2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 979-8-3503-3748-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/BIBM58861.2023.10385557

Integrating Automated Knowledge Extraction with Large Language Models for Explainable Medical Decision-Making

Haodi Zhang*, Jiahong Li*, Yichi Wang*, Yuanfeng Song†‡
*College of Computer and Software Engineering, Shenzhen University, Shenzhen, China
†HKUST, Hong Kong, China ‡AI Group, WeBank Co., Ltd, Shenzhen, China

***Abstract*—Large language models (LLMs) have demonstrated strong reasoning ability and inspired many previously unimaginable applications. In this paper, we aim to harness the strong reasoning capability of LLMs toward explainable medical diagnosis. As we know, deep learning has been widely adopted and shown improvement in medical diagnostics. However, it is often criticized for its lack of interpretability. To address this drawback, we propose the first method that innovatively combines Markov logic networks (MLNs) with external knowledge extracted using LLMs, aiming for improved both interpretability and accuracy. Specifically, our approach involves a new process, powered by LLMs and a search engine, to automatically collect and organize external medical knowledge. The outcome is a set of first-order logic (FOL) rules, which then become a key component for the following MLN-based diagnostic algorithm. The resulting MLN-based model can maintain the accuracy of deep networks while providing understandable reasoning for its decisions. By aiming to blend specific knowledge from the medical domain with LLM techniques, our work contributes towards the development of improved automatic diagnosis systems, with the potential for enhancing transparency and trust in medical diagnostics.**

***Index Terms*—Explainable Automatic Diagnosis, Large Language Models, Knowledge Extraction, Medical Reasoning**

I. Introduction

Automatic diagnosis has become more and more attractive since it can play a critical role in the clinical decision support system [1]. These systems aid clinicians in making more effective diagnostic decisions by allowing for an in-depth probe into symptoms and the drawing of conclusions from patient-agent interactions. The interactive inquiry process enables the agent to obtain more detailed information and form a more accurate diagnosis [2].

Recent studies of utilizing deep neural networks to address the above-mentioned task have shown to be quite promising. The prevailing approach usually treats the task as a sequential decision-making process, formulating it as a Markov decision process (MDP) and utilizing reinforcement learning (RL) for policy learning [3]–[7]. Furthermore, methods like Diaformer [2] design some sequence generation methods to tackle the imbalance between symptom inquiry and disease diagnosis inherent in RL methods.

Nevertheless, these models, acting as "black boxes", intrinsically lack transparency due to their dependency on learning through a complex neural network of hyperparameters. As a result, the decision-making process of these models remains uninterpretable, a disadvantage given the necessity for transparency and evidence-based reasoning in medical decision-making.

As we know, generative large language models (LLMs, e.g., ChatGPT) [8] have revolted the computer science area and expressed strong reasoning capabilities like Chain-of-Thought [9] ability. Despite their wide applications in common fields like dialogue systems [10], how to efficiently utilize their capability in medical diagnosis is still an open question. In this paper, we unprecedentedly combine LLMs with Markov logic networks (MLNs) [11]. MLNs refer to a class of probabilistic logic models and enjoy the advantages of probability theory and First-order Logic (FOL), offering both uncertainty handling and logical reasoning power. However, MLNs usually require medical knowledge in the form of FOL rules, which are often predefined by medical experts [12]. The process of translating a large amount of medical knowledge into FOL rules can be complex and time-consuming.

To address these challenges, our proposed framework systematically integrates LLMs and MLNs to enjoy the merits of both. To be more specific, our approach uses an LLM for automated knowledge acquisition and formalization, coupled with an MLN for logical medical reasoning. This framework effectively harnesses the robust information processing abilities of the LLM and the logical reasoning capabilities of the MLN, enabling automated knowledge processing while retaining essential interpretability in medical decision-making.

Our framework includes three stages, namely *Knowledge Acquisition*, *Knowledge Formalization*, and *Iterative Optimization*. In the knowledge acquisition stage, we restrict the search scope of the search engine to authoritative medical and pharmaceutical websites to ensure the reliability of the knowledge obtained. To guarantee accuracy, we only select the relevant top-K returned documents via semantic comparison. Finally, a summarization step over these relevant documents is further conducted by prompting the LLM. Knowledge formalization aims to translate the obtained medical knowledge into FOL rules, which includes defining the predicates, selecting the relevant candidate predicates, and prompting the LLM to generate the FOL rules. To further improve the quality of the obtained FOL rules, an iterative optimization stage is also

designed to adaptively prompt the LLM.

After the above-mentioned three stages, our framework effectively translates medical knowledge into FOL rules, and these rules act as interpretable systematic knowledge expressions that are critical for downstream diagnosis. Specifically, we train an MLN diagnosis system with these rules to achieve interpretable reasoning. An evidence database is constructed for each dataset to facilitate MLN weight learning. During the inference stage, the weighted FOL rules inferred by the MLN are comprehensible and can be translated back into natural language, resulting in interpretable reasoning. We conduct extensive experiments on both real-world and synthetic datasets. Our final results demonstrate that our proposed framework offers good interpretability as well as superior performance, outperforming various strong baseline models.

