Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

mCodeGPT: Enhancing Cancer Research through Zero-Shot Information Extraction from Clinical Free Text Data

Xiaoqian Jiang (xiaoqian.jiang@uth.tmc.edu)
University of Texas Health Science Center at Houston https://orcid.org/0000-0001-9933-2205
Kai Zhang
School of Biomedical Informatics, University of Texas Health Science Center at Houston
https://orcid.org/0000-0002-4519-609X
Tongtong Huang
University of Texas Health Science Center at Houston https://orcid.org/0000-0002-4007-8991
Bradley Malin
Vanderbilt University Medical Center
Travis Osterman
Vanderbilt University of Medical Center
Qi Long
University of Pennsylvania Perelman School of Medicine https://orcid.org/0000-0003-0660-5230

Article

Keywords:

Posted Date: February 2nd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3882757/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: (Not answered)

mCodeGPT: Enhancing Cancer Research through Zero-Shot Information Extraction from Clinical Free Text Data

Kai Zhang[1], Tongtong Huang[1], Bradley A Malin[2],

Travis Osterman[2], Qi Long[3], Xiaoqian Jiang[1*]

[1]Department of Data Science and Artificial Intelligence,
McWilliams School of Biomedical Informatics,
University of Texas Health Science Center at Houston, Houston, TX 77030
{kai.zhang.1, tongtong.huang, xiaoqian.jiang}@uth.tmc.edu

[2]Department of Biomedical Informatics, Vanderbilt University Medical Center, Nashville, TN 37203
{b.malin, travis.osterman}@vumc.org,

[3]Department of Biostatistics, Epidemiology and Informatics,
Perelman School of Medicine,
University of Pennsylvania, Philadelphia, PA 19104
{qlong}@pennmedicine.upenn.edu

* Corresponding author

Abstract

The rapidly growing collection of clinical free text data about cancer provides an unprecedented opportunity to harness rich, real-world insights for advancing cancer research and treatment. At the same time, there is a formidable challenge in efficiently extracting, standardizing, and structuring this vast and unstructured information to derive meaningful conclusions. In this paper, we address the problem of information extraction to standardize and structure such free text using Large Language Models (LLMs) following the minimal Common Oncology Data Elements (mCODE™) structure. We propose mCodeGPT, a state-of-the-art data standardization tool Cancer related clinical notes. mCodeGPT ingests Cancer ontology knowledge base, automatically generates prompts in a

hierarchical manner to extract information within the ontology from the clinical notes, and outputs results in a structured, tabular format. The pipeline is completely zero-shot, meaning it is annotation-free and does not require model training. Our new tool simplifies the conventional labor-intensive process of data collection, annotating, model training, and fine-tuning. To overcome challenges inherent in common LLMs, such as token size limitations, hallucination, and suboptimal accuracy in information extraction, we have developed three distinct methodologies for hierarchical prompt engineering. We conducted rigorous evaluations of mCodeGPT's efficacy in the domain of Cancer clinical note generation and information extraction. Employing the GPT-3.5-turbo model, we generated a diverse set of 400 simulated clinical notes spanning various Cancer types. The effectiveness of our three proposed prompt generation methods was further assessed using both OpenAI models and a selection of open-source LLM models. The outcomes of our research underscore the proficiency of our pipeline, showcasing an impressive 96% accuracy in information extraction. We published mCodeGPT on GitHub for other researchers to use.

1 Introduction

Cancer research has evolved dramatically with the emergence of modern technologies, generating an unprecedented amount of data[1]. The process of diagnosing, formulating treatment plans, conducting medical assessments, and creating genetic profiles generates a vast and diverse volume of information, which presents both opportunities and challenges in organizing and utilizing such data effectively for medical providers and researchers[2]. The core of the problem is attributed to the heterogeneous large amount of information in an unstructured data format, which hinders data interoperability because of a lack of an efficient and accurate tool to identify and extract all useful clinical information from the irrelevant narratives[3]. In EHR (electronic health records), unstructured data accounts for nearly 80% information of the entire data[4]. The reason is, many clinical procedures and findings, such as physician notes, radiology reports, and pathology findings, are often kept in the form as when they were first

created – in the form of free-text[5]. Despite its inherent complexity, it contains valuable information that can significantly advance Cancer research and patient care. These unstructured data sources can provide crucial details about a patient's medical history, treatment responses, and disease progression that are indispensable for developing personalized treatment plans and conducting research into novel therapies.

Standardized data serves as a crucial tool in attaining consistent results, and guarantees seamless integration of data from diverse sources, thereby fostering collaborative research initiatives[6]. The challenge of converting unstructured data to a structured format is great when dealing with specialized domains like Cancer, where the medical terminology and concepts are highly specific[7]. The topic of data standardization from clinical notes largely focuses on the concept of *Information-Extraction*, which congenitally includes *Term-Extraction*, *Relation-Extraction,* and the Natural Language Processing (NLP) techniques around these two tasks[8]. Conventional NLP methodologies often depend on manually curated rules and human-annotated datasets for their training procedures. This results in Named Entity Recognition (NER) or information extraction systems being less generalizable, therefore, exhibiting limited portability[9]. This pertains to the widely recognized issue of out-of-distribution prediction, wherein these systems may exhibit sub-optimal performance when deployed in novel or disparate contexts, a consequence of their substantial dependence on prior training data and labels.

OntoGPT is a novel solution using LLMs to query and extract ontology information from free text[10]. It creates or uses existing ontology and knowledge base and represents them in a YAML format, auto-interpret the YAML template and feed into LLMs for query-answering, and finally get the outputs specified to a pre-defined format. However, OntoGPT serves for general ontology query-answer tasks, for instance, food recipes, disease treatment, gene pathways, etc., and not specifically designed for domains such as Cancer ontology. For this reason, it has two key shortcomings:

- *YAML language has a sharp learning curve and human-labor is involved in creating a YAML template for an ontology or knowledge-based system.* OntoGPT requires representing ontology in YAML language which has its own syntax and grammar. Moreover, the building of YAML templates is not automated and requires large amount of human efforts.
- *Only effectively work for small ontology and input text, and often hit the token restrictions of LLMs*. OntoGPT-compatible ontology templates typically contain one to up to twenty named entities and relations. However, this type of production is too error-prone and time-consuming for large knowledge bases such as mCODE. Also, large ontologies like mCODE are unable to operate because the token limitations of large language models are readily exceeded.

To get beyond the token restrictions, an easier ontology representation toolkit, a hierarchical ontology processing and prompt engineering to get beyond the token restrictions, hierarchical ontology processing and strategic prompt engineering is thus required is thus required.

Consequently, we put forth a novel pipeline that markedly expands on the original study by 1) **Automating the ontology template generation process**; 2) **Introducing three pioneering hierarchical prompt generation and result refinement algorithms to bypass the token limitation problem**. First, the ontology usually embeds an intricate hierarchical structure with various relations between the ontology named entities (such as parent-child relation). This kind of structure can naturally be captured via a graphical model. The human process of YAML ontology template file preparation can be replaced by an automatically graph traversal algorithm, which automatically generates prompt through traversing on the graphical model from root to leaf. The algorithm significantly automates the process of cancer ontology template preparing hurdle. Second, we solve the token limitation problem by prompt engineering size (for example, 4K tokens for GPT-3.5-turbo, 16K tokens for GPT-3.5-turbo-16k, and 32K tokens for GPT-4-32k. 100 tokens are approximately 75 words). Specifically, we proposed three hierarchical prompt generation pipelines that strategically perform query-answering and then refine the prompts based on previous rounds answer by LLM.

