FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

DISSERTATION

DALY: Continuous Cancer Care Assistance via Large Language Models

Author:
Ana Guedes

Supervisor:
Professor Nuno Rodrigues
Professor António Coelho

Master in Biomedical Engineering

July 31, 2025



"I think we all change each other's paths. I don't know which law idea that is in physics, but I don't think any of us can live without affecting one another."

Frank Ocean

Resumo

Vários tratamentos contra o cancro estão associados a uma vasta gama de efeitos secundários que devem ser sistematicamente monitorizados durante o tratamento. Muitos efeitos secundários baseiam-se em relatos de doentes, que são manifestações subjetivas e dependem da descrição dos sintomas feita pelo doente. A realização de entrevistas estruturadas sobre os sintomas para recolher todas as informações relevantes para a classificação de acordo com os CTCAE exige muitas vezes tempo e esforço significativos por parte dos profissionais de saúde.

A dissertação discute a criação de um assistente clínico automatizado para facilitar este processo através da combinação de grandes modelos de linguagem e mecanismos baseados em grafos.

Esta estrutura híbrida permite que o sistema efetue uma entrevista estruturada, faça ajustes em tempo real à conversa e recolha todos os dados importantes para classificar o sintoma.

Uma vez concluída a entrevista, o sistema verifica automaticamente os graus atribuídos e gera resumos médicos com pontos-chave para análise.

A plataforma foi implementada de modo a que a totalidade da execução fosse local e possível sem acessos externos à API ou transferências de dados.

Uma validação clínica preliminar com um oncologista mostrou concordância total com o sistema no que diz respeito ao reconhecimento dos sintomas e à atribuição dos graus CTCAE.

A avaliação da carga de trabalho realizada com o instrumento NASA-TLX demonstrou uma carga mental baixa em ambas as interfaces, de 22.0 e 4.8, respetivamente, nas interfaces de revisão do paciente e do médico.

Os resultados da usabilidade também foram positivos, com pontuações de 87.5 e 100.0 no SUS na interface do paciente e na interface médica, respetivamente, e pontuações elevadas em todas as dimensões do QUIS (7.88 para a interface do doente e 8.78 para a interface médica).

Abstract

Several cancer treatments are associated with a broad range of side effects which must be monitored systematically during treatment. Numerous side effects are based on patient reports, and these are subjective manifestations and depend on the patient's description of the symptoms. Conducting structured symptom interviews to collect all grading relevant information according to CTCAE often requires significant time and effort from healthcare professionals.

The dissertation discusses the creation of an automated clinical assistant that would facilitate this process through the combination of large language models and graph-based mechanisms.

This hybrid framework enables the system to carry out a structured interview, make real-time adjustments to the conversation and collect all of the important data to grade the symptom.

With its completion, the system automatically checks the assigned grades and generates medical summaries with key bullet points for physician review.

The platform was deployed so that the entirety of local execution was possible with no external API accesses or data transfers.

A preliminary clinical validation with an oncologist showed that the clinician completely agreed with the system in respect to symptom recognition and assignment of CTCAE grades.

The workload evaluation carried out with the NASA-TLX instrument has demonstrated a low mental load of both interfaces of 22.0 and 4.8, respectively, in patient-facing and medical review interfaces. The usability outcomes were also positive, being 87.5 and 100.0 on SUS scores on the patient and medical interface respectively, and high scores on all QUIS dimensions (7.88 for the patient interface and 8.78 for the medical interface).

Agradecimentos

Primeiro de tudo, gostava de agradecer aos meus queridos pais pelo apoio, amor incondicional e coragem. À minha mãe um exemplo de mulher, mãe e, sobretudo, de pessoa, que me mostrou como ser e como agir. Ao meu pai, cujos valores são inquestionáveis, por todos os conselhos.

Gostava de agradecer ao meu orientador, o Professor Nuno Rodrigues, por toda a disponibilidade e por me ter guiado durante toda a dissertação.

Quero agradecer também aos meus avozinhos, por todo o amor que me deram.

À minha família por me apoiarem e por acreditarem sempre em mim.

À Carol, Sara, Pires e Rebhan, por tornarem a vida melhor.

Ao Sydney por me pôr sempre um sorriso na cara, cada elemento é uma parte de mim.

À minha Matilde, que está sempre presente.

Ao Rodrigo, que consegue sempre tornar o meu dia mais leve.

Por último, e igualmente importante, quero agradecer ao Jorge, que me acompanha e irá acompanhar para sempre.

Contents

1 Introduction **1**
1.1 Problem Statement . . . 2
1.1.1 Research Question . . . 3
1.1.2 Hypothesis . . . 3
1.2 Purpose . . . 3
1.3 Objectives . . . 4
1.4 Document Structure . . . 4
1.5 Contribution to the United Nations Sustainable Development Goals . . . 4

2 Background **7**
2.1 Symptom monitoring in cancer . . . 7
2.2 Common Terminology Criteria for Adverse Events . . . 8
2.2.1 Overview . . . 8
2.2.2 Development and Structure of CTCAE . . . 8
2.2.3 Alignment with Patient Reported Outcome . . . 9
2.3 Large Language Models . . . 10
2.3.1 Fine-Tuning . . . 11
2.3.2 Prompting . . . 13
2.3.3 Reinforcement Learning from Human Feedback . . . 14
2.3.4 Retrieval-Augmented Generation . . . 15
2.3.5 Context Management . . . 17
2.4 Graph Theory Foundations for Dialogue Systems . . . 18
2.4.1 Formal Definition of Graphs . . . 18
2.4.2 Directed Acyclic Graphs . . . 19
2.5 Task-Oriented Chatbots . . . 20
2.5.1 Traditional Task-Oriented Chatbot Architectures . . . 20
2.5.2 LLM-Driven Task-Oriented Chatbots Architectures . . . 20

3 Literature Review **27**
3.1 Methods . . . 27
3.1.1 Eligibility Criteria . . . 27
3.1.2 Selection of Sources and Screening . . . 29
3.2 Results . . . 29
3.2.1 Types of Digital Tools Developed . . . 32
3.2.2 Symptom Monitoring According to CTCAE . . . 33
3.2.3 Technology and Frameworks Used . . . 33
3.2.4 Outcomes . . . 34
3.3 Discussion . . . 36
3.3.1 Clinical Impact and Technology Type . . . 36
3.3.2 Usability and Patient-Centered Design . . . 37
3.3.3 Standardisation and Flexibility in Measurement of Symptoms . . . 37

3.3.4 Alerts and Workflow Integration . . . 37
3.3.5 Gaps and Research Opportunities . . . 37
3.3.6 Future Directions . . . 38
3.3.7 Conclusion . . . 39

4 Developing a Clinical Chatbot for Structured Symptom Reporting in Oncology 40
4.1 Part I: System Design . . . 41
4.1.1 Overview . . . 41
4.1.2 Symptom Modeling . . . 42
4.1.3 Dialogue Execution: Node Controller, Question Generator, and Response Assessor . . . 45
4.1.4 Symptom Identifier . . . 47
4.1.5 Node Summarization . . . 48
4.1.6 Grade Assigner . . . 48
4.1.7 Grade Validator . . . 49
4.1.8 Medical Report . . . 50
4.2 Part II: Technical Implementation . . . 51
4.2.1 Overview and Execution Pipeline . . . 51
4.2.2 Node Instantiation . . . 52
4.2.3 Symptom Identifier Pipeline: Embeddings and LLM Verification . . . 53
4.2.4 Evaluation of Symptom Retrieval System . . . 54
4.2.5 LLM Integration . . . 55
4.2.6 Deployment and Runtime Environment . . . 57
4.2.7 Database Architecture and Data Persistence . . . 59
4.2.8 Data Protection, Security and Clinical Safety . . . 60
4.2.9 User Interfaces and Interaction Design . . . 61