The contributions of this paper are summarized as follows:

- Our framework is the first to combine LLMs with MLNs to achieve interpretable reasoning in medical diagnosis. This framework achieves systematic integration of neural and symbolic methods, allowing for comprehensive patient information extraction through neural methods and diagnosis through interpretable symbolic reasoning.
- Our framework includes an innovative framework for automatic knowledge acquisition and formalization using LLMs. Specifically, the designed iterative optimization strategy can significantly refine the acquired FOL rules. The refined FOL rules by this approach are validated to be effective in enhancing the accuracy and interpretability of medical diagnosis.
- We conduct extensive evaluations on both English and Chinese datasets. The results demonstrate the effectiveness of our proposed model in providing good interpretability as well as superior performance, outperforming various strong baseline models.

The rest of the paper is organized as follows. Section II reviews the related work. Section III present the details of the designed methods. Section IV shows experimental results, followed by conclusion in Section V.

II. Related work

A. Automated Medical Diagnosis

The field of automated medical diagnosis has been a hotbed of research for many years. Earlier methods were primarily centered around reinforcement learning (RL) techniques [3]–[7]. These RL approaches learn how to make inquiries and diagnoses based on the rewards returned from their actions. More recently, Transformer-based [13] models have gained popularity in the clinical domain. For instance, Diaformer [2] reframes the task as a sequence generation process, significantly outperforming previous RL models and enhancing training efficiency. Neuralsympcheck [14] employs one model to suggest inquiry symptoms and another to perform the actual diagnosis.

Furthermore, rule-based expert systems have also been employed for medical diagnosis [12]. The performance of these systems largely depends on the quality of the rules and the medical knowledge bases they build upon. In contrast to these conventional rule-based systems, our approach is the first to take advantage of the impressive power of LLMs. We leverage LLMs to formulate medical rules and use Diaformer for symptom inquiry while relying on the MLN for final diagnosis. This systematic integration facilitates a more robust and interpretable medical diagnosis process.

B. Large Language Models

LLMs have demonstrated promising potential in information extraction tasks [15]–[17]. Even in the specialized domain of clinical text, models like InstructGPT have performed impressively in zero- and few-shot information extraction tasks, without domain-specific training [16]. In light of these capabilities, various methodologies have been proposed to further enhance the data processing and logical reasoning capabilities of LLMs. Chain of Thought (CoT) [9] and its related method like self-refine [18]–[20] have substantially extended the boundaries of what LLMs can achieve [9], [21]. The study by Wei et al. [9] has shown that providing LLMs with several guiding examples before posing a question can greatly improve the performance of logical reasoning tasks.

Additionally, retrieval-enhanced methods, such as those proposed in [22], [23], combine the strengths of CoT prompts with the retrieval of relevant knowledge. Rethinking with Retrieval [22], for instance, generates inference paths consisting of explanation-prediction pairs guided by CoT cues. Subsequently, it retrieves knowledge to bolster the explanation and selects the prediction supported by the most evidence. This approach not only improves the model's performance but also its interpretability. Our study focuses on harnessing the superior reasoning capability of LLMs towards interpretable medical diagnosis.

C. Markov Logic Networks

MLNs [11] combines the power of probabilistic graphical models with first-order logic, thereby integrating the strengths of logical and statistical artificial intelligence [24]. An MLN is composed of weighted first-order logic formulas that define templates for constructing Markov networks. This enables the MLN to produce a probabilistic distribution over possible worlds. The weight of each formula is indicative of its relative importance. As the weight approaches infinity, Markov logic converges to pure first-order logic. These weights can be either manually assigned or automatically learned from data.

In the context of our work, we leverage an innovative approach where we use LLMs to extract initial formulas from medical knowledge for the MLN. This approach differs significantly from traditional methodologies and adds a unique angle to the application of MLN in medical reasoning tasks.