Unlike OntoGPT, which combines all questions and input text into a single prompt and allows LLM to extract information, our approach is different from one-shot query-answering. The **major contributions** of the proposed tool are summarized as follows:

- **Annotation-Free Standardization.** We introduce mCodeGPT, leveraging the power of LLM, along with novel prompt engineering algorithms, to automatically extract unstructured clinical notes into standardized tables, validating to the mCODE Cancer ontology, Figure 1. Our pipeline performs zero-shot learning and is annotation-free.
- **Ontology Representation and Parsing.** We propose a novel graphical model to represent ontology, which can naturally preserve all ontology information in graph nodes (concepts) and edges (relations). We also design functions to traverse on the graph to automatically generate query-answering prompts from the graphical model for named entity extraction.
- **Prompt Refinement Algorithms.** Our hierarchically prompt generation method not only bypass the common LLMs' token count restriction, but also yield more accurate result through prompt refinement (refine the prompt based on past responses). The essential idea is to perform multi-shot query-answering, each querying only a subset of the entire ontology. The entire process is performed according to a hierarchical structure.

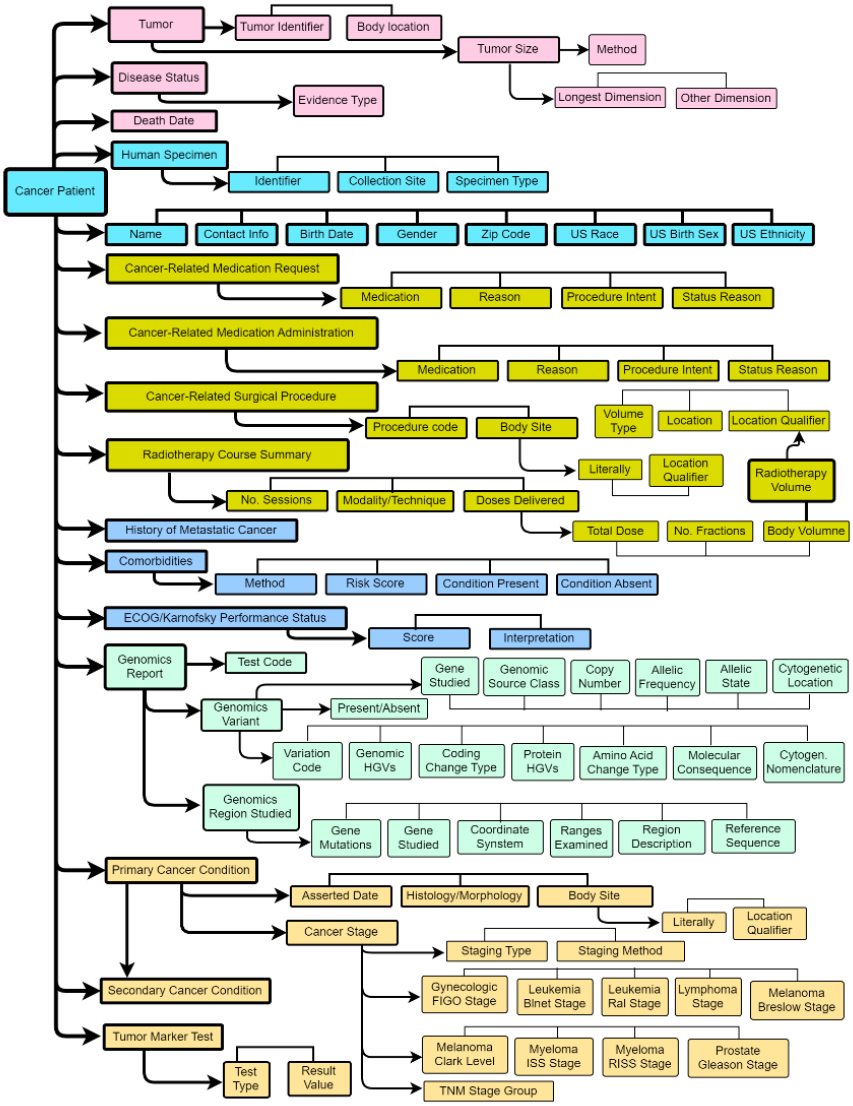


Figure 1. The mCODE structure from *www.fhir.org*. Different thematic groups are in different colors. Each box denotes a named entity. Arrows indicate an "attributes" relationship between two named entities. The named entities connected by line without arrow indicates they are in parallel all attributes of the parent.

2 Backgrounds

Information extraction has emerged as a promising approach to transform free-text cancer information into structured data. It involves a series of NLP techniques including named entity recognition, term extraction, relationship extraction and machine learning.

2.1 Named Entity Recognition

NER is a sub-task of information extraction that classifies named entities into predefined categories such as the person names, organizations, locations, medical codes, time expressions, diseases, and

symptoms. There are tools and libraries including Natural Language Toolkit (NLTK)[11], spaCy, and Stanford NLP[12].

In biomedical domain, abbreviations and acronyms are the largest challenge that traditional NER techniques face[13,14]. Medical texts can be long and commonly use abbreviations and acronyms to represent themselves, which can have multiple meanings in different contexts. For instance, "CRC" could mean "colorectal cancer" or "clinical research coordinator" based on context. In the meantime, variability in entity expression is another challenge where different healthcare providers or researchers might describe the same medical condition or procedure in various ways. For example, the medical condition “myocardial infarction” can be represented as the following entity expressions that are equivalent in meanings, General Public: "Heart attack"; Medical Professional: "MI"; Medical Research Paper: "Acute myocardial infarction (AMI)"; Clinical Documentation: "ST-segment elevation myocardial infarction (STEMI)" or "Non-ST segment elevation myocardial infarction (NSTEMI)"; Patient's Personal Description: "Heart event" or "heart episode". The Unified Medical Language System (UMLS) is discussed as an effort to integrate and harmonize various biomedical terminologies[15].

Recent models like BERT[16], RoBERTa[17], and their variants have shown great promise in NER tasks. By being trained on vast corpora, these models are more familiar with diverse linguistic constructs, abbreviations, and entity variations. Fine-tuning these models on domain-specific data can enhance their recognition ability. Some research has specifically focused on disambiguating abbreviations and acronyms in clinical texts. Techniques often combine supervised machine learning with resources like medical thesauri[18]. Medical ontologies and taxonomies like UMLS (Unified Medical Language System) or SNOMED CT can be valuable resources to map various synonyms, acronyms, and abbreviations to standard medical terms[19]. Generative models or rule-based methods can be used to introduce variability in training data, thus teaching models about different ways an entity can be expressed[20,21]. Especially in domains like radiology, combining text with images can help

disambiguate certain terms or provide context to entity mentions[22]. Combining predictions from different models or training a model to perform multiple related tasks can help in disambiguating challenging entity expressions.

2.2 Cancer Ontology

Initial cancer ontologies were rudimentary, primarily focused on cataloging histological types and anatomical locations of tumors[23]. With the advances in molecular biology and genetics, subsequent cancer ontology study started incorporating molecular and genetic markers, and study the heterogeneity of tumors and their different pathways[24]. The National Center for Biomedical Ontology (NCBO) provides a platform BioPortal, including over 270 biomedical ontologies and terminologies. This has substantial implications in ontology study, especially for enhanced data interpretation.

3 Methods

In order to efficiently represent the named entities and their relations in the ontology, a proper data structure is needed. This is for the purpose both data representation and efficiently automatically generate prompt for information extraction. In this section, we introduce the data structure to efficiently represent ontology. We also introduce three prompt engineering methods to bypass the LLM token limitation.

3.1 Graphical Model

An ontology formalizes a shared understanding of concepts, or named entities, and their relations - typically in a hierarchical structure from general to specific. To represent named entities and their relations within the ontology efficiently, we leverage a graph structure, not only due to its efficiency for data representation but also its potential to automate the prompts generation by leveraging graph traversal algorithms. We represent the ontology using a Directed Acyclic Graph (DAG) $G(V, E)$, where V represents the set of named entities and E represents the set of edges between the named entities.