5 Results and Discussion 71
5.1 Evaluation Methods . . . 71
5.2 Evaluation Results . . . 73
5.2.1 Physician Interaction: Patient Interface . . . 74
5.2.2 Physician Interaction: Medical Interface . . . 76

6 Conclusions and Future Work 79

A Evaluation Instruments: Full Question Set 99
A.1 Patient Review Interface Evaluation . . . 99
A.2 Medical Review Interface Evaluation . . . 100
A.3 Usability Evaluation (SUS and QUIS) . . . 100

B Medical Report 102

List of Figures

2.1 The proposed SGP-TOD architecture is depicted with a dialog example [1]. . . . 21
2.2 The framework of DiagGPT [2]. . . . 22
2.3 TOD-Flow architecture [3]. . . . 22
2.4 General architecture of the Script-Based AI Therapist [4]. . . . 24

3.1 PRISMA 2020 flow diagram of the study selection process. . . . 29

4.1 Structured Representation of the symptom Fatigue According to CTCAE Criteria. 44
4.2 Symptom graph traversal for fatigue. . . . 47
4.3 Relational database schema. . . . 60
4.4 Log in screen. . . . 62
4.5 Password recovery process. . . . 63
4.6 Patient interface: symptom reporting session. . . . 64
4.7 Patient interface: session history. . . . 65
4.8 Medical interface. . . . 66
4.9 Notification bell. . . . 67
4.10 Editing the grade and justification provider. . . . 67
4.11 Bar and line charts. . . . 68
4.12 Copy to clipboard functionality. . . . 69
4.13 Administrator panel. . . . 69
4.14 Patient's management interface. . . . 70

List of Tables

1.1 Contribution to SDGs and Performance Metrics 5

2.1 Comparison of modern Large Language Model (LLM)-driven task-oriented dialogue architectures. 24

3.1 Inclusion criteria. 28
3.2 Digital Tools Using CTCAE or PRO-CTCAE: Instruments and Clinically Relevant Outcomes 30

4.1 Accuracy comparison across retrieval strategies 55
4.2 Main functional components within the application container 58

5.1 Grades assigned by the system and confirmed by the oncologist 74
5.2 NASA-TLX scores: Patient Interface 74
5.3 Qualitative 5-point scores: Patient Interface 75
5.4 SUS scores: Patient Interface 75
5.5 QUIS scores: Patient Interface 76
5.6 NASA-TLX scores: Medical Interface 76
5.7 Qualitative 5-point scores: Medical Interface 77
5.8 SUS scores: Medical Interface 77
5.9 QUIS scores: Medical Interface 77

List of Abbreviations

AE	Adverse Event
AEs	Adverse Events
AERS	Adverse Event Reporting System
AI	Artificial Intelligence
ADL	Activities of Daily Living
app	application
API	Application Programming Interface
CSS	Cascading Style Sheets
CeHRes	Centre for eHealth Research Roadmap
CTCAE	Common Terminology Criteria for Adverse Events
CoT	Chain of Thought
DAG	Directed Acyclic Graph
DST	Dialogue State Tracking
EORTC	European Organisation for Research and Treatment of Cancer
ED	Emergency Department
EHR	Electronic Health Records
EMR	Electronic Medical Records
eHealth	electronic Health
ePROs	electronic Patient-Reported Outcomes
FACT	Functional Assessment of Cancer Therapy
FSM	Finite State Machines
GAD	Generalized Anxiety Disorder
GDPR	General Data Protection Regulation
GPT	Generative Pre-Trained Transformers
HL7	Health Level 7
HTML	Hypertext Markup Language
HRQoL	Health-Related Quality of Life
IADLs	Instrumental Activities of Daily Living
iOS	iPhone Operating System

IT	Information Technology
KGs	Knowledge Graphs
LLMs	Large Language Models
LLM	Large Language Model
LSTM	Long Short-Term Memory
LoRA	Low-Rank Adaptation
SOC	System Organ Class
MedDRA	Medical Dictionary for Regulatory Activities
mHealth	mobile Health
NASA	National Aeronautics and Space Administration
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NCI	National Cancer Institute
PAM	Patient Activation Measure
PHI	Personal health information
PHP	Personal Home Page
PRISMA-ScR	Preferred Reporting Items for Systematic Reviews and Meta-Analyses Extension for Scoping Reviews
PRO	Patient-Reported Outcome
PROs	Patient-Reported Outcomes
QoL	Quality of Life
QLQ-C30	core quality of life questionnaire
QUIS	Questionnaire for User Interaction Satisfaction.
QR	Quick Response
RAG	Retrieval-Augmented Generation
RLHF	Reinforcement Learning from Human Feedback
RNN	Recurrent Neural Networks
SDGs	Sustainable Development Goals
SMS	Short Message Service
SQL	Structured Query Language
SUS	System Usability Scale
TLX	Task Load Index
TOD	Task-Oriented Dialog

Chapter 1

Introduction

In 2022, 9.7 million people died from cancer, according to the World Health Organization, remaining one of the leading causes of death globally [5]. In addition of its high mortality rates, cancer also greatly impacts patients' daily lives both by the cancer itself and side effects of treatments [6, 7, 8].

Monitoring the symptoms significantly helps in getting better treatment outcomes and quality of life for cancer patients. Proactive symptom monitoring has been demonstrated to decrease emergency department visits, prolong treatment durations, and improve survival rates [9, 10].

The Common Terminology Criteria for Adverse Events (CTCAE) is a standardized classification and severity grading scale developed by the National Cancer Institute (NCI) for classification and reporting of Adverse Events (AEs) [11, 12, 13]. This global system is used in clinical practice and research to characterize and assess the severity and potential impact of adverse events linked to cancer therapies [11, 12].

The integration of Artificial Intelligence (AI), particularly through advancements in Large Language Models (LLMs) has great potential to transform the patient-doctor relationship [14]. LLMs are at the front of AI technologies that have an extraordinary ability to understand, process, and generate human-like text [14, 15, 16]. The alignment of AI approaches with the CTCAE has the potential to considerably improve the collection of symptoms. This alignment could allow systematic grading and tracking of adverse effects according to standardized criteria, thereby ensuring the consistency and comparability of symptom reporting over different phases and forms of treatment.

1.1 Problem Statement

Effective symptom management in oncology is often compromised by the underreporting of adverse symptoms and side effects by clinicians [17, 18, 19]. This underreporting may also result in inappropriate care approaches that lower the quality of life of the patient and increase the burden of the disease [20, 21].

In addition, traditional care approaches commonly react late to serious symptoms, addressing them when they have already caused a great deal of distress and require costly interventions [22, 23]. To fill this gap Patient-Reported Outcomes (PROs) have been adopted [19, 24]. PROs are direct reports from patients about their health status, symptoms, quality of life, or treatment response that rely on the internalization of health or treatment data without interpretation by people other than the patient. Often obtained via validated tools, such as the Patient-Reported Outcomes version of the Common Terminology Criteria for Adverse Events (PRO-CTCAE) that captures patient-reported symptoms associated with cancer therapies, PROs provide a distinctive view of the patient experience encompassing physical, emotional, and functional domains. These data are critical to assessing treatment effectiveness, tracking side effects, enhancing patient-provider communication and enabling a more patient-centered approach to care [19, 25, 26, 27].