III. Our Proposed Method

In this section, we present our proposed framework, an efficient and effective approach that automatically gathers and processes domain knowledge and then applies it to an MLN

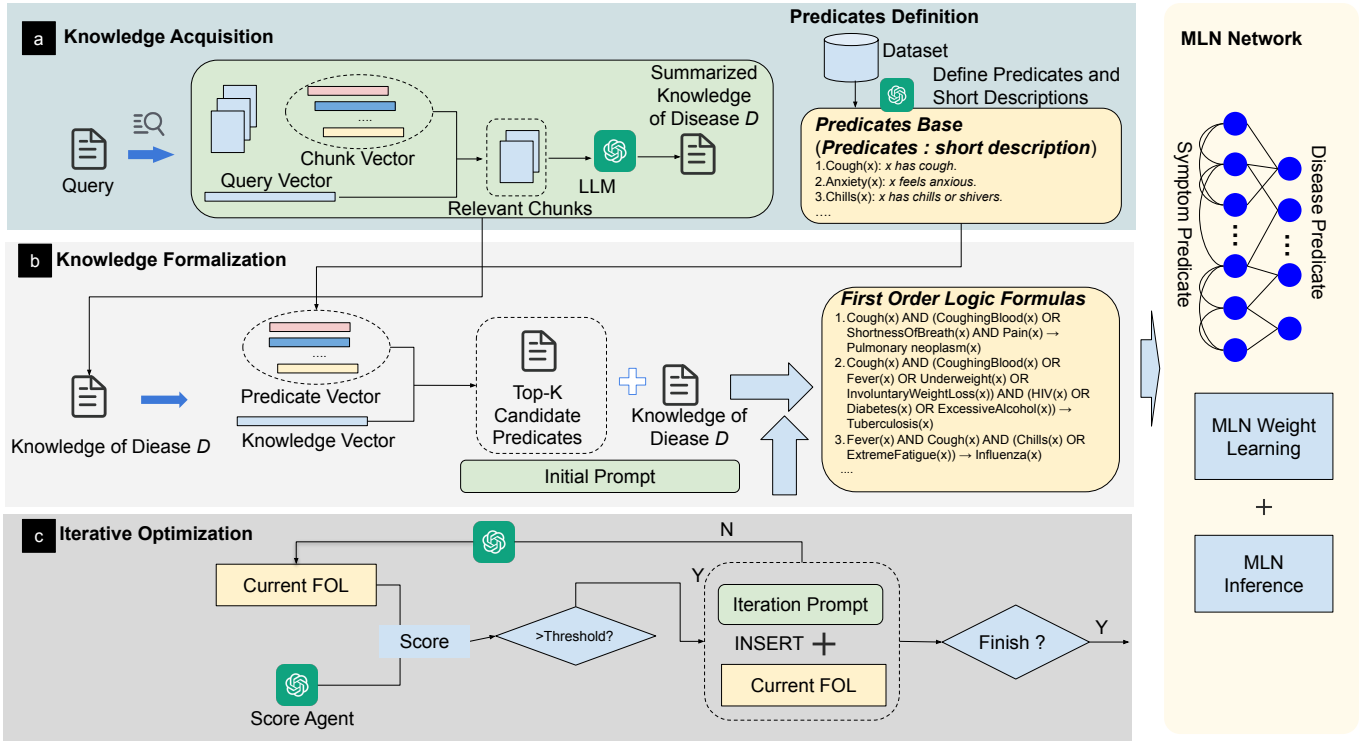

Fig. 1. The architecture of the proposed framework includes three critical steps: (a) Knowledge Acquisition, (b) Knowledge Formalization, and (c) Iterative Optimization. In particular, the Knowledge Acquisition step involves a search engine and LLMs to automatically collect and process critical domain knowledge. Then, the Knowledge Formalization step transforms the gathered domain knowledge into FOL using LLMs. Finally, the Iterative Optimization step enhances the quality of FOL through multiple iterations. After these three steps, our framework accumulates high-quality FOL rules of medical knowledge that will be used for the following MLN training and inference, resulting in an interpretable medical diagnosis system.

for automated diagnosis. Our pipeline supports the automated formulation of professional domain knowledge and employs the MLN to perform interpretable automatic reasoning.

A. Knowledge Acquisition

We begin by detailing our strategy for automatic medical knowledge acquisition, which leverages the capabilities of both search engines and LLMs. This process aims for the automated collection and processing of critical domain knowledge.

Initially, we craft queries for knowledge search. For instance, we could use prompts such as: "What are the symptoms of Pneumonia?" and then replace 'Pneumonia' with the disease of our interest. To ensure we retrieve reliable medical knowledge, our search is confined to authoritative medical and biomedical resources, such as the National Library of Medicine[1]. The search engine then returns the top K most relevant web pages. And in our study, K is set to 5.

Following this, to collect more accurate medical knowledge related to the query, we apply further data processing to the content of the retrieved web pages. Our target in this step is to filter out any content that is not relevant to the initial query. To achieve this objective, we employ an embedding model [25] to calculate the cosine similarity between the query and each instance of the web pages. Subsequently, we select the top K instances and also apply a threshold to identify the content most relevant to the query. This comprehensive approach to knowledge acquisition allows us to gather precise and useful information from trusted sources.

After the above-mentioned steps, we further utilize LLMs to summarize the most relevant content we obtained. Here, we try to harness the strong summarization capability of the LLMs by trying different prompts. Finally, the prompt used is as follows.

