3 The tree assembling subtask

In the tree assembling task, given the results of the node grouping step, we ask the language model to generate the whole decision tree. Figure 9 is the prompt and response templates, in which the special token [Text] denotes the input text, and [nodes] denotes the list of nodes from the previous subtask. In the response, [node_idx] denotes the index of a node, [triplets] denotes the list of extracted triplets in a node, [logical_rel] denotes the logical relation of the node, and [role] denotes the role label of the node. An example pair of prompt and target response is presented in Figure 8.

I Prompt templates and response formats for the end2end framework

I.1 The templates for the Gen method

For the Generation method in the end2end framework, we ask the language model to generate the whole decision tree given the medical guideline text. Figure 11 is the prompt and response templates, in which the special token [Text] denotes the input text. In the response, [node_idx] denotes the index of a node, [triplets] denotes the list of extracted triplets in a node, [logical_rel] denotes the logical relation of the node, and [role] denotes the role label of the node.

I.2 The templates for the COT-Gen-1 method

For the COT-Generation-1 method in the end2end framework, we ask the language model to generate the whole decision tree given the medical guideline text with the following steps: (a) generating the triplets. (b) generating the node grouping results. (c) complete the tree assembling subtask and generate the whole medical decision tree. Figure 12 is the prompt and response templates, in which the special token [Text] denotes the input

Example for the triplet extraction subtask

Prompt example

请根据下述医疗指南文本，提取三元组： (Please extract triplets based on the following medical guideline text:)
肥厚型心肌病患者@对于伴有左室流出道梗阻的患者，可采用药物治疗、植入ICD、化学消融以及手术治疗等方法以改善症状。对于无左室流出道梗阻的患者，治疗重点在于控制心律失常、改善左室充盈压力、缓解心绞痛和抑制疾病进展。 (Patients with hypertrophic cardiomyopathy@For patients with left ventricular outflow tract obstruction, drug treatment, ICD implantation, chemical ablation, and surgical treatment can be used to improve symptoms. For patients without left ventricular outflow tract obstruction, treatment focuses on controlling arrhythmias, improving left ventricular filling pressure, relieving angina, and inhibiting disease progression.)
说明：提取用来描述诊疗知识或者临床信息的三元组，作为条件/决策节点的内容，三元组关系共定义了6类："临床表现", "治疗药物", "用法用量", "治疗方案", "禁用药物", "基本情况" (Instruction: Extract the triplet used to describe diagnosis and treatment knowledge or clinical information as the content of the condition/decision node. The triplet relationship defines a total of 6 categories: "clinical manifestations", "therapeutic drugs", "usage and dosage", " Treatment plan", "Prohibited drugs", "Basic situation")

Response example

给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
["肥厚型心肌病患者", "临床表现", "左室流出道梗阻"], ["肥厚型心肌病患者", "治疗方案", "药物治疗"], ["肥厚型心肌病患者", "治疗方案", "植入ICD"], ["肥厚型心肌病患者", "治疗方案", "化学消融"], ["肥厚型心肌病患者", "治疗方案", "手术治疗"], ["肥厚型心肌病患者", "治疗方案", "控制心律失常"], ["肥厚型心肌病患者", "治疗方案", "改善左室充盈压力"], ["肥厚型心肌病患者", "治疗方案", "缓解心绞痛"], ["肥厚型心肌病患者", "治疗方案", "抑制疾病进展"] (["Patients with hypertrophic cardiomyopathy", "Clinical manifestations", "Left ventricular outflow tract obstruction"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Drug therapy"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "ICD implantation"], ["Patient with hypertrophic cardiomyopathy", "Treatment plan", "Chemical ablation"], ["Patient with hypertrophic cardiomyopathy", "Treatment plan", "Surgery Treatment"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Control arrhythmia"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Improve left ventricular filling pressure"], [" Patients with hypertrophic cardiomyopathy", "Treatment options", "Relieving angina"], ["Patients with hypertrophic cardiomyopathy", "Treatment options", "Inhibiting disease progression"])

Figure 4: An example of the prompt and response for the triplet extraction subtask.

COT Prompt and response templates for the triplet extraction subtask

Prompt template

请根据下述医疗指南文本，提取三元组： (Please extract triplets based on the following medical guideline text:)
[Text]
说明：提取用来描述诊疗知识或者临床信息的三元组，作为条件/决策节点的内容，三元组关系共定义了6类："临床表现", "治疗药物", "用法用量", "治疗方案", "禁用药物", "基本情况" (Instruction: Extract the triplet used to describe diagnosis and treatment knowledge or clinical information as the content of the condition/decision node. The triplet relationship defines a total of 6 categories: "clinical manifestations", "therapeutic drugs", "usage and dosage", " Treatment plan", "Prohibited drugs", "Basic situation")
指令：请先识别出文本中包含哪些关系，然后再提取三元组。 (Instruction: Please identify which relationships are contained in the text before extracting triples.)