Prior studies have shown that using PROs in patients undergoing cancer treatment promotes the early detection of symptoms [28, 29, 30]. Specifically, monitoring symptoms not only leads to better life quality in patients but also contributes to improved patient satisfaction through more effective communication between patients and health care providers [29, 31]. The CTCAE adoption insures that the evaluation is accurate, so oncologists can intervene more effectively and quickly, which might avoid emergencies. This is possible because CTCAE provides a standard framework for categorizing and grading adverse events, reducing subjective evaluation.

PROs are progressively adopted in clinical trials to assess symptoms and toxicities however, their integration in routine cancer care remains limited [19, 32, 33]. Paper based PROs can be cumbersome for patients and health-care practitioners. These often lead to delayed data processing, inconsistent data quality, and logistical challenges in retrieving and organizing information during clinical consultations [34]. This gap in symptom monitoring forms the basis for the development of electronic solutions like electronic Patient-Reported Outcomes (ePROs) that enable remote monitoring, improve communication, and thus can lead to earlier identification and addressing of modifications in treatment [35, 36, 37]. However, they still lack expressivity and the ability to cover nuances of patient experiences, since such instruments requires patients to select

from fixed responses, and that may impose some limitations in the information to be presented and miss unusual symptomatology or more contextual aspects that can be important when translating the data into clinical interpretation [38].

1.1.1 Research Question

To explore whether these limitations can be overcome, this dissertation proposes the following research question:

Does a chatbot based on large language models have the ability to collect and grade clinically relevant symptom data of patients in accordance with the standards of CTCAE?

1.1.2 Hypothesis

It is hypothesized that a chatbot guided by a large language model is able to conduct structured interviews with patients, retrieving the collected symptom data and grade according to the CTCAE.

1.2 Purpose

This project is part of the Osler Project, which aims to use artificial intelligence to change medical anamnesis. The collection of evidence concerning the history and symptomatology of patients, anamnesis, is rather time-consuming and inconsistent in various clinical environments. The Osler Project suggests the transition to structured, explainable, automated patient interviewing via LLMs, which would allow collecting data more efficiently and at a larger scale, without compromising data quality and enhancing clinical performance.

In the context of this framework, the DALY chatbot was created as a focused application in oncology care, as it poses great challenges in the ongoing monitoring of self-reported symptoms, as discussed in Section 1.1. Its purpose is to help in the assessment of symptoms at the pre-consultation stage. By guiding patients through task-oriented dialogue aligned with CTCAE v5.0 criteria, the system aims to produce structured symptom summaries and their severity grades. Such outputs should help clinicians in symptom assessment and can help to alleviate workload, making more timely interventions.

1.3 Objectives

The objectives of this project were:

- To create a conversational chatbot, which, based on large language models, is able to conduct an interview with the patient and obtain the clinical data about the symptoms.
- To align the collected data with the CTCAE guidelines, as it is useful for clinicians to analyze and adapt the cancer treatment according to the severity of the reported symptoms.
- To support both text and voice interactions so that patients, including those experiencing physical, cognitive, or literacy-related challenges, can use the chatbot.
- To test and assess the system with a small group of oncologists to evaluate the system's effectiveness and accuracy.

1.4 Document Structure

There are six chapters in this dissertation. In Chapter 1, the background and the problem are presented, as well as the expected outcomes. Chapter 2 provides the necessary background to understand the clinical and technical context of the problem. Covering symptom monitoring in oncology, as well as the CTCAE framework, and the technical background of large language models and task oriented dialogue systems. Chapter 3 presents the literature review on digital tools for symptom monitoring within CTCAE guidelines, detailing the types of tools identified, and the gaps in current approaches. Chapter 4 describes the proposed solution, starting with the conceptual system design and followed by the technical implementation details. Chapter 5 provides the assessment outcomes of the clinical testing with both the quantitative and qualitative responses of the physician. In the end, Chapter 6 summarizes the results of the study, as well as its limitations and future steps regarding the system expansion and refinement.

1.5 Contribution to the United Nations Sustainable Development Goals

The Sustainable Development Goals (SDGs) offered by the United Nations are a universal guide toward implementing a more sustainable and better future. It has 17 goals aimed to solve the most important problems in the whole world.

The current dissertation can, possibly, impact a few of this goals, such as healthcare, innovation, and equity. The identified goals and related targets, potential contributions of the project, and potential performance indicators are summarized in the following Table 1.1.

The specific Sustainable Development Goals mentioned have the following names:

SDG 3 Ensure healthy lives and promote well-being for all at all ages

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 10 Reduce inequality within and among countries

TABLE 1.1: Contribution to SDGs and Performance Metrics

SDG	**Target**	**Contribution**	**Performance Indicators and Metrics**
3	3.4	The system facilitates real-time, structured follow-ups of adverse effects of treatment in oncology, which enable early clinical intervention to avoid deterioration of patients' health. It automates the collection of symptoms descriptions to then grade them following CTCAE, while minimizing physician workload.	The quantity of symptoms that are automatically evaluated; The average time of information collection during sessions; The percentage of grades that were determined automatically and confirmed by clinical validation.
	3.8	The system allows universal access to standardized symptom monitoring through reduction of dependency on the availability and access to clinicians as well as maintaining consistency in data collection across resource-limited environments.	Decreased amount of clinician time spent on symptom data collection; Increased rates of full patient assessment carried out independently.

Continues on the next page

Tabela 1.1 (continuação)

SDG	**Target**	**Contribution**	**Performance Indicators and Metrics**
9	9.5	Uses natural language processing systems in clinical workflows as an example of innovation in using LLMs and edge computing to bring medical decision support in the clinical pathway.	Full implementation of local systems without cloud-type dependencies; The number of sessions held in clinical testing; Technical reliability of the system (up-time, data security).
10	10.3	Enables patients of different health and digital literacy to effectively report symptoms, by using simplification of language and questions (simplification of non-medical language and adaptive questions) and use of voice to report the symptoms.	Proportion of patients who were capable of doing interviews independently; Patient satisfaction and comprehension scores.

Chapter 2

Background

2.1 Symptom monitoring in cancer

Cancer therapy is often long and demanding, and patients often experience a range of side effects and symptoms. The average cancer patient usually experiences 9 to 14 symptoms, which are consequences of the disease and the treatment, having a significant effect on everyday life [6, 7, 8].

Because of these adverse symptoms, patients frequently require emergency medical care. A study found that 76% of cancer-related emergency department visits were due to oncology drug-related side effects, with 36% resulting in admission [9]. Another study demonstrated that cancer patients visited the emergency room on average 9.5 days after receiving anticancer drugs due to treatment side effects. Of these patients, 72.8% were admitted for further treatment, and 6.5% died [10]. These statistics demonstrate how crucial it is to identify and treat side effects in cancer patients as soon as possible.

Many of these symptoms are subjective, like pain, nausea or fatigue, so they are not detected with objective monitoring tools, like blood tests, imaging, or wearable sensors. Therefore these monitoring tools cannot replace the value of the patirnt reported information [39, 40]. Moreover, measurable physiological alterations usually follow subjective reports and consequently patient-reported symptoms represent an early alarm system of complications [41].

For effective treatment, oncologists depend on prompt and precise information from patients regarding their conditions [42, 43].

Structured symptom reporting has also been linked to improved patient outcomes. Studies show that patients who were instructed to report their symptoms themselves had a lower rate of emergency room visits, received treatment for a longer period, and had higher one-year survival rates than those who were managed in the standard care [44].

2.2 Common Terminology Criteria for Adverse Events

2.2.1 Overview

The precise and standardized measurement of adverse events AEs is key to guaranteeing safety and effectiveness of cancer therapies. The NCI has developed the CTCAE which offers a widely accepted and structured system of classifying and grading of AEs. Initially set to apply to oncology clinical research trials, the CTCAE has since become a tool in research and everyday cancer treatment, allowing consistent communication, logical documentation and evidence-based choices [45].