We introduce five attributes for each named entity $v_i (\in V)$:

$$\text{Attribute}(v_i) = \{\text{Label}(v_i), \text{Description}(v_i), \text{IsLeaf}(v_i), \text{Level}(v_i), \text{Present}(v_i)\} \quad (1)$$

where

- $\text{Label}(v_i)$ is the name of the entity,
- $\text{Description}(v_i)$ is a string that represents the prompt message to be used for extracting the named entity v_i from the input text, and
- $\text{IsLeaf}(v_i)$ is a Boolean value that represents if the node is a leaf node (i.e., a node devoid of siblings in the graph).
- $\text{Level}(v_i)$ is an integer that represents the level of the graph model (tree) the node v_i is from and
- $\text{Present}(v_i)$ is a Boolean value that represents if the information associated with v_i is present in the input text.

Each directed edge $e_{ij} \in E$ from the node v_i to the node v_j represents the relation between the two nodes. In the mCODE context, all relations are essentially "has" (or parent-child) relations where a named entity v_i has a named entity v_j as an attribute.

3.2 Formatting and Prompt Completion

The output of the LLM is also free text by default, unless specific instructions about format is provided to constrain the output text. For the prompts generated from the graph G, the following format is applied,

$$\text{\tab Label}(v_i)\text{: <Description}(v_i)\text{> \n}$$

where each named entity v_i form a single line, and all named entities will be concatenated together as multiple lines. The final prompt to be fed into LLM for named entity extraction will be of the following format,

$$\text{prompt}(G)=\text{Leading-Question + Formatted-Prompt + Background-Info + Input-Text}$$

where

- Leading-Question: a concise task description to instruct the LLM of what specific task it needs to perform, e.g., “Please extract the following information from the text.”.
- Formatted-Prompt: a line-by-line prompts generated from graph G in the format of (6).
- Background-Info*:* this is the indication of the beginning of the provided free text, it can be “Here is the text:”.
- Input-Text: the provided free text to extract information from.

3.3 Algorithms

We introduce three distinct algorithms that demonstrate varied strategies for capitalizing on LLMs in conjunction with ontology. Our comparative analysis seeks to highlight their strengths and limitations, thereby providing insight into the optimal conditions for enhancing LLM performance with ontology integration. An illustrative example for each of the three algorithms is provided in Appendix A.

Root-to-Leaf Streamliner (RLS). Although the leaf node information is what is needed by LLM, the named entities on the path provide background information for the leaf node. The RLS strategically harmonizes the information from the root node to each leaf node. This results in a compact, yet informative, prompt that encapsulates the entire lineage of the ontology tree. For each leaf node v_i where IsLeaf(v_i) is true, RLS finds the path from the root node to v_i,

$$root \rightarrow v_a \rightarrow v_b \rightarrow \cdots \rightarrow v_i \tag{2}$$

and generates a prompt for v_i. Given that v_i alone might contain limited information, we incorporate comprehensive background information by including the descriptions of all its parent nodes. For example, consider the following path:

$$cancer\ patient\ (root) \rightarrow tumor\ (v_a) \rightarrow tumor\ size\ (v_b) \rightarrow\ measurement\ method\ (v_i) \quad (3)$$

Instead of Description(v_i) being *"What is the tumor measurement method?"*, RLS will create Description(v_i) as "*If the patient has a tumor, what is the measurement method to measure the tumor size?*". An example RLS prompt is in Figure 2.

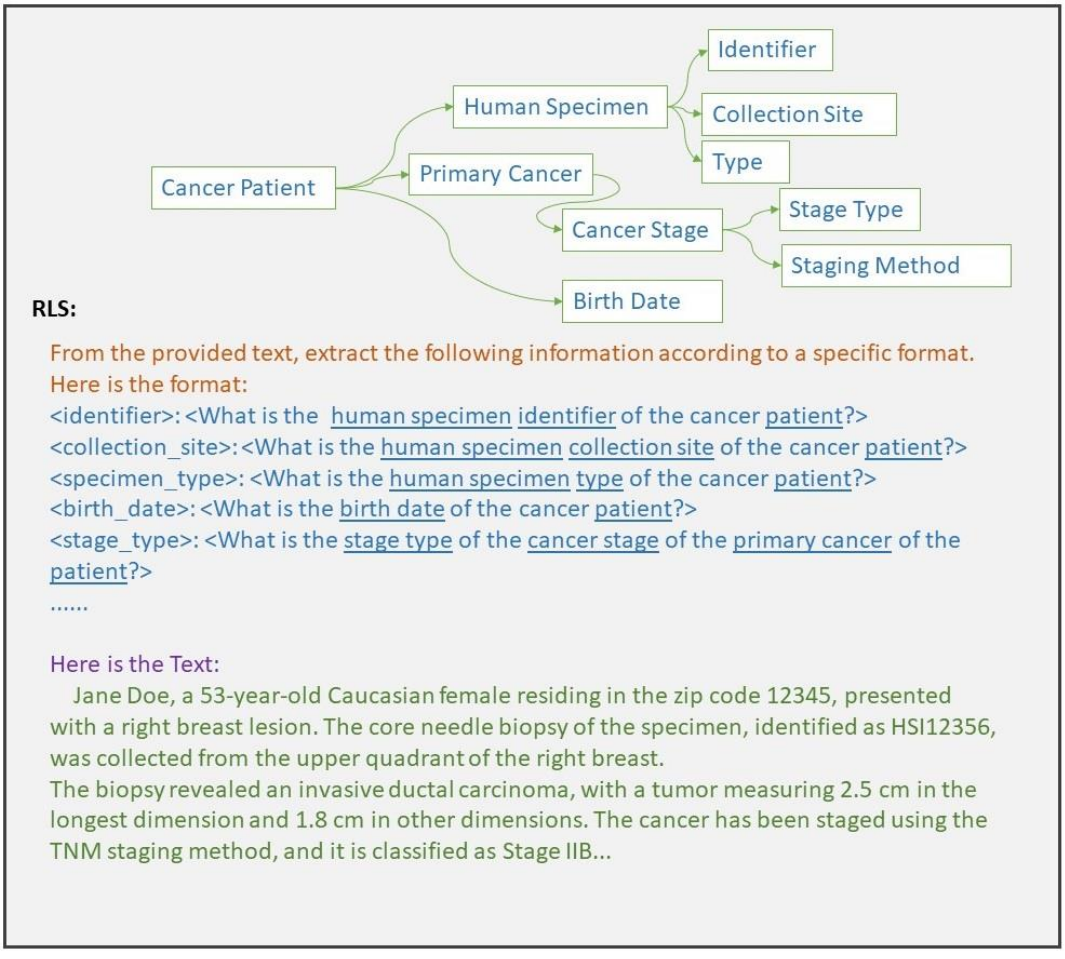


Figure 2. Example of the RLS method. The above is the ontology graph G and below is the prompt generated using Root-to-Leaf Streamliner (RLS). Each leaf named entity generates a query that encorporates all background information from root to this root.

Breadth-First Ontology Pruner (BFOP). In clinical data extraction, it is crucial to recognize that an individual clinical note may not encapsulate all entities present in a comprehensive ontology. Utilizing an extensive prompt for information retrieval can be counterproductive, as it may induce "hallucination", a scenario where models erroneously produce or identify non-existent entities within

the data. These hallucinations often result from models being overly influenced by a detailed ontology, leading to false identifications that could undermine the accuracy of clinical decision-making. Consequently, careful pruning of ontological entities is imperative to ensure precision in the extraction process. BFOP is engineered to pinpoint a relevant subgraph tailored to the designated target text. This approach is inspired by the Chain-of-Thought prompting methodology [25], which utilizes a parsing strategy akin to the breadth-first search algorithm, employing a hierarchical level-wise analysis.