Response template

给定的指南文本中的三元组关系有： (The triple relationships in the given guidance text are:)
[relations]

给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
[triplets]

Figure 5: COT style prompt and response templates for the triplet extraction subtask.

Example for the triplet extraction subtask

Prompt example

请根据下述医疗指南文本，提取三元组： (Please extract triplets based on the following medical guideline text:)
肥厚型心肌病患者@对于伴有左室流出道梗阻的患者，可采用药物治疗、植入ICD、化学消融以及手术治疗等方法以改善症状。对于无左室流出道梗阻的患者，治疗重点在于控制心律失常、改善左室充盈压力、缓解心绞痛和抑制疾病进展。 (Patients with hypertrophic cardiomyopathy@For patients with left ventricular outflow tract obstruction, drug treatment, ICD implantation, chemical ablation, and surgical treatment can be used to improve symptoms. For patients without left ventricular outflow tract obstruction, treatment focuses on controlling arrhythmias, improving left ventricular filling pressure, relieving angina, and inhibiting disease progression.)
说明：提取用来描述诊疗知识或者临床信息的三元组，作为条件/决策节点的内容，三元组关系共定义了6类："临床表现","治疗药物","用法用量","治疗方案","禁用药物","基本情况" (Instruction: Extract the triplet used to describe diagnosis and treatment knowledge or clinical information as the content of the condition/decision node. The triplet relationship defines a total of 6 categories: "clinical manifestations", "therapeutic drugs", "usage and dosage", " Treatment plan", "Prohibited drugs", "Basic situation")
指令：请先识别出文本中包含哪些关系，然后再提取三元组。 (Instruction: Please identify which relationships are contained in the text before extracting triples.)

Response example

给定的指南文本中的三元组关系有： (The triple relationships in the given guidance text are:)
临床表现，治疗方案 (Clinical manifestations, treatment options)
给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
["肥厚型心肌病患者","临床表现","左室流出道梗阻"],["肥厚型心肌病患者","治疗方案","药物治疗"],["肥厚型心肌病患者","治疗方案","植入ICD"],["肥厚型心肌病患者","治疗方案","化学消融"],["肥厚型心肌病患者","治疗方案","手术治疗"],["肥厚型心肌病患者","治疗方案","控制心律失常"],["肥厚型心肌病患者","治疗方案","改善左室充盈压力"],["肥厚型心肌病患者","治疗方案","缓解心绞痛"],["肥厚型心肌病患者","治疗方案","抑制疾病进展"] (["Patients with hypertrophic cardiomyopathy", "Clinical manifestations", "Left ventricular outflow tract obstruction"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Drug therapy"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "ICD implantation"], ["Patient with hypertrophic cardiomyopathy", "Treatment plan", "Chemical ablation"], ["Patient with hypertrophic cardiomyopathy", "Treatment plan", "Surgery Treatment"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Control arrhythmia"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Improve left ventricular filling pressure"], [" Patients with hypertrophic cardiomyopathy", "Treatment options", "Relieving angina"], ["Patients with hypertrophic cardiomyopathy", "Treatment options", "Inhibiting disease progression"])

Figure 6: An example of the prompt and response with COT for the triplet extraction subtask.

Prompt and response templates for the node grouping subtask

Prompt template

请根据下述医疗指南文本，以及从其中抽取的三元组信息，将这些三元组组合成若干个节点，并指出这个节点内的三元组的逻辑关系： (Please combine these triples into several nodes based on the following medical guideline text and the triplet information extracted from it, and indicate the logical relationship of the triplets within this node:)
医疗指南文本： (Medical guideline text:)
[Text]
给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
[triplets]
说明：如果若干个三元组组成一个节点，则说明这些三元组两两之间具有and或者or的逻辑关系。如果一个三元组与其他三元组没有and或者or关系，则说明这个三元组需要独立成为一个节点 (Note: If several triples form a node, it means that there is an and or or logical relationship between these triples. If a triple does not have an and or or relationship with other triples, it means that the triple needs to become a node independently.)

Response template

根据给定的指南文本及其三元组信息，决策树的节点组成如下： (Based on the given guideline text and its triplet information, the nodes of the decision tree are composed as follows:)
如下的三元组构成决策树的一个节点：[triplets]。这个节点的逻辑关系为：[logical_rel] (The following triples constitute a node of the decision tree: [triplets]. The logical relationship of this node is: [logical_rel])
如下的三元组构成决策树的一个节点：[triplets]。这个节点的逻辑关系为：[logical_rel] (The following triples constitute a node of the decision tree: [triplets]. The logical relationship of this node is: [logical_rel])

Figure 7: Prompt and response templates for the node grouping subtask.