2.2.2 Development and Structure of CTCAE

Evolution of the CTCAE

Since the first release in the 1990s, the CTCAE has been revised several times, in the context of the development of new therapeutic strategies and the need to make toxicity reporting more specific and sensitive [45]. The latest, CTCAE v5.0 (issued in 2017) contains 837 terms of the AEs divided by System Organ Class (SOC) and mapped to the Medical Dictionary for Regulatory Activities (MedDRA) ontology [46].

Its structure is designed to ensure that the reporting is concise, coherent and accurate in a way that it would aid in data analysis and more importantly, informed decision making in clinical trials.

CTCAE Grading System

Each adverse event listed in the CTCAE is associated with a severity grade ranging from 1 to 5. While the grading structure is uniform across the system, the specific clinical criteria for each grade vary depending on the nature of the adverse event. These criteria are designed to reflect symptom intensity, functional impairment, and the level of medical intervention required [13]. Below is a general outline of the grading framework:

- **Grade 1 (Mild):** Asymptomatic or mild symptoms; clinical or diagnostic observations only; no intervention required.
- **Grade 2 (Moderate):** Minimal, local, or noninvasive intervention indicated; some limitations in age-appropriate instrumental Activities of Daily Living (ADL).

- **Grade 3 (Severe):** Medically significant but not immediately life-threatening; hospitalisation or prolongation of hospital stay recommended; disabling; limits self-care ADL.
- **Grade 4 (Life-threatening):** Life-threatening consequences; urgent intervention required.
- **Grade 5:** Death related to the adverse event.

It is essential to consult the CTCAE event-specific rubric when grading symptoms, as each adverse event has its own detailed criteria for each severity level.

Limitations of Clinician Based Reporting

This standard grading structure of AEs is heavily dependent on clinician assessment, which can lead to underreporting of subjective symptoms. Research indicates that clinicians usually tend to underreport the severity or occurrence of such symptoms compared to what patients report about themselves [47].

2.2.3 Alignment with Patient Reported Outcome

In order to deal with the shortcomings of AEs reporting by clinicians alone, the NCI created the Patient-Reported Outcome (PRO)-CTCAE. This instrument enables patients to self-report symptoms with the help of structured questionnaires based on the CTCAE terms and makes the toxicity monitoring more accurate and patient-centered, particularly with regard to subjective symptoms [48].

CTCAE and PRO-CTCAE present two different sides of the same coin: the first is a clinical opinion, and the second is a patient experience. In combination they offer a clearer view of treatment tolerability, allowing the earlier identification of complications and the better management of symptoms [49].

Outcomes

PRO-CTCAE usage has resulted in a more precise and complete report of symptoms than clinician evaluations, resulting also in assessing a wider variety of adverse events and allowing earlier action to be taken [50, 51]. According to the reports of the patients, they started to feel more empowered and involved, which helped them live better and feel more satisfied with care [51].

Moreover, studies revealed the correlation between the use of PRO-CTCAE and clinical outcomes, such as the decrease in acute care utilization and the improvement of survival rates due to the better control of treatment-related toxicities [51, 52].

2.3 Large Language Models

The field of chatbots has improved in the recent years because of the large language models. These models are trained on abundant data and have an architecture that allows fluent, relevant and reliable human-like text generation to became a possibility [53].

Prior to the LLM, Natural Language Processing (NLP) was dependent on sequential algorithms, especially Recurrent Neural Networks (RNN) and their successors Long Short-Term Memory (LSTM). These tools proved to be very effective then but because they depended on several parallel mechanisms, they were not satisfactory when it came to capturing long term relationship in the text [54, 55]. With the release of the transformer architecture in 2017 [56], there was a change in NLP.

The transformer brought a new philosophy of design as compared to its predecessors. Using a multi-headed self-attention layer, many parts of an input sequence can be processed simultaneously, considering the relationship in the whole sequence. This advancement not only enhanced the model's context comprehension ability but also made obsolete such typical downsides as vanishing gradients which were common for older architectures [54].

There are two main variants within the transformer architecture when used in LLMs, which are the decoder-only architecture as well as the encoding-decoding architecture:

- **Decoder-Only Models:** These models (e.g., LLaMA3, GPT-4, Qwen2.5) are used in text generation where one word at a time is predicted and the previous predicted words are used as context. This qualifies them to be suitable in open-ended dialogues or creative activities used in chatbots [57, 55].

- **Encoder-Decoder Models:** These models (e.g., T5, BART) are used in situations when the input and output are strongly connected, such as translation or summarization [58].

These architectural advances enable modern chatbots that are based on LLMs, like ChatGPT, Claude or Gemini to benefit in various aspects. They are capable of:

- Sustaining contextually and coherent multi-topics conversations [59].

- Doing few-shot learning, zero-shot learning enabling them to transfer domains with little fine-tuning [59, 60].
- Supporting task-oriented dialogue while preserving the flexibility of open-domain interaction [60].
- Understanding and creating text in a human like way [57, 55, 60].

Other than architecture, recent developments have directed their interest to improving LLM performance in particular domains. Techniques such as Reinforcement Learning from Human Feedback (RLHF) have proved to be useful in the effort of bringing the model outputs closer to human preferences. Prompt engineering solutions such as Chain of Thought prompting [61] provide the possibility of structured reasoning, and a variability of zero-shot and few-shot learning allows the models to generalize to unseen tasks they have not been directly trained on. Soft prompting, multitask prompting among others enhance flexibility among domains [62].

Fine-tuning strategies, prompting methods, and retrieval augmented generation, among others, are considered the major approaches to adapting LLMs to particular tasks or domains and are discussed in the following sections.

2.3.1 Fine-Tuning

Full Fine-Tuning

Full fine-tuning is used for adapting a pre-trained large language model, where every models' parameter is optimized for a specific objective. This approach ensures that while the model retains and uses the pre-trained knowledge, it is adjusted for specific requirements. This approach is computationally expensive, however, it is shown to be a reliable technique to redirect a LLM on such tasks that are more specific [63].

Large language models, when being trained, are given large corpuses which results in parameters $\theta_{\text{pretrained}}$ that provide an understanding of language. By fine-tuning these parameters, through the process of model optimization, the model can be fitted for targeted datasets [64].

The goal of fine-tuning is to minimize a task-specific loss function $L(\theta)$ [65], defined as:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i),$$

where:

- $f_\theta(x_i)$: The model's output for input x_i, parameterized by θ.
- y_i: The true label associated with x_i.
- $\ell(\cdot,\cdot)$: A task-specific loss function, such as cross-entropy for classification or mean squared error for regression.

Full fine-tuning updates all parameters θ using gradient-based methods [65]. The parameters are adjusted iteratively:

$$\theta \leftarrow \theta - \eta \nabla_\theta L(\theta),$$

where:

- η: Learning rate, controlling the step size of updates.
- $\nabla_\theta L(\theta)$: Gradient of the loss with respect to the parameters.

Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning techniques selectively update specific components of a pre-trained model instead of all of their parameters, thereby resulting in a significant reduction of memory costs [66, 67].

Moreover, these approaches do not hinder the general knowledge encapsulated in the LLMs, instead they enable a low-resource and task-specific fine-tuning. Parameter-efficient methods are key in the sense that they allow the use of LLMs in certain applications without having to alter most of the already trained weights in the pre-trained model, by freezing most of the pre-trained parameters and modifying only a small subset. In other words, only a small number of free parameters are adjusted, such fine tweaking of parameters is extremely useful during deployment in specific tasks [66, 68].

Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning method that introduces low-rank updates ΔW to the frozen weight matrices W_0 of the pre-trained model. Instead of directly modifying W_0, LoRA adds a task-specific update [69, 70] represented as:

$$W = W_0 + \Delta W, \quad \Delta W = AB^\top,$$

where:

- $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{d\times r}$ are low-rank matrices.

- $r \ll d$, ensuring that ΔW introduces only a small number of additional parameters.

2.3.2 Prompting

Soft Prompting

Soft prompting is a method for guiding the behavior of large language models without fine-tuning the parameters. Instead of explicitly crafting text-based instructions (hard prompts), soft prompting optimizes embeddings that prompt the model in desired ways [71]. These embeddings act as learned prompts that are annexed to the input, subtly steering the model's response [72].

Supposing the original input sequence is $X = \{x_1, x_2, \ldots, x_n\}$ with n tokens, and the target sequence is Y, to implement soft prompting, the sequence of learned prompt embeddings $P = \{p_1, p_2, \ldots, p_m\}$ is added, where m represents the number of prompt tokens [73]. This results in an extended input sequence Z defined as:

$$Z = \Phi([P;X])$$

where $[P;X]$ denotes the concatenation of the prompt embeddings P with the original input X. The encoder Φ processes this combined sequence Z [73], allowing the prompt embeddings to influence the model's activations without adjusting the main model parameters, meaning without fine-tuning [72].

During training, only the prompt embeddings P and the classification head h_θ are updated, while the parameters of Φ remain fixed [72].

Hard Prompting

The hard prompt is raw human-written instruction or sample with clear and discrete goal aimed to get the response from the language model. In contrast to soft prompting, where learnt embeddings are used to guide the language model, hard prompting makes use of direct text inputs to guide the LLM into generating the predicted outputs. This approach is often implemented through carefully constructed phrases or sentences that frame the model's task [73].

Hard prompting leverages the pre-trained knowledge of an LLM by framing tasks as natural language inputs. The prompts are carefully crafted to include the task's context, instructions, and, if necessary, a few examples.

1. **Instruction-Only Prompt:** The prompt includes a plain task description.

2. **Few-shot prompting:** Tasks's examples are included to guide the model.

3. **Structured Prompt:** Prompts are explicitly formatted to produce outputs in a specific structured format.

4. **Schema Prompt:** Features the utilization of templates that introduce the input context, as well as the format the answers should follow.

Chain of Thought

Chain of Thought (CoT) is a type of hard prompting that aims at improving the reasoning ability of language models by designing the prompts to imitate the natural process of solving a problem. When humans try to solve complex and or multi-step tasks, they tend to view the problem in smaller parts towards the final answer step by step. In this way, chain-of-thought prompting allows LLMs to complete the same steps encouraging the generation of several intermediate steps leading to the final answer [74, 75].

Traditional prompting relies on generating a single-step response directly from the input. In contrast, CoT prompting structures the input to guide the model into producing intermediate steps that lead to the final answer. Let the input sequence be $X = \{x_1, x_2, \ldots, x_n\}$, and the target response $Y = \{y_1, y_2, \ldots, y_m\}$. CoT extends X by adding intermediate reasoning steps $R = \{r_1, r_2, \ldots, r_k\}$, forming an augmented sequence [76, 77]:

$$Z = [X; R; Y].$$

The model generates R and Y sequentially, maximizing the joint probability [76, 77]:

$$P(R, Y \mid X) = P(R \mid X) \cdot P(Y \mid R, X).$$

By explicitly modeling the intermediate reasoning process R, CoT prompts encourage the LLM to "think out loud," improving interpretability and task performance.

2.3.3 Reinforcement Learning from Human Feedback

RLHF is a training paradigm that is used to match the output of large language models with the preferences of humans. In contrast to the approach of working with predetermined labels or datasets, RLHF introduces the human judgment aspect into the learning process, obtaining more reliable, safe, and contextually correct answers [78, 79].

Normally, there are three phases in RLHF pipeline:

1. **Supervised Fine-Tuning (SFT):** It is the original fine-tuning on human curated and labeled set of prompts and the desired answers [78].

2. **Reward Model Training:** Human annotators rank several responses that the model produces. The rankings are then used to train a reward model $r_\phi(x,y)$, that predicts a scalar reward for a prompt response pair (x,y), modelling the preference signal implicitly, meaning it will estimate response quality [80].

3. **Reinforcement Learning:** A reinforcement learning algorithm (e.g., Proximal Policy Optimization) is used to train the base model to maximise the expected reward [80]. Thus, adjusting the policy of the model $\pi_\theta(y|x)$ so that the expected reward of the responses increases:

$$\mathbb{E}_{y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right]$$

   Here, $\mathbb{E}$ denotes the expected value that is, the average reward the model receives when generating responses y according to its current policy π_θ.

2.3.4 Retrieval-Augmented Generation

In the case of generative base models, especially for unusual situations or specific queries, their main limitation is the fact that a model's responses will depend on the information that it initially trained from. This restriction can lead to hallucinations or out-of-context responses [81, 73].

Retrieval-Augmented Generation (RAG) is a hybrid approach that combines the generative model to do dynamic retrieval in relevant documents. RAG retrieves documents related to a query and generates a response by incorporating the retrieved information, in a way that the response is now contextually informed [81, 82].

RAG consists of two primary components:

- **Retriever**: Identifies relevant knowledge from external sources, such as documents or structured datasets, and scores them based on relevance to a given query [81, 82].

- **Generator**: Combines the retrieved knowledge to produce an answer in the context input, guaranteeing that the response is both relevant to specific context as well as true [81, 82].

Retrieval

For a guiven query q, the retrieval step in RAG is responsible for finding information relevant to that query. The goal of the retriever is to rank the k relevant documents $d_k \in D$. The query undergoes the process of indexing to transform it into a vector [82, 83]. For each document d_k, the retrieval probability $P(d_k|q)$ is computed, which represents how relevant d_k is to q. A common approach is to calculate this probability based on cosine similarity between vector embeddings of q and d_k [83].

Generation

After retrieving the top $k$documents documents $\{d_1, d_2, \ldots, d_k\}$ the generation model produces a response via maximizing the conditional probability of the answer a, given both the query q and each of the retrieved documents. The formulated query and selected documents are combined to create a structured prompt, which is then presented to a large language model to generate a response [82, 84].

- Combine q with the retrieved passages to form an augmented input for the generative model:

$$Z = [q; d_1; d_2; \ldots; d_k]$$

- The model generates a response conditioned on Z.

Knowledge Graphs

Knowledge Graphs (KGs) are structured representation of well-established entities and their relationships often represented in a graph format. While unstructured document retrieval may lack precision and explainability, KGs present a rich, structured framework for linking common related concepts [85, 86].

A knowledge graph is defined as $G = (V, E)$, where:

- V is the set of **nodes** used to represent individual elements [87, 88].
- E is the set of **edges** that connect nodes and characterize the relationships between them [87, 88].
- Attributes provide aditional contextual details for nodes and edges, improving reasoning:

  - **Node attributes:** Gives specific information about an entity [89].
  - **Edge attributes:** Add metadata about the connection, such as confidence scores, rules, or conditions to refine the relationships between nodes [90].

The graph enables structured traversal and reasoning.

2.3.5 Context Management

For large language model based chatbots, coherence and relevance throughout repeated human-computer interactions requires both comprehension of current human inputs as well as features to maintain human context across longer interactions or sessions. Naturaly, LLMs manage short-term context through information retention in limited model depth, while their lack of long-term memory causes them to forget previous interactions that must be maintained externally [91, 92].