Medical Knowledge Summarization
Here is the content of the medical knowledge of D:
CHUNK 1
CHUNK 2
...
CHUNK K
Summarize the above medical knowledge chunks with the query:{query} within 500 words.

In our study, the LLMs demonstrate promising performance in correctly summarizing a large amount of medical knowledge into specific knowledge we want. By all the above-mentioned steps, we finally achieve automatic accurate medical knowledge acquisition with low cost and limited human effort.

B. Knowledge Formalization

In this subsection, we describe the process of converting natural language-form knowledge into first-order logic (FOL) form. FOL rules can be expressed as an implication operation ($\Rightarrow$) with a conjunction of predicates on the left and a single

[1]https://www.ncbi.nlm.nih.gov/

Algorithm 1 Knowledge Acquisition

Input: Query Q
Definition: Search refers to the search engine, SearchScope represents the websites to be used during the search, LLM_Summarize means that use LLM to summarize the content.
Output: Knowledge related to a disease D
Begin Search:
Restrict the search scope to SearchScope
Execute the search with the Query Q, retrieve top n relevant web pages $P = p_1, p_2, ..., p_n$.
Begin Processing:
for $i = 1$ to n **do**
    Divide p_i into chunks $C_i = c_{i1}, c_{i2}, ..., c_{im}$.
    Calculate the cosine similarity $S_{ij} = \frac{Q \cdot c_{ij}}{\|Q\| \|c_{ij}\|}$, for each chunk $c_{ij} \in C_i$.
end for
Compute cosine similarities for all chunks: $S_{ij} = \frac{Q \cdot c_{ij}}{\|Q\| \|c_{ij}\|}$ for each $c_{ij} \in C_i$.
Order chunks by similarity and select the top k: $C' = \{c_j | j \in \text{top}_K(Q)\}$.
End Processing.
Begin Summarization:
$D = LLM\_summarize(C')$
End Summarization.
Return: Return the Knowledge of D.
End Search.

predicate on the right. In other words, a rule can be expressed in the form of "P AND Q AND ... ⇒ R", where P, Q, and R are predicates. The implication in this format denotes that if P, Q, etc. are all true, then R is also true. Our objective of this Knowledge Formalization step is to convert natural language-form medical knowledge into such FOL rules.

While MLNs could offer interpretability in medical reasoning, creating an efficient MLN requires the formulation of high-quality FOL rules from medical knowledge, and this task was previously undertaken by medical experts. In our study, we tune into the information extraction and integration capabilities of LLMs. That is, we generate FOL expressions directly by integrating summarization content into the prompt as in-context information via prompting the LLMs.

In traditional methods, defining the predicate is the first step when formulating a FOL expression. Similarly, despite their exceptional information extraction abilities, LLMs like ChatGPT and GPT-4 cannot produce effective FOL expressions without pre-defining predicates. These pre-defined predicates play a critical role for the LLMs to avoid using varying descriptions for the same symptom, such as representing the symptom "shortness of breath" as "ShortnessOfBreath" or "ShortOfBreath". This inconsistency is not conducive to subsequent training and reasoning.

In our implementation, we start by defining all predicates and their corresponding meanings in natural language. For instance, let's consider the symptom "cough". We can represent it in predicate form as $Cough(x)$, where x is a placeholder for the patient. The meaning of this predicate $Cough(x)$ can be simply interpreted as "individual x is experiencing a cough". Incorporating all predicates and their short descriptions into the context could easily surpass the window limit of existing LLMs. To solve this problem, we select $K_{Candidate}$ candidate predicates based on the cosine similarity between the short descriptions of the predicates and the disease knowledge. Here, $K_{Candidate}$ is set to 20.

Upon formalizing the templates of FOL for each disease, we integrate these candidate predicates and medical knowledge into the in-context, enabling the LLM to formulate corresponding logical rules for diagnosis. However, this process does not guarantee the appropriateness of the generated response. To enhance the quality of these logic rules, we employ a self-improvement mechanism to refine them. Specifically, the self-improvement mechanism is an iterative optimization process, with the details described in Algorithm 2.

Algorithm 2 Iterative Optimization Algorithm

Input: Initial prompt with disease knowledge and the candidate predicates
Initialize: i=0, N_{Iter}=Iteration times, T_{score}=defined threshold score
Definition: $Prompt_{Initial}$ denotes our initial prompt, $Prompt_{Iter}$ denotes our iterative prompt, $ChatGPT_{score}$ means score agent
Function: $Insert$ denotes the action of inserting the FOL into the placeholder of the $\{previous\_answer\}$
$Initial\_FOL = ChatGPT(Prompt_{Initial})$
$Prompt_{Iter} = Insert(Prompt_{Iter}, Initial\_FOL)$
while $i < N_{Iter}$ **do**
    $Current\_FOL = ChatGPT(Prompt_{Iter})$
    $Score = ChatGPT_{score}(Current\_FOL)$
    if $Score > T_{score}$ **then**
        $Prompt_{Iter} = Insert(Prompt_{Iter}, Current\_FOL)$
    end if
    $i++$
end while

Knowledge Formalization
Formalize the following knowledge into first-order logical expression.
Here are the examples for you:
{examples}
Here is the knowledge of D
{knowledge}
(End of knowledge)
Candidate predicates:
{candidate_predicate}
Now output a first-order logical expression for diagnosing D:

Overall, the algorithm is inspired by the self-refinement approach for LLM [18], and it leverages the few-shot learning capabilities of LLMs to iteratively improve their responses.