Example for the triplet extraction subtask

Prompt example

请根据下述医疗指南文本，以及从其中抽取的三元组信息，将这些三元组组合成若干个节点，并指出这个节点内的三元组的逻辑关系： (Please combine these triples into several nodes based on the following medical guideline text and the triplet information extracted from it, and indicate the logical relationship of the triplets within this node:)
医疗指南文本： (Medical guideline text:)
肥厚型心肌病患者@对于伴有左室流出道梗阻的患者，可采用药物治疗、植入ICD、化学消融以及手术治疗等方法以改善症状。对于无左室流出道梗阻的患者，治疗重点在于控制心律失常、改善左室充盈压力、缓解心绞痛和抑制疾病进展。 (Patients with hypertrophic ardiomyopathy@For patients with left ventricular outflow tract obstruction, drug treatment, ICD implantation, chemical ablation, and surgical treatment can be used to improve symptoms. For patients without left ventricular outflow tract obstruction, treatment focuses on controlling arrhythmias, improving left ventricular filling pressure, relieving angina, and inhibiting disease progression.)
给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
["肥厚型心肌病患者", "临床表现", "左室流出道梗阻"], ["肥厚型心肌病患者", "治疗方案", "药物治疗"], ["肥厚型心肌病患者", "治疗方案", "植入ICD"], ["肥厚型心肌病患者", "治疗方案", "化学消融"], ["肥厚型心肌病患者", "治疗方案", "手术治疗"], ["肥厚型心肌病患者", "治疗方案", "控制心律失常"], ["肥厚型心肌病患者", "治疗方案", "改善左室充盈压力"], ["肥厚型心肌病患者", "治疗方案", "缓解心绞痛"], ["肥厚型心肌病患者", "治疗方案", "抑制疾病进展"] (["Patients with hypertrophic cardiomyopathy", "Clinical manifestations", "Left ventricular outflow tract obstruction"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Drug therapy"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "ICD implantation"], ["Patient with hypertrophic cardiomyopathy", "Treatment plan", "Chemical ablation"], ["Patient with hypertrophic cardiomyopathy", "Treatment plan", "Surgery Treatment"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Control arrhythmia"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Improve left ventricular filling pressure"], [" Patients with hypertrophic cardiomyopathy", "Treatment options", "Relieving angina"], ["Patients with hypertrophic cardiomyopathy", "Treatment options", "Inhibiting disease progression"])
说明：如果若干个三元组组成一个节点，则说明这些三元组两两之间具有and或者or的逻辑关系。如果一个三元组与其他三元组没有and或者or关系，则说明这个三元组需要独立成为一个节点 (Note: If several triples form a node, it means that there is an and or or logical relationship between these triples. If a triple does not have an and or or relationship with other triples, it means that the triple needs to become a node independently.)

Response example

根据给定的指南文本及其三元组信息，决策树的节点组成如下： (Based on the given guideline text and its triplet information, the nodes of the decision tree are composed as follows:)
如下的三元组构成决策树的一个节点：[["肥厚型心肌病患者", "治疗方案", "手术治疗"], ["肥厚型心肌病患者", "治疗方案", "药物治疗"], ["肥厚型心肌病患者", "治疗方案", "化学消融"], ["肥厚型心肌病患者", "治疗方案", "植入ICD"]]。这个节点的逻辑关系为：or (The following triplet constitutes a node of the decision tree: [["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Surgical treatment"], ["Patients with hypertrophic cardiomyopathy", Treatment plan", "Medical treatment"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Chemical ablation"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "ICD implantation"]]. The logical relationship of this node is: or)
如下的三元组构成决策树的一个节点：[["肥厚型心肌病患者", "临床表现", "左室流出道梗阻"]]。这个节点的逻辑关系为：null (The following triplet constitutes a node of the decision tree: [["Patients with hypertrophic cardiomyopathy", "Clinical manifestations", "Left ventricular outflow tract obstruction"]]. The logical relationship of this node is: null)
如下的三元组构成决策树的一个节点：[["肥厚型心肌病患者", "治疗方案", "控制心律失常"], ["肥厚型心肌病患者", "治疗方案", "抑制疾病进展"], ["肥厚型心肌病患者", "治疗方案", "改善左室充盈压力"], ["肥厚型心肌病患者", "治疗方案", "缓解心绞痛"]]。这个节点的逻辑关系为：and (The following triplet constitutes a node of the decision tree: [["Hypertrophic cardiomyopathy patient", "Treatment plan", "Arrhythmia control"], ["Hypertrophic cardiomyopathy patient", "Treatment plan", "Inhibit disease progression"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Improve left ventricular filling pressure"], ["Hypertrophic cardiomyopathy Patients with cardiomyopathy", "Treatment options", "Relieving angina pectoris"]]. The logical relationship of this node is: and)

Figure 8: An example of the prompt and response for the node grouping subtask.