Essentially, BFOP is a layer-wise prompt generation method. It refines the next-level named entity prompts based on whether the information of the named entities from the current level is present. Initially, entities on the first level (the root entity) of the ontology are parsed and transformed into prompts suitable for querying (Algorithm BFOP lines 1-7). Subsequently, these prompts are fed into the LLM to extract pertinent information from patient notes (Algorithm BFOP line 8).

A salient feature of our approach is its dynamic pruning mechanism. If a particular named entity is absent in the patient's record, then it is natural to infer that its children will also be devoid of relevant data, which eliminates the need to parse them (Algorithm BFOP lines 9-12). Therefore, in the next level, $L + 1$, based on the previous level (L) result, some with Present(v_i) = False can be skipped. An example prompt for BFOP algorithm is in Figure 3.

Algorithm BFOP

set Present(v_i) as true for all $v_i \in V$
 # For named entities on a certain level in the ontology tree graph $G(V, E)$
for level $L = 0$ to [height of the tree graph $G(V, E)$]
    for all v_i s.t. Level(v_i) $= L$ and Present(v_i) is true
        if IsLeaf(v_i) is true
            set Description(v_i) = "What is the <Label(v_i)> from the input text?"
        if IsLeaf(v_i) is false
            set Description(v_i)= "Does the text contain any information about <Level(v_i)>? Answer YES or NO."
    call LLM and extract information for all named entities only from level L
    # If information not present, set all child to non-present
    if Present(v_i) is false:
        set Present(v_i) to false

set Present(v_j) to false, for all descendants v_j of v_i

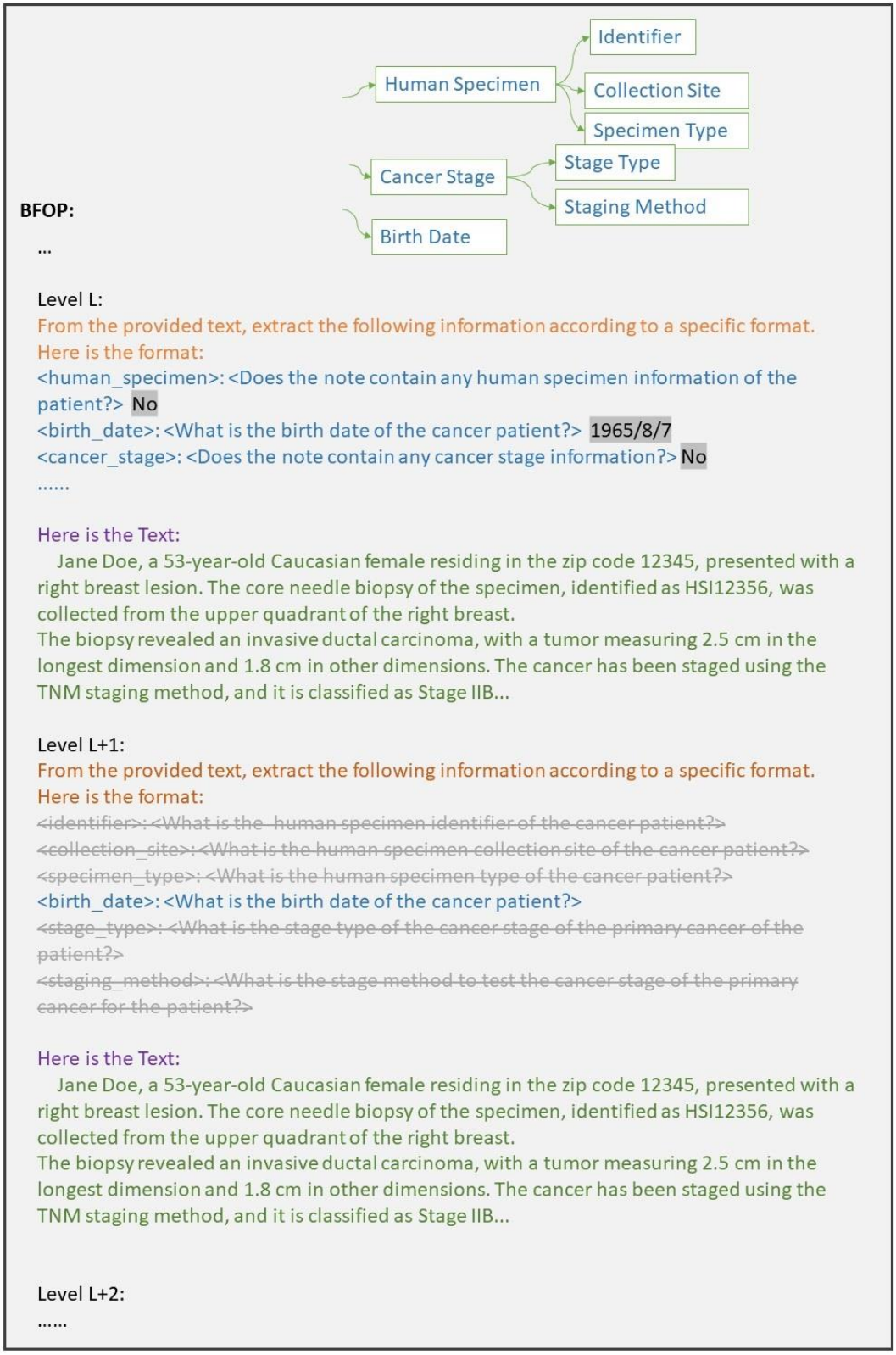
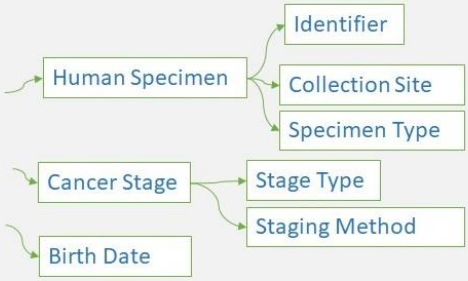


Figure 3. Example of the Breadth-First Ontology Pruner (BFOP) method. After each named entities at level *L* get result (highlighted in gray background), based on these results, the *L+1* level named entities can be trimmed down. Certain queries do not need to be performed as this information are deemed not available from round *L*.

Two-Phase Ontology Parser (2POP). The main idea of the 2POP methodology is that queries about the mere existence of a named entity are simpler to resolve than those seeking details about the entity, particularly in the context of intricate ontologies such as the mCODE. 2POP aims to isolate a pertinent subgraph from the comprehensive ontology that correlates with the target text. Unlike the incremental, hierarchical extraction process characteristic of BFOP, 2POP adopts a comprehensive approach, extracting the entire pertinent structure in one step. The 2POP algorithm employs a two-pronged strategy: First, it queries all leaf nodes, classifying the outcomes based on binary responses. Then, the second phase utilizes only the entities confirmed by a "Yes" response to create a prompt. This prompt is then utilized to prompt an LLM to extract information from unstructured text, thereby enabling the retrieval of relevant data (Algorithm 2POP). An example prompt is in Figure 4.