This design not only enables the model to learn how to score its outputs but also incrementally enhances the quality of the generated FOL rules. Initially, the input of the algorithm comprises the prompt with $K_{Candidate}$ selected candidate predicates for each disease and their corresponding knowledge. In the initial iteration, the algorithm feeds this prompt directly into the LLM.

Upon initiation of the FOL, the optimization process begins. During each iteration, a scoring agent evaluates the quality of the generated FOL with a range of 0 to 20. If the resulting score exceeds a predefined threshold (i.e., 15), the quality of this generated FOL is considered to be satisfactory. Then, this FOL will be added to the iterative prompt, enriching the context of the LLMs. By doing this, it progressively refines the generated FOL. The number of iterations, denoted as N_{Iter}, can be set empirically. In this study, we set its value to 10.

In this algorithm, the Evaluation of the FOL and the Iterative Optimization modules play a critical role and we would like to give a detailed illustration.

1) Evaluation of FOL: During the process of FOL extraction using LLM, we observed four common types of errors:

- Incorrect usage of symbols;
- Misapplication of predicates, such as using non-existent predicates to represent related medical knowledge;
- Inaccurate depiction of logical relationships within the medical knowledge;
- Failure to fully express medical knowledge in first-order logic.

Our scoring agent primarily focuses on these errors. We give three examples in the context, allowing the LLM to learn how to evaluate the quality of the FOL effectively. Moreover, the LLM can learn to provide suggestions for improvements that can be implemented in the next iteration.

2) Iterative Optimization: We implement iterative prompts to gradually improve the quality of LLM's responses. This process involves merging the previous answers along with their corresponding actionable suggestions into the iterative prompts. The number of iterations can be decided empirically or based on the computational resources. By using iterative prompts, we not only achieve improved accuracy in medical diagnoses but also maintain model interpretability.

Iterative Optimization
Formalize the following knowledge into first-order logical expression.
Here is the knowledge of D
{knowledge}
(End of knowledge)
Candidate predicates:
{candidate_predicate}
Your previous answer:
{previous_answer}
Now you should refine the first-order logical expression you output before.

C. Learning and Inference with Markov Logic Network

In this subsection, we illustrate how we employ the formulated FOL for learning and reasoning using a Markov Logic Network (MLN).

An MLN is essentially a collection of weighted first-order formulas that are used as blueprints for creating Markov networks. To enable these first-order formulas to learn their respective weights, we construct an evidence database from our dataset. During the inference step, the MLN serves as a probabilistic logic solver that performs soft reasoning rather than deterministic hard reasoning.

Inspired by [14], we design a dual-model system that is composed of symptom inquiry and diagnosis prediction. Specifically, we utilize the Diaformer [2] to generate a set of potential symptoms, and an MLN to predict the final diagnosis. This approach allows for a comprehensive representation of the medical diagnostic process.

Algorithm 3 Weight Learning and Inference of MLN

Input: Predicates, unweighted Rules, Evidences
Definition: Learnwts means the function to learn each weight of rules, Infer means the function to do inference
Output: WeightedRules, Inference Results
Build evidence database:
Extract the evidence in predicates set from patient records in the training and the testing sets $\rightarrow$ $Evidences_{train}$, $Evidences_{test}$
Begin learning weights:
Initialize the weights of Rules by zero
$Learnwts(Predicates, Rules, Evidences_{train}) \rightarrow WeightedRules$
End learning weights
Begin Inference
for $case_i$ in $Evidences_{test}$ **do**
  for D in $\mathcal{D}$ **do**
    $P_D = Infer(WeightedRules, Evidence_{case_i})$
  end for
  $Diagnosis_{case_i} = \arg\max_{D \in \mathcal{D}} P_D$
end for
End Inference

1) Weight Learning: We obtain the training evidence database by leveraging the true positive symptom predicates generated by a well-trained Diaformer. Although Diaformer may not always output all positive symptoms, MLN can conduct inferences based on this incomplete information. For instance, a training evidence database may look as follows:

SoreThroat (Train_PatientID$_1$)
ConsultingPain (Train_PatientID$_1$)
CoughColoredSputum (Train_PatientID$_1$)
Cough (Train_PatientID$_1$)
IncreasedSweating (Train_PatientID$_1$)
Fever (Train_PatientID$_1$)
URTI (Train_PatientID$_1$)

where named like "SoreThroat" refers to the disease and Train_PatientID$_1$ denotes the ID of the patent. Disease predicates, like URTI (Train_PatientID$_1$), are present in the training database but absent from the test database. Specifically, these predicates are employed as labels during the MLN training process. A patient's ID following a predicate indicates that the patient exhibits the evidence represented by that predicate. Diaformer can generate a sequence of symptoms to inquiry patients based on some given explicit symptom. We simulate the process using the training set. For the patient in the dataset, we use Diaformer to generate a sequence of symptoms. Then we discarded those symptoms that the patient didn't have. In this process, Diaformer may not be able to completely generate all the true positive symptoms of patients in the training set in some cases, however, the MLN can handle these cases and conduct the training with incomplete information.

The network is trained with discriminative learning and it optimizes the model's ability to predict disease predicates Y from evidence X. The We maximize the conditional probability $P(y|x)$, calculated as

$$P(y|x) = \frac{1}{Z_x} exp(\sum_{i \in F_i} w_i n_i(x, y)),$$

where F_Y contains all the MLN formulas that are grounded to the disease predicate, w_i refers to the formula's weight, and $n_i(x, y)$ denotes the total number of true groundings of the i^{th} formula involving disease predicate. In particular, we formulate the rules as an array of linear equations and employ the conjugate gradient algorithm to iteratively compute the weights. Rules that are satisfied more frequently in the data are more effective, so their weights become larger. After learning, rules with greater weights are more probable based on the evidence.

2) Inference: The training process will produce a set of weighted rules, which could be further employed by the MLN to infer diseases on the testing set. During the inference stage, our approach follows the symptom sequences from Diaformer, but the final diagnosis is made by the MLN. In particular, Diaformer generates the symptom sequence and we extract the true positives and input them to the MLN to reason about the disease. As mentioned previously, MLN represents the joint probabilities of observed and unobserved variables. For diagnosis, we mark disease predicates as queries so diagnosis becomes a multi-class classification task. We then use MCMC sampling to infer the marginal probability of each disease given the evidence. We determine the final predicted disease by selecting the one with maximum marginal probability based on the MLN inference as

$$Diagnosis_{case_i} = \arg\max_{D \in \mathcal{D}} P(D, case_{\mathrm{i}})$$

IV. Experiments and Analysis

In this section, we describe our experimental setup and report the evaluation results. Through comparative analysis with other baselines, we demonstrate the efficacy of our model in the field of disease diagnosis inference.

A. Dataset

Our experiments are conducted on a series of both real-world and synthetic datasets [4]. The real-world datasets, DXY [5] and Muzhi [3], are collected from the conversations between patients and doctors. The Synthetic [4] dataset is a much larger dataset constructed by the symptom-disease dataset. The statistics of these datasets are shown in Table I.

TABLE I
STATISTICS OF THE DATASETS

Statistics	Muzhi	DXY	Synthetic
Number of diseases	4	5	90
Number of symptoms	66	41	444
Number of cases in training set	568	423	24000
Number of cases in test set	142	104	6000

For better illustration, we also show an instance from the DXY dataset. In this example, the patient tells the doctor about his/her symptoms like *"Cough", "Runny nose"*. The doctor, in response, poses two queries: *"Do you have Allergy?"* and *"Do you have Sneeze?"*. The patient then provides the relevant responses. An ideal trained model is expected to act as the doctor's role, posing queries to the patient and making a diagnosis based on the responses.

"disease": "URTI"
"explicitly informed findings":["Cough": True, "Runny Nose":True],
"implicitly informed findings":["Allgery": False, "Sneeze":True]

B. Implement Details

As mentioned in Section 3.2, to extract FOL as knowledge, we need to define predicates in each dataset. We can directly use the symptoms from the dataset, we just transform the symptoms and disease into corresponding predicates, and the short description of the predicate is generated by ChatGPT.

These descriptions, providing semantic insight into the predicate, are crucial for a couple of reasons. Firstly, It can provide better results for the calculation of cosine similarity with medical knowledge text in experiments. Secondly, the semantic understandability of these predicates amplifies the interpretability of the FOL expressions in the context of medical diagnosis.

In our experiments, we employ a variety of LLMs and other tools. ChatGPT-3.5-turbo is used to summarize the disease knowledge, text-embedding-ada-002 for document embeddings, GPT-4 for formalizing the medical knowledge, and an iterative optimization process to refine the FOL iteratively. To learn the weights of the rules and perform inference, we utilize Alchemy as the Markov Logic Network (MLN), and also employ the Markov Chain Monte Carlo (MCMC) sampling method for inference.