Prompt and response templates for the tree assembling subtask

Prompt template

请根据下述医疗指南文本，以及从其中抽取的节点信息，将这些节点组成一个决策树： (Please form a decision tree based on the following medical guideline text and the node information extracted from it:)
医疗指南文本： (Medical guideline text:)
[Text]
给定的指南文本中的节点组成如下： (The nodes in a given guideline text are composed as follows:)
[nodes]
说明：(1)诊疗决策树是由条件节点和决策节点组成的二叉树，旨在通过简洁的结构化信息表达指南文本，既要求将文本中的核心实体和关系挖掘出来，也需要将这些信息进行串联，形成一个完整的决策流程；(2)在诊疗决策二叉树中，非叶子节点是条件节点，叶子节点是决策节点。对于条件节点，当条件判断结果为"是"时，将转到左侧子节点进行下一个判断或决策，当条件判断结果为"否"时，将转到右侧子节点进行下一个判断或决策。(3)每个节点输出为一个dict，包含三个字段：(3a) "role"，即节点角色类型；(3b) "triples"，即三元组列表；(3c) "logical_rel"，表示节点的逻辑关系。(4)整个诊疗决策树以广度优先策略排列为一个列表。
(Note: (1) The diagnosis and treatment decision tree is a binary tree composed of conditional nodes and decision nodes. It aims to express guideline text through concise structured information. It requires not only to dig out the core entities and relationships in the text, but also to carry out this information. They are connected in series to form a complete decision-making process; (2) In the diagnosis and treatment decision-making binary tree, non-leaf nodes are condition nodes and leaf nodes are decision nodes. For the condition node, when the condition judgment result is "yes", it will go to the left child node for the next judgment or decision. When the condition judgment result is "no", it will go to the right child node for the next judgment or decision. (3) The output of each node is a dict, containing three fields: (3a) "role", which is the node role type; (3b) "triples", which is a list of triples; (3c) "logical_rel", which represents the node logical relationship. (4) The entire diagnosis and treatment decision tree is arranged into a list using the breadth-first strategy.)

Response template

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

Figure 9: Prompt and response templates for the tree assembling subtask.

text. In the response, [node_idx] denotes the index of a node, [triplets] denotes a list of extracted triplets, [logical_rel] denotes the logical relation of the node, and [role] denotes the role label of the node.

I.3 The templates for the COT-Generation-2 method

For the COT-Generation-2 method in the end2end framework, we ask the language model to generate the whole decision tree given the medical guideline text with the following steps: (a) generating the entities; (b) extract the triplets; (c) grouping the triplets into nodes; (d) determining the role labels of the nodes; (e) and finally assembling the whole medical decision tree. Figure 13 is the prompt and response templates, in which the special token [Text] denotes the input text. In the response, [node_idx] denotes the index of a node, [entities] denotes a list of entity mentions, [triplets] denotes a list of extracted triplets, [role_labels] denotes a list of role labels, [logical_rel] denotes the logical relation of the node, and [role] denotes the role label of the node.

I.4 The templates for the COT-Generation-3 method

For the COT-Generation-3 method in the end2end framework, we ask the language model to generate the whole decision tree given the medical guideline text with the following steps: (a) generating the triplets, and then (b) generate the whole medical decision tree. Figure 14 is the prompt and response templates, in which the special token [Text] denotes the input text. In the response, [node_idx] denotes the index of a node, [triplets] denotes a list of extracted triplets, [logical_rel] denotes the logical relation of the node, and [role] denotes the role label of the node.

I.5 The templates for the COT-Generation-4 method

For the COT-Generation-4 method in the end2end framework, we ask the language model to generate the whole decision tree given the medical guideline text with the following steps: (a) generating the entity mentions, (b) generate the triplets, and then (c) generate the whole medical decision tree. Figure 15 is the prompt and response templates, in which the special token [Text] de-

Example for the tree assembling subtask

Prompt example

请根据下述医疗指南文本，以及从其中抽取的节点信息，将这些节点组成一个决策树： (Please form a decision tree based on the following medical guideline text and the node information extracted from it:)

医疗指南文本： (Medical guideline text:)

肥厚型心肌病患者@对于伴有左室流出道梗阻的患者，可采用药物治疗、植入ICD、化学消融以及手术治疗等方法以改善症状。对于无左室流出道梗阻的患者，治疗重点在于控制心律失常、改善左室充盈压力、缓解心绞痛和抑制疾病进展。 (Patients with hypertrophic cardiomyopathy@For patients with left ventricular outflow tract obstruction, drug treatment, ICD implantation, chemical ablation, and surgical treatment can be used to improve symptoms. For patients without left ventricular outflow tract obstruction, treatment focuses on controlling arrhythmias, improving left ventricular filling pressure, relieving angina, and inhibiting disease progression.)