	Algorithm 2POP
1	set Present(v_i) as true for all leaf nodes $v_i \in V$
2	for all leaf nodes v_i
3	set Description(v_i) = "Does the information of <Label(v_i)> (in the context of the named entities from root to v_i) exist in the text"
4	call LLM and extract information for all named entities leaf nodes v_i
5	set Present(v_i) to true/false based on the LLM result
7	for all leaf nodes v_i
8	set Description(v_i) = "What is the <Label(v_i)> (in the context of the named entities from root to v_i)"
9	call LLM and extract information for all named entities leaf nodes v_i



2POP:

Round 1:

From the provided text, extract the following information according to a specific format.
Here is the format:
<identifier>: <Does the note contain any information about any human specimen identifier information of the cancer patient?> No
<collection_site>: < Does the note contain any information about human specimen collection site of the cancer patient?> No
<specimen_tupe>: < Does the note contain any information about any human specimen type of the cancer patient?> Yes
<birth_date>: < Does the note contain any information about birth date of the cancer patient?> No
<stage_type>: < Does the note contain any information about cancer stage type of the cancer patient?> Yes
<staging_method>: <Does the note contain any information about cancer staging method of the cancer patient?> No

Here is the Text:
Jane Doe, a 53-year-old Caucasian female residing in the zip code 12345, presented with a right breast lesion. The core needle biopsy of the specimen, identified as HSI12356, was collected from the upper quadrant of the right breast.
The biopsy revealed an invasive ductal carcinoma, with a tumor measuring 2.5 cm in the longest dimension and 1.8 cm in other dimensions. The cancer has been staged using the TNM staging method, and it is classified as Stage IIB...

Round 2:

From the provided text, extract the following information according to a specific format.
Here is the format:
~~<identifier>: <Does the note contain any information about any human specimen identifier information of the cancer patient?>~~
~~<collection_site>: < Does the note contain any information about human specimen collection site of the cancer patient?>~~
<specimen_tupe>: <What is the human specimen type of the cancer patient?>
~~<birth_date>: < Does the note contain any information about birth date of the cancer patient?>~~
<stage_type>: < What is the stage type of the cancer stage of the primary cancer of the patient?>
~~<staging_method>: <Does the note contain any information about cancer staging method of the cancer patient?>~~

Here is the Text:
Jane Doe, a 53-year-old Caucasian female residing in the zip code 12345, presented with a right breast lesion. The core needle biopsy of the specimen, identified as HSI12356, was collected from the upper quadrant of the right breast.
The biopsy revealed an invasive ductal carcinoma, with a tumor measuring 2.5 cm in the longest dimension and 1.8 cm in other dimensions. The cancer has been staged using the TNM staging method, and it is classified as Stage IIB...

Figure 4. Example of the Two-Phase Ontology Parser (2POP) method. Similar to the Root-to-Leaf Streamliner, prompts are generated from all leaf named entities. The second-round prompts will be adjusted based on the result (highlighted in gray) of the first round.

The entire mCodeGPT pipeline is shown in Figure 5.

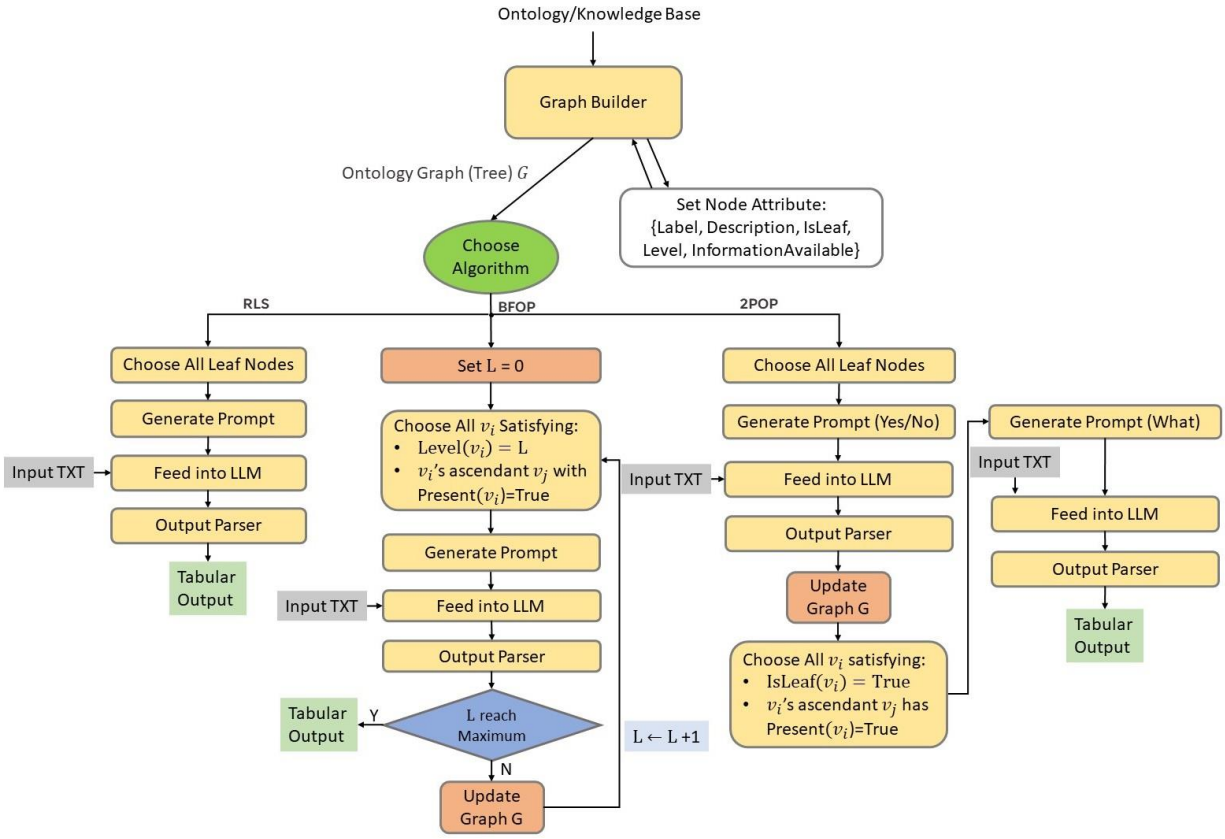


Figure 5. The pipeline of mCodeGPT and explanation of steps.

4 Experiments and Results

4.1 Experiment setting

We evaluate the efficacy of the proposed LLM-based named entity extraction and the effectiveness of each prompt-generating method. The experiments were conducted using high-quality synthetic oncology notes, produced by the advanced generative model GPT3.5-turbo using OpenAI API (Chat Completions API). The model was configured with a temperature setting of 0.8 and a maximum token limit of 4,096 to create a dataset of simulated clinical notes. These notes encompassed a diverse range of cancer types, with the subject of each note being randomly selected from a predetermined list of cancer categories. Table 1 specifies the Cancer types in this experiment. A total of 400 cancer reports

(clinical notes) were generated, each focusing on the characteristics and details specific to the cancer type it represents. For each note, a random cancer type was chosen from **Table 1** to generate biologically meaningful descriptions.

Table 1. Types of Cancers

Category	Types of Cancer
Skin Cancers	Basal Cell Carcinoma, Squamous Cell Carcinoma, Melanoma
Kidney Cancers	Renal Cell Carcinoma
Breast Cancers	Ductal Carcinoma In Situ, Invasive Ductal Carcinoma, Adenocarcinoma
Soft Tissue Sarcomas	Osteosarcoma, Liposarcoma, Rhabdomyosarcoma
Blood Cancers	Leukemia, Acute Lymphoblastic Leukemia, Acute Myeloid Leukemia, Chronic Lymphocytic Leukemia, Chronic Myeloid Leukemia
Lymphomas	Non-Hodgkin Lymphoma, Hodgkin Lymphoma
Brain and Central Nervous System	Glioblastoma, Astrocytoma, Meningioma, Medulloblastoma
Gynecologic Cancers	Cervical Cancer, Ovarian Cancer, Uterine Cancer, Vaginal Cancer, Vulvar Cancer
Respiratory Cancers	Lung Cancer, Small Cell Lung Cancer, Non-Small Cell Lung Cancer
Genitourinary Cancers	Prostate Cancer, Bladder Cancer
Gastrointestinal Cancers	Colon and Rectal Cancer, Pancreatic Cancer, Stomach (Gastric) Cancer, Esophageal Cancer
Endocrine Cancers	Thyroid Cancer
Other Organ-Specific Cancers	Liver Cancer, Testicular Cancer, Nasopharyngeal Cancer, Laryngeal and Hypopharyngeal Cancer, Penile Cancer, Oral and Oropharyngeal Cancer
Pediatric Cancers	Neuroblastoma, Wilms Tumor, Retinoblastoma

We examined the effectiveness of various models, including open-source models and OpenAI models, along with the proposed methods, RLS, BFOP, and 2POP. The open-source models include four from GPT4ALL [26], namely Orca-mini-13b, Nous-hermes-13b, Llama-2-7b-chat, WizardLM-13b-uncensored, and one from FastChat [27], the Fastchat-vicuna-13b-1.5. The experiments were performed

on a local server with an AMD EPYC 7742 64-core Processor. Optionally, GPU resources can be used if available, which can reduce the running time for queries.