Since LLMs have a limited context of tokens, a number of approaches have been proposed to maintain long-term memory.

- **Memory Augmentation** This frameworks, such as *MemoryBank*, provide a structured way to store and manage information beyond the context window. The *Memory Bank* organizes entries with timestamps, ensuring that information remains accessible but also decays over time based on relevance, exactly how the human memory works, where the information that is no longer needed and critical fades [92].

- **Structured Memory Systems** These systems implement the long-term memory by aplying structures that enable the chatbot to navigate and retrieve relevant data more effectively [93, 94]. Two prominent domains in this area are tree-shaped structures and knowledge graph-based systems.

  - **Tree Based Systems** For example, the tree-based memory system such as *MemTree* arranged information out in hierarchical form akin to a decision tree. The nodes of the tree refer to a unique piece of information, and the branches represent the link or relationship between the information. This causes the chatbots to only perform targeted retrievals by narrowing down paths in the tree based on the context of the query [94].
  - **Knowledge Graph Based Systems** Models like *RecallM* serve as graph-based memory systems that expand beyond hierarchical relationships to indicate interactions as

nodes and interconnections as edges. This method captures richer, non-linear relationships and allows for more sophisticated reasoning over temporal and relational data. The graph structures enable the chatbot to track the development of concepts across time. As new details updates include data are added, the graph is maintained to adjust edges and nodes to encapsulate the altered context or relationships [93].

- **Retrieval-Augmented Generation**

  Retrieval Augmented Generation, as mentioned in the earlier chapter, allows external knowledge bases or databases and a LLM to work together, so the chatbot can look up information in real-time [92, 95, 96, 97].

- **Dialogue Summarization**

  Management of context to fit within the token limit of the LLM requires summarization. Summarization approaches extracts the most relevant information from earlier parts of the interchange and injects it back into the conversation as needed, preserving the continuity of the dialogue without going beyond the contextual window [98, 99, 100].

2.4 Graph Theory Foundations for Dialogue Systems

Graphs are used in several ways in designing dialogue systems. The particular attention in this section is paid to the graph-based dialogue control mechanisms in which nodes correspond to dialogue states or actions, and the edges represent the set of possible transitions between them, and thus determine the arrangement of conversational flow. These control graphs form the basis of dialogue progression control, state management and decision handling [101].

It is important to distinguish these dialogue control graphs from external knowledge graphs, which represent domain-specific factual knowledge and were discussed in the Section 2.3.4 on RAG.

2.4.1 Formal Definition of Graphs

A graph is formally defined as an ordered pair:

$$G = (V, E)$$

where:

- V is a finite set of vertices (nodes).
- E is a set of edges, where each edge represents a connection between two vertices.

Graphs can be either undirected or directed, depending on organization. In undirected architecture, the edges are simply written as unordered pairs (u,v), without a particular direction between the two vertices. On the other hand, in directed graphs edges have a specific direction: each one is written as an ordered pair (u,v), showing a sequential move from node u to node v [102].

2.4.2 Directed Acyclic Graphs

A Directed Acyclic Graph (DAG) is a directed graph that contains no cycles. They can be defined as:

$$\forall v_i \in V, \quad \text{no path exists such that } v_i \to \cdots \to v_i$$

In a graph without any loops, there exists only one direction, so it is only allowed to travel to forward nodes. However, it is worth mentioning that a node can have several predecessors as well as sucessors [103].

Path Graphs

A *path graph* is a special case of DAG where the nodes are organized as a linear sequence. Each node, besides the first and last, has one predecessor and one successor [104]:

$$P_n = (V,E), \quad V = \{v_1, v_2, \ldots, v_n\}, \quad E = \{(v_i, v_{i+1}) \mid 1 \leq i < n\}$$

where:

- P_n denotes the path graph of order n;
- V is the set of vertices (nodes), indexed sequentially from v_1 to v_n;
- E is the set of directed edges, where each edge connects node v_i to its immediate successor v_{i+1}, for all i from 1 to $n-1$.

2.5 Task-Oriented Chatbots

2.5.1 Traditional Task-Oriented Chatbot Architectures

The classical task-oriented chatbot was built on a modular architecture where the different functionalities were separable into different dedicated components [105]. There were typically three modules in the pipeline: Natural Language Understanding (NLU), Dialogue Management, that included Dialogue State Tracking (DST) and Policy Learning, and Natural Language Generation (NLG) [106].

The NLU module interprets user speech to identify the intents and identify the associated entities or slots. As an example, a medical appointment system could have the slots containing the name of the patient, symptoms, and the preferred date of appointment. The dialogue manager therefore maintains the dialogue state by recording the filled slots and determines the system's next action by using either a rule-based reasoning or learnt dialogue policies. Finally, NLG module converts the planning of the system into natural language answers usually through hand-built templates [106].

The earlier systems initiated their operations using finite-state machines, which correspond to simple directed graphs [107] , or decision trees. Consequently, the dialogue flows were rather deterministic and interpretable. These systems were able to perform well in structured fields, due to the rule-based slot fillings, yet they could not understand language well enough and reacted poorly to unsual user input [108].

Besides, the pipeline structure had a susceptibility towards error propagation: inaccuracies at the NLU stage (wrong intent classification and entity extraction) could propagate downstream and compromise the overall coherence of the dialogue. Though later individual modules were improved by statistical methods, inherent modularity would impose upper bounds on flexibility and naturalness [106].

2.5.2 LLM-Driven Task-Oriented Chatbots Architectures

The application of large language models in chatbots has revolutionized the development of dialogue systems since they have the potential to provide generative and language understanding capabilities [53].

Chatbots can be used to process a huge range of user interactions without adhering to predefined sets of rules or slots through the use of LLMs [109]. Such flexibility allows having more

fluent and coherent conversations, as chatbots are able to recognize various expressions and complex sentence constructions [110].

Recent research has explored several strategies for integrating LLMs into Task-Oriented Dialog (TOD) systems.

SimpleTOD [111] formulates TOD as a single sequence generation problem using a LLM. The model predicts sequentially the belief state, database search outcomes, action choices and delexicalized system responses at each turn, given the entire history of the dialogue. This single unification allows performing end-to-end training and inference without any modular supervision or task-specific modules.

SGP-TOD [1] dialog management is broken into specific prompting modules. The DST Prompter encourages the LLM to directly retrieve the belief state as slot-value pairs via dialogue history, and the Policy Prompter specifies the system action to be taken next via policy skeletons which define permissible system behaviors. Such architecture retains complete schema control through prompting only, meaning that it can be adapted to new domains with zero-shot learning by only changing the task schema definitions, without fine-tuning the model. The architecture of this system is illustrated in Figure 2.1.

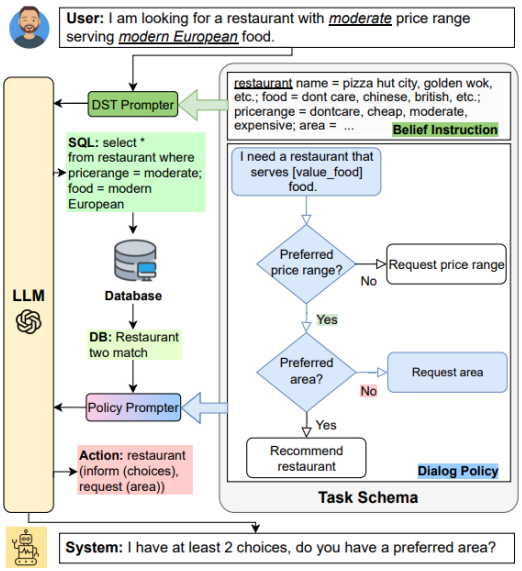


FIGURE 2.1: The proposed SGP-TOD architecture is depicted with a dialog example [1].