给定的指南文本中的节点组成如下： (The nodes in a given guideline text are composed as follows:)

如下的三元组构成决策树的一个节点：[["肥厚型心肌病患者", "治疗方案", "手术治疗"], ["肥厚型心肌病患者", "治疗方案", "药物治疗"], ["肥厚型心肌病患者", "治疗方案", "化学消融"], ["肥厚型心肌病患者", "治疗方案", "植入ICD"]]。这个节点的逻辑关系为：or (The following triplet constitutes a node of the decision tree: [["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Surgical treatment"], ["Patients with hypertrophic cardiomyopathy", Treatment plan", "Medical treatment"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Chemical ablation"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "ICD implantation"]]. The logical relationship of this node is: or)

如下的三元组构成决策树的一个节点：[["肥厚型心肌病患者", "临床表现", "左室流出道梗阻"]]。这个节点的逻辑关系为：null (The following triplet constitutes a node of the decision tree: [["Patients with hypertrophic cardiomyopathy", "Clinical manifestations", "Left ventricular outflow tract obstruction"]]. The logical relationship of this node is: null)

如下的三元组构成决策树的一个节点：[["肥厚型心肌病患者", "治疗方案", "控制心律失常"], ["肥厚型心肌病患者", "治疗方案", "抑制疾病进展"], ["肥厚型心肌病患者", "治疗方案", "改善左室充盈压力"], ["肥厚型心肌病患者", "治疗方案", "缓解心绞痛"]]。这个节点的逻辑关系为：and (The following triplet constitutes a node of the decision tree: [["Hypertrophic cardiomyopathy patient", "Treatment plan", "Arrhythmia control"], ["Hypertrophic cardiomyopathy patient", "Treatment plan", "Inhibit disease progression"], ["Patients with hypertrophic cardiomyopathy", "Treatment plan", "Improve left ventricular filling pressure"], ["Hypertrophic cardiomyopathy Patients with cardiomyopathy", "Treatment options", "Relieving angina pectoris"]]. The logical relationship of this node is: and)

说明：(1)诊疗决策树是由条件节点和决策节点组成的二叉树，旨在通过简洁的结构化信息表达指南文本，既要求将文本中的核心实体和关系挖掘出来，也需要将这些信息进行串联，形成一个完整的决策流程；(2)在诊疗决策二叉树中，非叶子节点是条件节点，叶子节点是决策节点。对于条件节点，当条件判断结果为“是”时，将转到左侧子节点进行下一个判断或决策，当条件判断结果为“否”时，将转到右侧子节点进行下一个判断或决策。(3)每个节点输出为一个dict，包含三个字段：(3a) "role"，即节点角色类型；(3b) "triples"，即三元组列表；(3c) "logical_rel"，表示节点的逻辑关系。(4)整个诊疗决策树以广度优先策略排列为一个列表。 (Note: (1) The diagnosis and treatment decision tree is a binary tree composed of conditional nodes and decision nodes. It aims to express guideline text through concise structured information. It requires not only to dig out the core entities and relationships in the text, but also to carry out this information. They are connected in series to form a complete decision-making process; (2) In the diagnosis and treatment decision-making binary tree, non-leaf nodes are condition nodes and leaf nodes are decision nodes. For the condition node, when the condition judgment result is "yes", it will go to the left child node for the next judgment or decision. When the condition judgment result is "no", it will go to the right child node for the next judgment or decision. (3) The output of each node is a dict, containing three fields: (3a) "role", which is the node role type; (3b) "triples", which is a list of triples; (3c) "logical_rel", which represents the node logical relationship. (4) The entire diagnosis and treatment decision tree is arranged into a list using the breadth-first strategy.)

Response example

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)

节点0：role=C；logical_rel=null；triples=[["肥厚型心肌病患者", "临床表现", "左室流出道梗阻"]]

节点1：role=D；logical_rel=or；triples=[["肥厚型心肌病患者", "治疗方案", "药物治疗"], ["肥厚型心肌病患者", "治疗方案", "植入ICD"], ["肥厚型心肌病患者", "治疗方案", "化学消融"], ["肥厚型心肌病患者", "治疗方案", "手术治疗"]]

节点2：role=D；logical_rel=and；triples=[["肥厚型心肌病患者", "治疗方案", "控制心律失常"], ["肥厚型心肌病患者", "治疗方案", "改善左室充盈压力"], ["肥厚型心肌病患者", "治疗方案", "缓解心绞痛"], ["肥厚型心肌病患者", "治疗方案", "抑制疾病进展"]]

Figure 10: An example of the prompt and response for the tree assembling subtask.