4.2 Results

4.2.1 Performance

This section evaluates the performance of mCodeGPT on named entity extraction tasks from the mCODE ontology using different LLM as backbones combined with different prompt generation methods. We apply the following rules when counting correct, error, and hallucinated named entities.

- **Correct**: The information extraction result (IE) is the same as the ground truth (GT), or semantically equivalent; for example, GT is “Female” but IE is “female”; GT is “Normal” but IE is “within the Norm range”; GT is “15-Jan-75” but IE is “Jan 15, 1975”
- **Error**: The information extraction result is inconsistent with ground truth. This includes two cases:
  - **Misidentification:** The named entity has information in the notes (GT) but the information extraction result is none (“N/A”, “non”, “Not mentioned”, “N/A (not mentioned in the text)”, etc.
  - **Misplacement:** The named entity has information in the notes, and the information extraction result is not correct, but from the notes itself (not hallucinated).
- **Hallucination**: Information extraction results in hallucinated information (not from the original notes), i.e., information that is fabricated or generated and is not present in the original text.

Table 2 represents the accuracy of each model on the 400 clinical notes. With respect to the incorrect results, we also identified the incorrect incidence rate and hallucination rate. The table shows our pipeline generally performed better with the 2POP method, consistently having the highest

accuracy. This aligns with the initial design intent of the method, to perform a first-round screening (ask presence) before prompting more difficult questions (ask details).

Among the open-source models, Orca-mini-13b achieved the highest accuracy with 2POP at 0.937, closely followed by Llama-2-7b-chat at 0.925. Among all the models, the OpenAI model GPT3.5-turbo-16k outperformed its counterparts, achieving an accuracy of 0.956 using 2POP. By contrast, RLS tended to achieve the lowest accuracy across most models. This finding aligns well with the nature of the RLS method, which issues queries for all named entities straightforwardly, without leveraging hierarchical structuring or an easy-to-difficult design approach.

Hallucinations happened quite rarely (< 2.0%) for both open-source models and Open-AI API-based models. In general, BFOP and 2POP slightly reduced hallucination compared with RLS, which can be explained by the designed chain-of-thought queries. Overall, the OpenAI models hallucinated less than the open-source models.

Table 2. Evaluation of Information Extraction Efficacy. This table presents the assessment of information extraction performance using different model-method pairings, quantified through accuracy, error rate, and hallucination rate, which collectively sum to unity. The analysis was performed on a corpus of 400 notes. The bold fonts represent the best performance entry in each column.

	Accuracy			Error Rate			Hallucination Rate		
	RLS	BFOP	2POP	RLS	BFOP	2POP	RLS	BFOP	2POP
Open-Source Models									
Orca-mini-13b	0.792	0.887	0.937	0.194	0.100	0.049	**0.014**	0.013	0.014
Nous-hermes-13b	**0.843**	0.836	0.920	**0.142**	0.154	0.067	0.015	**0.010**	0.013
Llama-2-7b-chat	0.840	**0.900**	0.925	0.146	**0.087**	0.063	**0.014**	0.013	0.012
Fastchat-vicuna-13b-1.5	0.787	0.867	0.902	0.194	0.116	0.088	0.019	0.017	**0.010**

WizardLM-13b-uncensored	0.831	0.865	**0.940**	0.153	0.120	**0.048**	0.016	0.015	0.012
OpenAI Models									
GPT3.5-turbo-16k	0.865	0.952	0.956	0.120	0.038	0.034	0.015	0.010	0.010
GPT3.5-turbo	0.868	0.926	0.934	0.118	0.065	0.057	0.014	**0.009**	0.009
GPT4	**0.879**	**0.932**	0.946	**0.108**	**0.056**	0.045	0.013	0.012	0.009
GPT4-32k	0.870	0.930	**0.950**	0.118	0.059	**0.042**	0.012	0.011	**0.008**

4.2.2 Running Time

The running times varied across models depending on the method employed, which provides insights into the efficiency and directly impacts the practical utility and feasibility of each model-method combination.

Figure 6 shows the running time for each model concerning each prompting method. In general, open-source models took longer to process depending on the local computing resources. WizardLM-13b-uncensored, exhibited considerably longer running times. This model required 80 hours with the RLS method to run on 400 notes. By contrast, the OpenAI models demonstrated considerably faster running times. The GPT3.5-turbo model emerged as one of the most efficient, with running times under 1.5 hours across all methods. It is worth noting that the maximum token per minute (TPM) of Azure OpenAI models differs by what pricing tier is being chosen, which can affect the frequency of calling OpenAI APIs and, therefore affect the OpenAI models' performance using our pipeline. For example, under the pricing tier s0, our Azure OpenAI limits up to 240 TPM (GPT3.5-turbo, GPT3.5-turbo-16k), 20 TPM (GPT4), and 60 TPM (GPT4-32k), respectively.

It should be noted that the extended processing time does not necessarily imply ineffectiveness. All models exhibited only around a 5% accuracy drop compared to OpenAI's advanced models (GPT4, GPT4-32k) – and, more importantly, they do not require any overhead for token consumption.

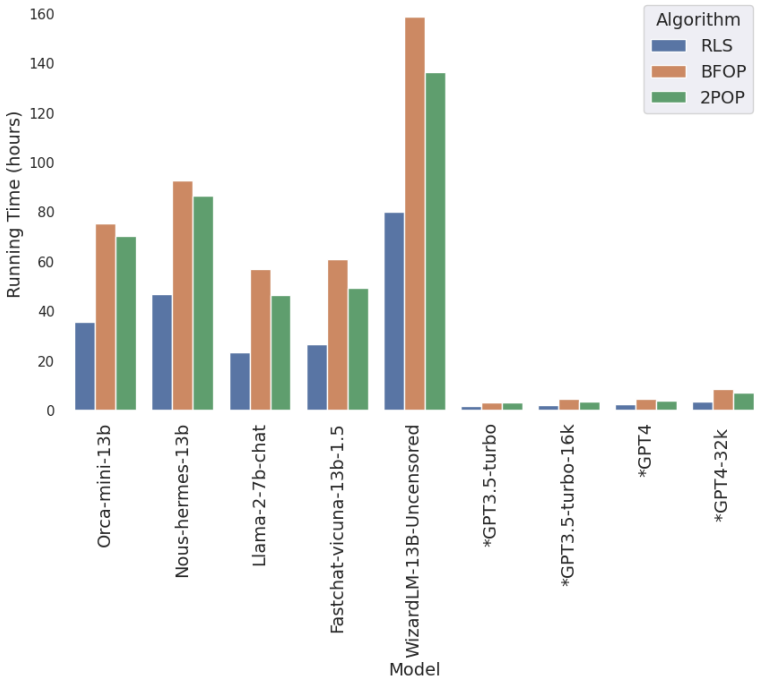


Figure 6. The running time for each model when using different algorithms.

4.2.3 Performance on Medical Abbreviations

In practice, physicians often use abbreviations in their notes. To assess our model's performance in more realistic situations, we modified the clinical notes generation process and let the clinical notes embed as many abbreviations, drawn from various medical terminologies, as possible. The abbreviations are provided in Appendix B.