More recently, Cao et al. [2] introduced DiagGPT, a multi-agent framework that orchestrates LLMs to manage dialogue state in complex task-oriented conversations. DiagGPT does not use slot-filling or belief state extraction, instead uses a *Topic Manager* module to dynamically decide

what the next dialogue step should be based on topic transitions, keeping a stack-like structure to keep track of pending, active and completed topics. All modules, such as topic enricher and chat agent, interact through dedicated LLM prompts, allowing the system to both actively walk users through predefined checklists and freely deal with topic changes and user-instigated deviations. Figure 2.2 shows the general architecture of DiagGPT and how its Topic Manager, Topic Enricher, Context Manager, and Chat Agent components help to control the flow of topic-based dialogues.

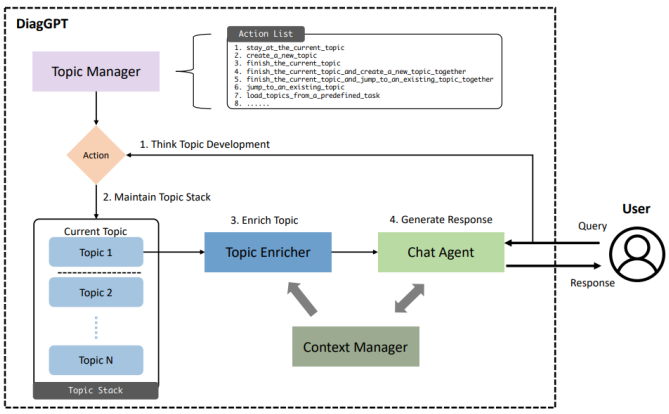


FIGURE 2.2: The framework of DiagGPT [2].

Sohn et al. [3] proposed **TOD-Flow** , a graph-based control framework in which the latent task structure of task-oriented dialogues is learned directly from annotated dialogue act data. The TOD-Flow graph emcompasses three kinds of connections amid dialog acts: *Can* (the permissibility of an action), *Should* (the preferred actions), and *Should Not* (the actions that should be avoided) depending upon the existing state of dialogue. These relations are inferred via decision tree models trained on observed dialog trajectories without requiring manual definition of task schemas or workflows. TOD-Flow can be thought of as a filtering and ranking layer, once learned it constrains the outputs of both dialogue policy models and end-to-end LLM-based dialogue generators, significantly increasing prediction accuracy and consistency of responses. This system architecture is in the Figure 2.3

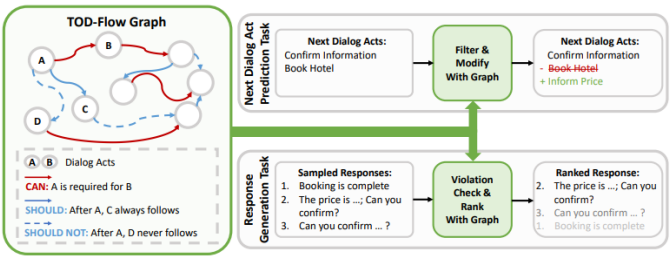


FIGURE 2.3: TOD-Flow architecture [3].

Walker et al. [112] introduced **GraphWOZ** a dialogue management framework in which the dialogue state is modelled as a dynamic knowledge graph instead of the static slot-based representation. The nodes of the graph represent entities (e.g., persons, events, locations, groups) or dialogue elements (e.g., utterances, entity mentions) and the edges represent semantic relations (attendance, organization, or coreference, etc.). As the conversation progresses, new nodes and relations are incrementally added or updated. The dialogue management system works in two major components: conversational entity linking, mentions in user utterances are linked to graph entities with neural classifiers over string-based and graph-based features; response ranking, where subgraphs are natural language summaries and prepended to the dialogue history to condition response selection by a fine-tuned language model.

AutoTOD [106] is a fully end-to-end task-oriented dialogue system introduced by Xu et al. that abolishes the modular design of TOD systems by letting a LLM take full dialogue control. Within this framework, all fundamental dialog capabilities such as intent recognition, dialogue state tracking, policy learning, slot filling, Application Programming Interface (API) calling, error handling and natural language generation are handled by the LLM. The system is defined purely in terms of schema-guided zero-shot prompting, where a task schema, a description of available APIs, slot definitions and constraints, is inserted into the prompt to direct the reasoning of the LLM at each step. Structural knowledge is offered by this schema, nonetheless, no dialogue flow management or state machine is enacted explicitly, the LLM takes dynamic control of interpreting both the user input and the present dialogue context to decide independently what to do subsequent.

The LLM ContextBridge architecture [113], which was designed as an automotive virtual assistant, features an architecture of a dual interpretation pipeline that takes advantage of the strengths of both rule-based systems and generative systems. To answer simple questions, an intent classification and slot extraction is done by a classical NLU module. When user utterances are unclear, incomplete or require vast context (e.g. “and tomorrow?), the system makes use of an LLM to rephrase the utterance into an explicit and full command. This reformulated input is again fed back into the common pipeline so that improved comprehension can be achieved without retraining and changing downstream parts.

The Script-Based AI Therapist [4] integrates deterministic finite-state machines and the generative properties of LLMs to guarantee clinical safety and natural flow of therapeutic dialogue. The conversation is controlled by a pre-scripted finite state machine, and every state denotes a fixed therapeutic step under psychological protocols. The planning and the selection of the next action is not the responsability of the LLM, but it has two roles. First, it translates the structured

instruction of each state into fluent, caring, and contextually appropriate natural language. Second, it determines whether the response of the user meets the clinical expectations of the step they are on, and thus decides whether it is reasonable to continue in the script or rephrase the prompt. Such a dual-role design allows a strict adherence to the protocol, utilization of the advantages of the LLM in empathy and flexibility of language understanding, without any safety or interpretability sacrifices.

Figure 2.4 illustrates the general architecture of the Script-Based AI Therapist. The Assessor LLM identifies whether the input of the user meets the therapeutic objective of the state in question; when this is the case, the Dispatcher LLM will use the next part of the structured script, and the Dialog LLM will verbalize this in a natural, empathetic manner.

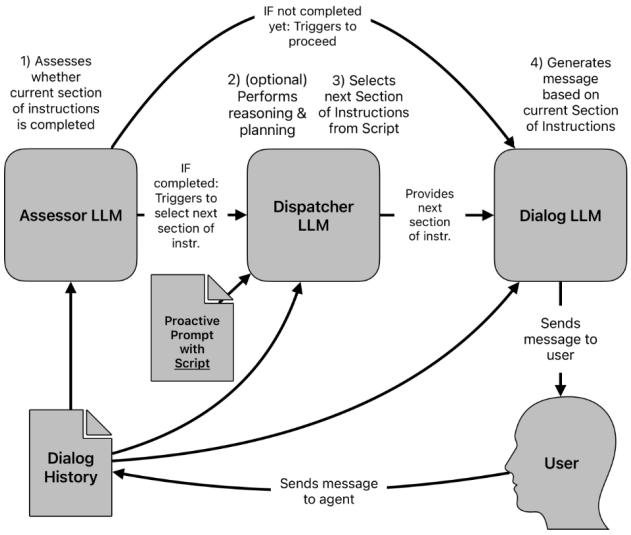


FIGURE 2.4: General architecture of the Script-Based AI Therapist [4].

The main architectural characteristics of the discussed LLM-driven task-oriented dialogue systems are summarized in Table 2.1.

TABLE 2.1: Comparison of modern LLM-driven task-oriented dialogue architectures.