Prompt and response templates for the Gen method

Prompt template

请对下述医学指南文本生成诊疗决策树： (Please generate a diagnosis and treatment decision tree for the following medical guideline text:)
[Text]
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。
(Task description: (1) Based on the given medical guideline text, create a binary tree, including conditional nodes and decision nodes, to succinctly display the guideline content while capturing core entities and relationships; (2) Conditional nodes are used for judgment, based on the results Point to the left or right child node to make the next decision. (3) The output of each node is a dict, containing three fields: (3a) "role" indicates the node type, which can be a condition node ("C") or a decision node ("D"); (3b) "triples" is a list of triples describing diagnosis and treatment knowledge or clinical information, including "clinical manifestations", "therapeutic drugs", "usage and dosage", "treatment plan", "Prohibited drugs", "Basic situation" six types of relationships; (3c) "logical_rel" represents the logical relationship between multiple triples (the values are and, or, null, when there is only one triple. The logical relationship is null when it is a tuple). (4) The finally generated diagnosis and treatment decision tree is arranged into a list according to the breadth-first strategy.)

Response template

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

Figure 11: Prompt and response templates for the Gen method.

notes the input text. In the response, [node_idx] denotes the index of a node, [triplets] denotes a list of extracted triplets, [logical_rel] denotes the logical relation of the node, and [role] denotes the role label of the node.

J Appendix for case studies

We report two case studies in Figure 16 and 17, analyzing .

Prompt and response templates for the COT-Gen-1 method

Prompt template

请对下述医学指南文本生成诊疗决策树： (Please generate a diagnosis and treatment decision tree for the following medical guideline text:)
[Text]
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。
(Task description: (1) Based on the given medical guideline text, create a binary tree, including conditional nodes and decision nodes, to succinctly display the guideline content while capturing core entities and relationships; (2) Conditional nodes are used for judgment, based on the results Point to the left or right child node to make the next decision. (3) The output of each node is a dict, containing three fields: (3a) "role" indicates the node type, which can be a condition node ("C") or a decision node ("D"); (3b) "triples" is a list of triples describing diagnosis and treatment knowledge or clinical information, including "clinical manifestations", "therapeutic drugs", "usage and dosage", "treatment plan", "Prohibited drugs", "Basic situation" six types of relationships; (3c) "logical_rel" represents the logical relationship between multiple triples (the values are and, or, null, when there is only one triple. The logical relationship is null when it is a tuple). (4) The finally generated diagnosis and treatment decision tree is arranged into a list according to the breadth-first strategy.)
生成步骤说明: 请一步步的完成决策树的生成，(a) 先从上述文本中抽取三元组；(b) 根据三元组抽取结果，将三元组分配到不同的节点中；(c) 最后生成完整的决策树。 (Instructions for the generation steps: Please complete the generation of the decision tree step by step. (a) First extract triples from the above text; (b) According to the triple extraction results, allocate the triples to different nodes; (c) and finally generate a complete decision tree.)

Response template

给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
[triplets]

根据给定的指南文本及其三元组信息，决策树的节点组成如下： (Based on the given guideline text and its triplet information, the nodes of the decision tree are composed as follows:)
如下的三元组构成决策树的一个节点：[triplets]。这个节点的逻辑关系为：[logical_rel] (The following triples constitute a node of the decision tree: [triplets]. The logical relationship of this node is: [logical_rel])
如下的三元组构成决策树的一个节点：[triplets]。这个节点的逻辑关系为：[logical_rel] (The following triples constitute a node of the decision tree: [triplets]. The logical relationship of this node is: [logical_rel])

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

Figure 12: Prompt and response templates for the COT-Gen-1 method.

Prompt and response templates for the COT-Gen-2 method

Prompt template

请对下述医学指南文本生成诊疗决策树： (Please generate a diagnosis and treatment decision tree for the following medical guideline text:)

[Text]

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。
(Task description: (1) Based on the given medical guideline text, create a binary tree, including conditional nodes and decision nodes, to succinctly display the guideline content while capturing core entities and relationships; (2) Conditional nodes are used for judgment, based on the results Point to the left or right child node to make the next decision. (3) The output of each node is a dict, containing three fields: (3a) "role" indicates the node type, which can be a condition node ("C") or a decision node ("D"); (3b) "triples" is a list of triples describing diagnosis and treatment knowledge or clinical information, including "clinical manifestations", "therapeutic drugs", "usage and dosage", "treatment plan", "Prohibited drugs", "Basic situation" six types of relationships; (3c) "logical_rel" represents the logical relationship between multiple triples (the values are and, or, null, when there is only one triple. The logical relationship is null when it is a tuple). (4) The finally generated diagnosis and treatment decision tree is arranged into a list according to the breadth-first strategy.)