C. Performance Comparison

Following the settings of Diaformer, we deploy a Support Vector Machine (denoted as **SVM-exp**) to predict diseases based on explicit symptoms alone, without inquiring about any symptom. This SVM-exp method serves as a baseline with limited diagnostic performance. Several competitive Reinforcement Learning-based models are also utilized for comparison, including Flat-DQN [3], HRL [4], KR-DS [5], GAMP [6], and PPO [7], as well as the original Diaformer. We directly report the results given by these original papers.

In addition, we also compare our model with generative LLMs such as ChatGPT. However, it is quite challenging to directly query the LLM without specific training on the LLM construct process. Therefore, we allow the LLM to access all the true positive symptom sequences directly, and make predictions solely from the candidate diseases. Given the huge volume of candidate diseases in the synthetic dataset and the considerable size of the dataset, we limit our experiments to the Muzhi and DXY datasets. To obtain better results, we also incorporate the technique of self-consistency when conducting few-shot learning using the LLM. In other words, we query ChatGPT output multiple times (N=5 in our experiment) and the answer with the highest probability is chosen.

TABLE II
EXPERIMENTAL RESULTS (CLASSIFICATION ACCURACY (%)) OF OUR METHOD COMPARED WITH OTHER BASELINES.

Method	Muzhi	DXY	Synthetic	Average
SVM-exp	67.3	64.0	34.1	55.1
Flat-DQN	69.0	72.0	35.6	58.8
HRL	69.4	69.5	49.6	62.8
KR-DS	73.0	74.0	-	-
GAMP	73.0	76.9	-	-
PPO	73.2	74.6	61.8	69.8
Diaformer	74.2	83.9	**73.3**	77.1
Ours	**76.4**	**84.9**	72.9	**78.0**

Overall, our proposed framework can achieve competitive results with Diaformer and ChatGPT. Our framework outperforms other models on both Muzhi and DXY. But on Synthetic our model is slightly weaker than Diaforemr. In terms of average performance, we are higher than Diaformer. This indicates that as the number of features and symptoms increase, our model shows some decrease in performance, which can be seen as a cost for interpretability.

TABLE III
PERFORMANCE COMPARISON WITH CHATGPT

Method	Muzhi	DXY	Average
ChatGPT	64.1	65.4	64.8
ChatGPT + Self-Consistency	64.8	68.3	66.6
Ours	**76.4**	**84.9**	**80.7**

D. Ablation Study: Impacts of Knowledge Integration and Iterative Optimization

In this section, we conduct an ablation study to assess the influence of our knowledge-based FOL and iterative optimization on the performance of the model. We check the accuracy under different configurations: i) utilizing a conventional MLN without predefined FOL (**MLN**); ii) removing our summarization module and directly truncating the searched content when it reaches the length limit (4096 tokens) (**w/o SUM**); and iii) removing the iterative optimization module (**w/o IO**).

The results of the ablation study, presented in Table IV, confirm the importance of the designed summarization module and iterative optimization step in our proposed model. We observe that iterative optimization plays an essential role in knowledge formalization, as it significantly enhances the quality of the FOL. The summarization module is also found to be very important since it helps the LLM focus on important information and alleviates errors when the initial input context is lengthy. By reducing the context length, the summarization module could also reduce the cost during the iterative optimization process. As such, both of these two design mechanisms contribute to forming a higher-quality FOL with medical knowledge.

TABLE IV
ABLATION STUDY ON THE EFFECTS OF KNOWLEDGE AND REFINEMENT

Method	Muzhi	DXY	Synthetic
MLN	28.4	30.3	16.4
Ours (w/o SUM + IO)	48.6	54.8	46.4
Ours (w/o IO)	50.2	60.0	61.8
Ours (w/o SUM)	73.0	78.8	68.2
Ours	**76.4**	**84.9**	**72.9**

E. Case Study: Illustrating the Interpretability of MLN

In this section, we utilize a case to vividly show the interpretability of our proposed framework. Intuitively, our framework facilitates the selection of the most fitting FOL expressions for a patient's condition. These utilized FOL expressions shed light on the reasoning process of the framework and increase the interpretability of the model.

Consider a patient with two explicit symptoms: *Ankle pain* and *fever*. We employ the Diaformer model to simulate an inquiry session with the patient. Diaformer proposes one symptom at a time and a patient simulator decides whether the proposed symptom is within the set of implicit symptoms. If the symptom is not within the set, the model suggests the next symptom with the highest probability for inquiry. If the proposed symptom is indeed an implicit one, the patient simulator replies with a confirmation (True) or denial (False). This symptom is then added to the sequence of the symptoms, and the model proceeds to predict new probabilities for the next inquiry symptom. Eventually, other symptoms such as *Sore throat*, *Shoulder cramps* or *spasms*, and *Pain during pregnancy* are determined.