Table 3. Comparative Performance Analysis of Notes with High Utilization of Medical Terminology Abbreviations Versus Notes Without Abbreviations.

	Accuracy		
	RLS	BFOP	2POP
Open-Source Models			
Orca-mini-13b	$0.6589_{\downarrow 0.1334}$	$0.6807_{\downarrow 0.2064}$	$0.7051_{\downarrow 0.2318}$
Nous-hermes-13b	$0.7254_{\downarrow 0.1178}$	$0.7262_{\downarrow 0.1102}$	$0.7601_{\downarrow 0.1602}$
Llama-2-7b-chat	$0.7191_{\downarrow 0.1208}$	$0.7441_{\downarrow 0.1562}$	$0.7600_{\downarrow 0.1651}$

Fastchat-vicuna-13b-1.5	$0.6508_{\downarrow 0.1366}$	$0.6984_{\downarrow 0.1726}$	$0.7290_{\downarrow 0.1731}$
WizardLM-13b-uncensored	$0.6934_{\downarrow 0.1378}$	$0.7312_{\downarrow 0.1337}$	$0.7333_{\downarrow 0.2070}$
OpenAI Models			
GPT3.5-turbo-16k	$0.7314_{\downarrow 0.1331}$	$0.7451_{\downarrow 0.2070}$	$0.7501_{\downarrow 0.2061}$
GPT3.5-turbo	$0.7266_{\downarrow 0.1411}$	$0.7199_{\downarrow 0.2065}$	$0.7334_{\downarrow 0.2010}$
GPT4	$0.7452_{\downarrow 0.1340}$	$0.7564_{\downarrow 0.1757}$	$0.7795_{\downarrow 0.1660}$
GPT4-32k	$0.7460_{\downarrow 0.1243}$	$0.7614_{\downarrow 0.1686}$	$0.7571_{\downarrow 0.1923}$

Table 3 summarizes the performance of the different models and prompting methods in terms of accuracy. GPT-4 demonstrated superior proficiency in interpreting medical terminologies and abbreviations. For all models and methods, the performance dropped by 11% to 20% when compared to results based on notes without abbreviations. The decline in performance does not diminish the utility of the proposed tool, since medical abbreviations are usually limited and can easily be summarized. It is also worth noting that the table shows the worst-case performance because all medical terms were replaced by their abbreviations in the 400 notes, which is unlikely to occur in reality. A full list of common abbreviations of medical concepts can be found in **Table 4**.

Table 4. List of Common abbreviations in medical notes.

Medical terms: abbreviations				
Activating: Act	Death: Dth	Leukemia: Leuk	Protein: Prot	system: sys
Administration: Admin	Description: Desc	Level: Lvl	Radiotherapy: RT	System: Sys
Allelic: Allel	Diabetes: DM	Location: Loc	Ranges: Rngs	Technique: Tech
Amino: Aa	Dimension: Dim	Lymphoma: Lympho	Reason: Rsn	Transplant: Transpl
applicable: App	Disease: Dz	Marker: Mrk	Reference: Ref	treatment: Tx
Attribute: Attr	Ethnicity: Ethn	Medication: Med	Region: Reg	Tumor: Tum
Binet: Binet	Evidence: Evd	Melanoma: Mel	related: rel	Type: Typ

Breslow: Breslow	Examined: Exam	Metastatic: Mets	Related: Rel	Value: Val
Cancer: CA	fetoprotein: fetoprotein	Method: Meth	Report: Rpt	Variant: Var
Caucasian: Cau	Frequency: Freq	Missense: Miss	Request: Req	Variation: Var
Center: Ctr	Gender: Gen	Modality: Mod	Result: Rslt	Volume: Vol
Change: Chg	Genomic: Gen	Molecular: Mol	Secondary: Sec	
Class: Cls	Gleason: Gleason	Morphology: Morph	Sequence: Seq	
Code: Code	Gynecologic: Gyn	mutation: mut	Sessions: Sess	
Coding: Cng	Hepatocellular: HepatoCA	Mutations: Mut	Somatic: Som	
Collection: Coll	Heterozygous: HetZ	Myeloma: Myelo	Sorafenib: Sorafenib	
Comorbidities: Comorb	HGVS: HGVS	Nomenclature: Nomen	Source: Src	
Conditions: Conds	Hispanic: Hisp	Normal: Norm	Specimen: Spec	
Consequence: Conseq	Histology: Hist	Number: Num	Stage: Stg	
Contact: Cont	History: Hist	Palliative: Pall	Staging: Stg	
control: ctrl	Hypertension: HTN	Patient: Pt	State: St	
Coordinate: Coord	Identifier: ID	preference: pref	Status: Stat	
Copy: Cpy	Imaging: Img	Primary: Prim	Studied: Stud	
Cytogenetic: Cytogen	Information: Info	Procedure: Proc	Substitution: Subst	
Date: Dt	Intent: Intent	Prostate: Prostate	Summary: Summ	

4.2.4 Heterogenous Difficulty of Named Entities

The performance varied across various named entities within specific information extraction tasks. Figure 7 illustrates the evaluation of named entity recognition performance within the dataset comprising 400 documents. Each bar in the graph represents the accuracy with which the system identifies each named entity, with a filled bar denoting flawless recognition across the entire corpus. The data indicates a perfect extraction for 65% of the named entities, while the accuracy for the remaining 35% varied between 25% and 99%.

The named entities that achieved perfect performance primarily consisted of terms related to medical terms, genomic variants, and cancer-related information. These terms are technical and likely contain domain-specific abbreviations. By contrast, the imperfect (<100% accurate on all 400 notes) named entities comprise terms pertinent to patient details, such as demographics, procedures, and medication administration. These terms are more focused on patient-related data and administrative dimensions.