生成步骤说明: 请一步步的完成决策树的生成，(a) 先从上述文本中抽取实体；(b) 再抽取三元组；(c) 根据三元组抽取结果，将三元组分配到不同的节点中；(d) 决定节点的角色标签； (e) 最后生成完整的决策树。 (Instructions for the generation steps: Please complete the generation of the decision tree step by step. (a) First extract entities from the above text; (b) Then extract triples; (c) According to the triple extraction results, assign triples to in different nodes; (d) determine the role label of the node; (e) finally generate a complete decision tree.)

Response template

给定的指南文本中的医学实体如下：

[entities]

给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)

[triplets]

根据给定的指南文本及其三元组信息，决策树的节点组成如下： (Based on the given guideline text and its triplet information, the nodes of the decision tree are composed as follows:)

如下的三元组构成决策树的一个节点：[triplets]。**这个节点的逻辑关系为：**[logical_rel] (The following triples constitute a node of the decision tree: [triplets]. The logical relationship of this node is: [logical_rel])

如下的三元组构成决策树的一个节点：[triplets]。**这个节点的逻辑关系为：**[logical_rel] (The following triples constitute a node of the decision tree: [triplets]. The logical relationship of this node is: [logical_rel])

上述各个节点的角色标签为: [role_labels]

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)

节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

Figure 13: Prompt and response templates for the COT-Gen-2 method.

Prompt and response templates for the COT-Gen-3 method

Prompt template

请对下述医学指南文本生成诊疗决策树： (Please generate a diagnosis and treatment decision tree for the following medical guideline text:)

[Text]

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。
(Task description: (1) Based on the given medical guideline text, create a binary tree, including conditional nodes and decision nodes, to succinctly display the guideline content while capturing core entities and relationships; (2) Conditional nodes are used for judgment, based on the results Point to the left or right child node to make the next decision. (3) The output of each node is a dict, containing three fields: (3a) "role" indicates the node type, which can be a condition node ("C") or a decision node ("D"); (3b) "triples" is a list of triples describing diagnosis and treatment knowledge or clinical information, including "clinical manifestations", "therapeutic drugs", "usage and dosage", "treatment plan", "Prohibited drugs", "Basic situation" six types of relationships; (3c) "logical_rel" represents the logical relationship between multiple triples (the values are and, or, null, when there is only one triple. The logical relationship is null when it is a tuple). (4) The finally generated diagnosis and treatment decision tree is arranged into a list according to the breadth-first strategy.)

生成步骤说明: 请一步步的完成决策树的生成，(a) 先从上述文本中抽取三元组；(b) 然后生成完整的决策树。 (Instructions for the generation steps: Please complete the generation of the decision tree step by step. (a) First extract triples from the above text; (b) and then generate a complete decision tree.)

Response template

给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)

[triplets]

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)

节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

Figure 14: Prompt and response templates for the COT-Gen-3 method.

Prompt and response templates for the COT-Gen-4 method

Prompt template

请对下述医学指南文本生成诊疗决策树： (Please generate a diagnosis and treatment decision tree for the following medical guideline text:)
[Text]
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。
(Task description: (1) Based on the given medical guideline text, create a binary tree, including conditional nodes and decision nodes, to succinctly display the guideline content while capturing core entities and relationships; (2) Conditional nodes are used for judgment, based on the results Point to the left or right child node to make the next decision. (3) The output of each node is a dict, containing three fields: (3a) "role" indicates the node type, which can be a condition node ("C") or a decision node ("D"); (3b) "triples" is a list of triples describing diagnosis and treatment knowledge or clinical information, including "clinical manifestations", "therapeutic drugs", "usage and dosage", "treatment plan", "Prohibited drugs", "Basic situation" six types of relationships; (3c) "logical_rel" represents the logical relationship between multiple triples (the values are and, or, null, when there is only one triple. The logical relationship is null when it is a tuple). (4) The finally generated diagnosis and treatment decision tree is arranged into a list according to the breadth-first strategy.)
生成步骤说明: 请一步步的完成决策树的生成，(a) 先从上述文本中抽取医学实体；(b) 提取医学三元组；(c) 然后生成完整的决策树。 (Generation step instructions: Please complete the decision tree generation step by step, (a) first extract medical entities from the above text; (b) extract medical triples; (c) then generate a complete decision tree.)

Response template

给定的指南文本中的医学实体如下： (The medical entities in the given guidance text are as follows:)
[entities]

给定的指南文本中的三元组如下： (The triples in the given guideline text are as follows:)
[triplets]

根据给定的指南文本抽取的诊疗决策树如下： (The diagnosis and treatment decision tree extracted based on the given guideline text is as follows:)
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])
节点[node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets] (node [node_idx]：role=[role]；logical_rel=[logical_rel]；triplets=[triplets])

Figure 15: Prompt and response templates for the COT-Gen-4 method.