FOL rules related to these symptoms are:

- Rule (1): "Ankle pain(x) AND Fever(x) AND Sore throat(x) AND Shoulder cramps or spasms(X) $\Rightarrow$ Dengue fever(x)"
- Rule (2): "Shoulder cramps or spasms(x) AND Ankle pain(x) $\Rightarrow$ Gas gangrene(x)"
- Rule (3): "Shoulder cramps or spasms(x) AND Ankle pain(x) AND Wrist pain(x) $\Rightarrow$ Air embolism(x)"

These rules imply potential diagnoses for the patient, and they enjoy different weights of 8.2, 1.7, and 2.8, respectively. As such, the rule with the maximum inference probability, that is, Rule (1) will be chosen as the most relevant rule and leads to the final diagnosis of *"Dengue fever"*.

V. Conclusion

In this paper, we developed an innovative approach that leverages the capabilities of large language models for automatic knowledge extraction and formalization, combined with the interpretability of Markov logic networks for medical diagnosis. Our framework delivers both impressive performance and robust interpretability, showing effectiveness across diverse datasets and outperforming both traditional RL methods and generative models. This work not only embodies a substantial leap forward in AI applications for healthcare but also demonstrates the powerful future of combining large language models with logical reasoning. We hope the findings of this study will shed light on exciting opportunities for future research and development of advanced, interpretable, and AI-driven medical diagnosis systems.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments and effort to improve this manuscript. Yuanfeng Song is the corresponding author.

References

[1] Kai-Fu Tang, Hao-Cheng Kao, Chun-Nan Chou, and Edward Y Chang. Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. In *NIPS workshop on deep reinforcement learning*, 2016.

[2] Junying Chen, Dongfang Li, Qingcai Chen, Wenxiu Zhou, and Xin Liu. Diaformer: Automatic diagnosis via symptoms sequence generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4432–4440, 2022.

[3] Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuan-Jing Huang, Kam-Fai Wong, and Xiang Dai. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–207, 2018.

[4] Kangenbei Liao, Qianlong Liu, Zhongyu Wei, Baolin Peng, Qin Chen, Weijian Sun, and Xuanjing Huang. Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning. *arXiv preprint arXiv:2004.14254*, 2020.

[5] Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7346–7353, 2019.

[6] Yuan Xia, Jingbo Zhou, Zhenhui Shi, Chao Lu, and Haifeng Huang. Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1062–1069, 2020.

[7] Milene Santos Teixeira, Vinícius Maran, and Mauro Dragoni. The interplay of a conversational ontology and ai planning for health dialogue management. In *Proceedings of the 36th annual ACM symposium on applied computing*, pages 611–619, 2021.

[8] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, pages 1–9, 2023.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[10] Sanghwan Bae, Donghyun Kwak, Sungdong Kim, Donghoon Ham, Soyoung Kang, Sang-Woo Lee, and Woomyoung Park. Building a role specified open-domain dialogue system leveraging large-scale language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2128–2150, 2022.

[11] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62:107–136, 2006.

[12] Yoichi Hayashi. A neural expert system with automated extraction of fuzzy if-then rules and its application to medical diagnosis. *Advances in neural information processing systems*, 3, 1990.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[14] Aleksandr Nesterov, Bulat Ibragimov, Dmitriy Umerenkov, Artem Shelmanov, Galina Zubkova, and Vladimir Kokh. Neuralsympcheck: A symptom checking and disease diagnostic neural model with logic regularization. In *International Conference on Artificial Intelligence in Medicine*, pages 76–87. Springer, 2022.

[15] Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*, 2023.

[16] Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, 2022.

[17] David Kartchner, Selvi Ramalingam, Irfan Al-Hussaini, Olivia Kronick, and Cassie Mitchell. Zero-shot information extraction for clinical meta-analysis using large language models. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 396–405, Toronto, Canada, July 2023. Association for Computational Linguistics.

[18] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

[19] Varun Nair, Elliot Schumacher, Geoffrey Tso, and Anitha Kannan. Dera: enhancing large language model completions with dialog-enabled resolving agents. *arXiv preprint arXiv:2303.17071*, 2023.

[20] Chong Ma, Zihao Wu, Jiaqi Wang, Shaochen Xu, Yaonai Wei, Zhengliang Liu, Lei Guo, Xiaoyan Cai, Shu Zhang, Tuo Zhang, et al. Impressiongpt: an iterative optimizing framework for radiology report summarization with chatgpt. *arXiv preprint arXiv:2304.08448*, 2023.

[21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[22] Hangfeng He, Hongming Zhang, and Dan Roth. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*, 2022.

[23] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.

[24] Pedro Domingos, Daniel Lowd, Stanley Kok, Aniruddh Nath, Hoifung Poon, Matthew Richardson, and Parag Singla. Unifying logical and statistical ai. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–11, 2016.

[25] Ryan Greene, Ted Sanders, Lilian Weng, and Arvind Neelakantan. New and improved embedding model. *OpenAI blog*, 2022.