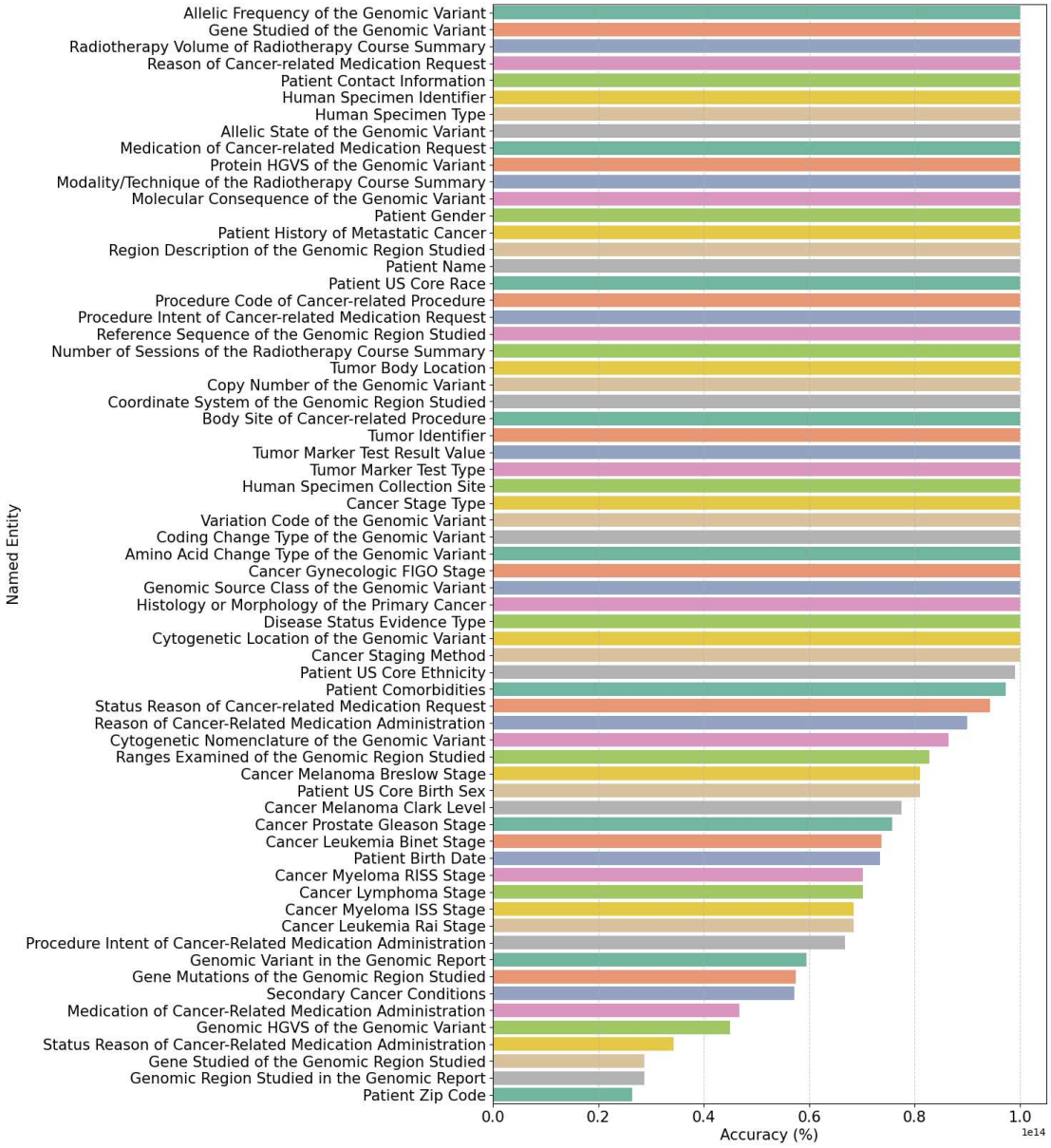


Figure 7. Accuracy for each named entity on the 400 notes using model GPT3.5.

5 Discussion

Our findings offer valuable insight into the possibilities and power of LLMs, especially when used with clinical ontologies. Beyond its direct application to cancer ontologies, our methodologies can be extended and applied to various other medical specialties and healthcare domains.

Despite achieving high accuracy, it is essential to have human oversight to ensure the fidelity of the results. There were instances where not every named entity was accurately identified. Thus, a review of the outcomes is necessary to prevent any oversight or misidentification.

To promote in-context learning and increase mCodeGPT's performance when handling abbreviations, a list of abbreviations can be injected into the prompt as background knowledge -- used as a one-shot exemplar. Earlier studies have shown that one-shot in-context learning [28], by presenting in the prompt an example of the task it is asked to do, can significantly boost the performance of LLMs.

There are several potential areas for future research for further refinement and adaptation of mCodeGPT. First, multimodal integration – with the evolving capability of OpenAI GPT models to incorporate multimodality, mCodeGPT can be enhanced to leverage both textual and image-based information for NER tasks, especially in the fields of radiology and pathology. Second, the automated generation of ontologies becomes crucial for expanding into domains where ontological frameworks are not well-established. Parsing clinical notes and autonomously creating ontologies for NER is of paramount importance in such scenarios.

6 Conclusions

Our mCodeGPT architecture represents a significant advancement in cancer research and clinical informatics, introducing innovative methods for zero-shot extraction of structured data from complex clinical narratives. This transformation of unstructured information into actionable insights has the potential to fundamentally alter our comprehension and approach to cancer treatment and research. By streamlining the process of NER, mCodeGPT can substantially reduce the time and labor traditionally required for data extraction and standardization. This increased efficiency allows healthcare

professionals to allocate more time to direct patient care and clinical assessments. Consequently, the reduced administrative burden may accelerate diagnostic procedures and the initiation of treatment, which could lead to enhanced patient outcomes.

Reference

1. *Cancer research: past, present and futureCancer research: past, present and future*.
2. Chin, L., Andersen, J. N. & Futreal, P. A. Cancer genomics: from discovery science to personalized medicine. *Nat. Med.* **17**, 297–303 (2011).
3. Jensen, P. B., Jensen, L. J. & Brunak, S. Mining electronic health records: towards better research applications and clinical care. *Nat. Rev. Genet.* **13**, 395–405 (2012).
4. Li, I. *et al.* Neural Natural Language Processing for unstructured data in electronic health records: A review. *Computer Science Review* **46**, 100511 (2022).
5. Consultant, H. I. T. Why unstructured data holds the key to intelligent healthcare systems [Internet]. *Atlanta (GA): HIT Consultant* (2015).
6. Kowalczyk, S. & Shankar, K. Data sharing in the sciences. *Annual review of information science and technology* **45**, 247–294 (2011).
7. Warner, J. L. *et al.* Development, implementation, and initial evaluation of a foundational open interoperability standard for oncology treatment planning and summarization. *J. Am. Med. Inform. Assoc.* **22**, 577–586 (2015).
8. Parsing, C. Speech and language processing. *Power Point Slides* (2009).
9. Sarawagi, S. Information Extraction. *Foundations and Trends® in Databases* **1**, 261–377 (2008).
10. Caufield, J. H. *et al.* Structured prompt interrogation and recursive extraction of semantics (SPIRES): A method for populating knowledge bases using zero-shot learning. *arXiv [cs.AI]* (2023).
11. Bird, S. NLTK: The natural language toolkit. https://aclanthology.org/P06-4018.pdf (2006).

12. Manning, C. *et al.* The Stanford CoreNLP natural language processing toolkit. in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (Association for Computational Linguistics, 2014). doi:10.3115/v1/p14-5010.
13. Chang, J. T., Schütze, H. & Altman, R. B. GAPSCORE: finding gene and protein names one word at a time. *Bioinformatics* **20**, 216–225 (2004).
14. Liu, H., Lussier, Y. A. & Friedman, C. Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method. *J. Biomed. Inform.* **34**, 249–261 (2001).
15. Bodenreider, O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.* **32**, 267D – 270 (2004).
16. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv [cs.CL]* (2018).
17. Liu, Y. *et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv [cs.CL]* (2019).
18. Wu, Y., Xu, J., Zhang, Y. & Xu, H. Clinical Abbreviation Disambiguation Using Neural Word Embeddings. in *Proceedings of BioNLP 15* 171–176 (Association for Computational Linguistics, 2015).
19. Wu, Y., Xu, J., Jiang, M., Zhang, Y. & Xu, H. A study of neural word embeddings for Named Entity Recognition in clinical text. *AMIA Annu. Symp. Proc.* **2015**, 1326–1333 (2015).
20. Wei, J. & Zou, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv [cs.CL]* (2019).
21. Sennrich, R., Haddow, B. & Birch, A. Improving neural machine translation models with monolingual data. in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, 2016). doi:10.18653/v1/p16-1009.

22. Hassanpour, S. & Langlotz, C. P. Information extraction from multi-institutional radiology reports. *Artif. Intell. Med.* **66**, 29–39 (2016).

23. Kleihues, P., Cavenee, W. K. & Others. WHO classification of tumours. *Pathology & genetics. Tumors of the nervous system. Lyon, France: IARCpress* (2000).

24. Stratton, M. R., Campbell, P. J. & Futreal, P. A. The cancer genome. *Nature* **458**, 719–724 (2009).

25. Wei, J. *et al.* Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **35**, 24824–24837 (2022).

26. Anand, Y., Nussbaum, Z., Duderstadt, B. & Schmidt, B. GPT4All: Training an assistant-style chatbot with large scale data distillation from GPT-3.5-turbo. http://static.nomic.ai.s3.amazonaws.com/gpt4all/2023_GPT4All_Technical_Report.pdf (2023).

27. Zheng, L. *et al.* Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv [cs.CL]* (2023).

28. Brown, T. *et al.* Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).