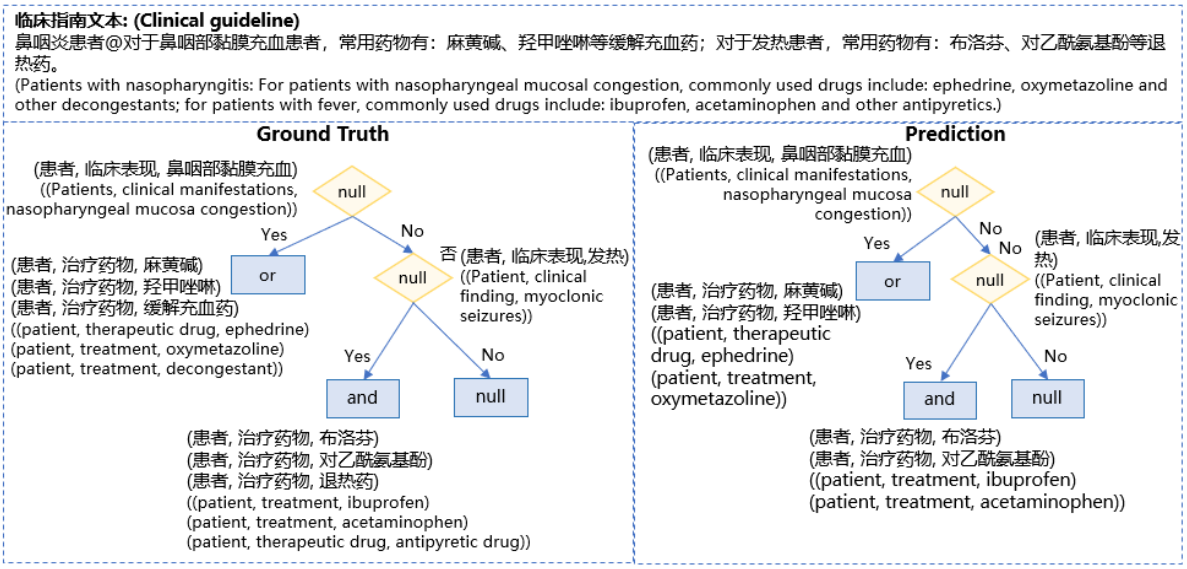

Figure 16: Example (a), an error case of the COT-Generation-3 method on the Text2MDT test samples.

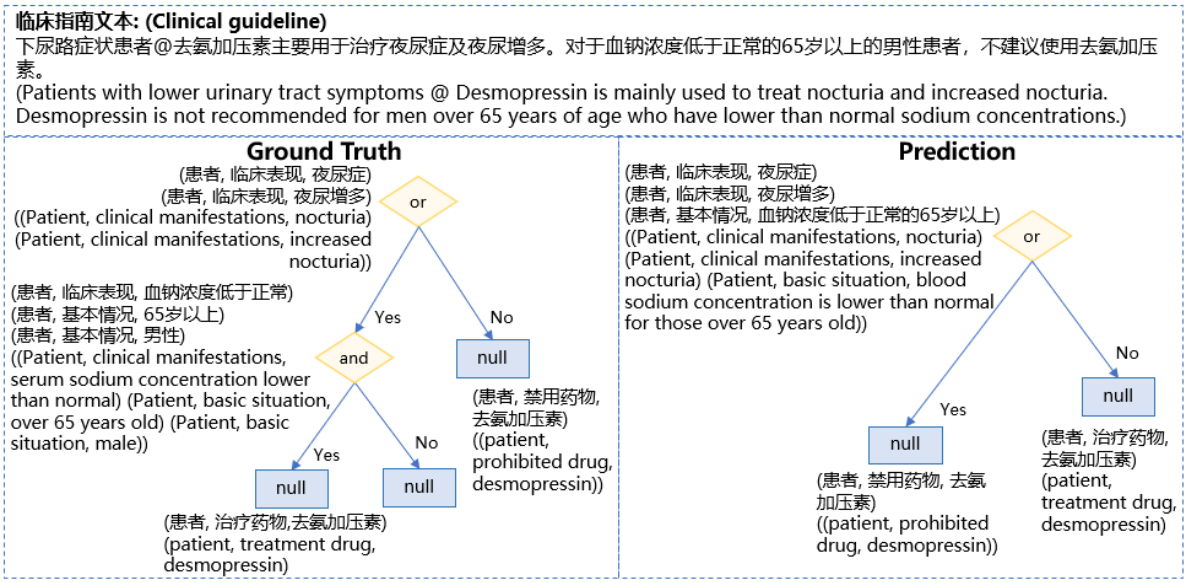

Figure 17: Example (b), an error case of the COT-Generation-3 method on the Text2MDT test samples.