Approach	**Architecture Type**	**Role ofLLM**	**Dialogue Control**	**Limitations**
SimpleTOD [111]	End-to-end supervised	Full generation (state tracking, policy, response)	Fully implicit (no external control)	Requires large annotated datasets; lack of transparency:

Continued on next page

Table 2.1 – continued from previous page

Approach	**Architecture Type**	**Role ofLLM**	**Dialogue Control**	**Limitations**
SGP-TOD [1]	Schema-guided prompting	Dialogue state tracking and policy generation via schema-based prompts	Schema-constrained prompting	Domain Adaptability; schema dependency
DiagGPT [2]	Multi-agent prompting orchestration	LLM per module (topic manager, topic enricher, chat agent)	Topic stack tracking with LLM-controlled transitions	Complex prompt design; limited explicit control
TOD-Flow [3]	Graph-inferred policy learning	Policy ranking conditioned on graph	Learned dialog act graph (Can/ Should/ ShouldNot)	Requires dialog act annotations
GraphWOZ [112]	Entity-centric knowledge graph	Response generation conditioned on entity subgraphs	Knowledge graph state tracking	Entity linking ambiguity
AutoTOD [106]	Fully zero-shot schema-prompted LLM	Full LLM decision making (intent, policy, API calls)	Fully implicit control with schema injected in prompt	No interpretability; lack of transparency
LLM ContextBridge [113]	Dual NLU pipeline with LLM refiner	Reformulates ambiguous utterances for classical pipeline	Dialogue state and flow managed by legacy pipeline	Latency and consistency issues; dual pipeline complexity
Script-Based AI Therapist [4]	Script-based state machine + LLM surface realization	Transforms structured prompts into natural therapist dialogue	Finite-state controller ensures clinical protocol adherence	Requires intensive scripting; limited handling of off-script input

Synthesis

End-to-end approaches such as SimpleTOD and AutoTOD aim at complete automation, with the LLMs having complete control of the dialogue without any structure. Although this results in compelling fluency and simplicity, it directly costs interpretability, clinical traceability and the capability of enforcing domain-specific protocols, such that these architectures are not appropriate in safety-critical settings like clinical symptom assessment.

Systems such as SGP-TOD and DiagGPT use modular prompting approaches, in which various subtasks (e.g. state tracking, policy generation) are addressed with structured prompts. These systems maintain flexibility while preserving control via schemas or topic stacks, offering a middle ground between raw LLM power and structured task execution. Their versatility to fit in new areas without any retraining is particularly attractive in the changing health care environment.

TOD-Flow and GraphWOZ use graph-based representations to model dialogue flow or knowledge states. Whereas TOD-Flow implicitly learns the dependencies between actions based on dialog act information (Can / Should / Should Not), GraphWOZ deals with the entity-level semantics by use of dynamic graphs. These methods provide formal organization and allow decision-making to be consistent, but GraphWOZ adds more complexity that is unnecessary in situations requiring a protocol-based approach to decision-making.

Hybrid designs like LLM ContextBridge combine deterministic flows with selective LLM intervention. The fundamental flow of conversation in the system is determined by some predetermined rules or scripts. The LLM is only called when the user does not do what is expected, e.g. when the input is vague or not on-topic, to rephrase, clarify or deal with exceptions. But the LLM does not make the decision of whether the system will proceed to the next step, this is still in the control of the rule based engine.

The Script-Based AI Therapist follows a strictly scripted flow defined by a finite-state machine, but integrates the LLM more actively and continuously. At every step, there are two distinct roles that the LLM fulfills. First, to transform structured therapeutic instructions into natural, empathetic language. Second, it will serve as an assessor that determines whether the input of the user meets the clinical goal of the current state. It is only in case of such positive validation that the system goes to the next scripted step. This configuration guarantees maximum adherence to the protocol and takes the best of the benefits of LLMs in the context of natural language processing and user interaction, which is extremely applicable to sensitive and safety-sensitive areas like clinical interviews.

Chapter 3

Literature Review

In order to methodically find and categorise existing approaches and solutions for the research question, a scoping literature review was conducted to evaluate and compile the current status of the creation and application of digital symptom monitoring tools for cancer patients. It specifically evaluated the technology employed, the methodology's adherence to CTCAE guidelines, the usefulness and efficacy of the suggested solutions, and the effect these solutions have on patient experience and symptom control. This review sought to provide guidance for the following question: **What digital tools based on CTCAE are being used for cancer patient symptom monitoring, and what are their features, usability, and reported effects on symptom management?**

3.1 Methods

A scoping review following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses Extension for Scoping Reviews (PRISMA-ScR) guidelines [114], was employed in this scoping review to provide a more standardized approach to reporting the investigation of digital tools for monitoring cancer treatment symptoms based on the CTCAE.

3.1.1 Eligibility Criteria

Studies were selected according to predefined inclusion criteria in Table 3.1 focusing on digital tools for monitoring cancer treatment related symptoms aligned with CTCAE criteria. Eligible studies included adult or pediatric cancer patients undergoing active treatment where a digital intervention (such as an mobile Health (mHealth) app, web platform, AI based system, or other electronic tool) was used for symptom tracking or toxicity monitoring in accordance with CTCAE or

PRO-CTCAE standards. Only original research articles (including intervention studies, pilot/feasibility studies, or system development papers) published in English from 2014 onward were included. This date cutoff was chosen to capture recent advancements following the introduction of PRO-CTCAE and the rise of mobile and digital health solutions in oncology. Studies focusing purely on therapy outcomes without a symptom monitoring component, not using CTCAE criteria, not involving a digital tool, or not describing the tool's development/implementation were excluded.

TABLE 3.1: Inclusion criteria.

Category	Inclusion Criteria
Population	Cancer patients undergoing treatment with relevant symptom monitoring.
Intervention	Digital tools for monitoring cancer treatment symptoms aligned with CTCAE criteria.
Technology Type	Digital tools, such as mHealth, eHealth, AI-based, or electronic platforms for oncology symptom tracking.
Outcome	Study reports on symptom monitoring outcomes using CTCAE.
Study Focus	Development, design, or implementation of digital tools for oncology symptom monitoring.
Language	Published in English.
Publication Date	From 2014 onward to capture recent advancements in digital health.
Study Type	Original research (e.g. clinical trials, pilot studies, feasibility studies, implementation studies).

A comprehensive search was conducted across four databases: PubMed, Scopus, EBSCO, IEEE Xplore and ACM Digital Library. The search was structured to capture relevant studies on digital tools designed for symptom monitoring in oncology, leveraging the following boolean query: ("cancer" OR "oncology") AND ("digital*" OR "mHealth" OR "electronic*" OR "eHealth" OR "AI" OR "Artificial Intelligence") AND ("monitor*" OR "evaluate*" OR "assess*" OR "track*") AND ("symptom*" OR "side effects" OR "Adverse Events") AND ("CTCAE" OR "Common Terminology Criteria for Adverse Events"). This query was adapted as needed for each database's search interface. The final search was executed in december 2024, and an updated sweep was done in february.

3.1.2 Selection of Sources and Screening

The initial search yielded 593 articles. A total of 203 duplicate articles were removed using Excel, leaving 390 unique articles. Titles and abstracts of these articles were screened, leading to the exclusion of 248 articles based on irrelevance to digital symptom monitoring or non-aligment with CTCAE criteria. After a more refined review, 54 articles were excluded for not being digital tools, 48 for evaluating existing tools without detailing their development, and 10 more because they did not discuss the development of the digital tool in question. After this thorough screening, 15 articles were included and the study selection process is illustrated in a PRISMA flow diagram in Figure 3.1.

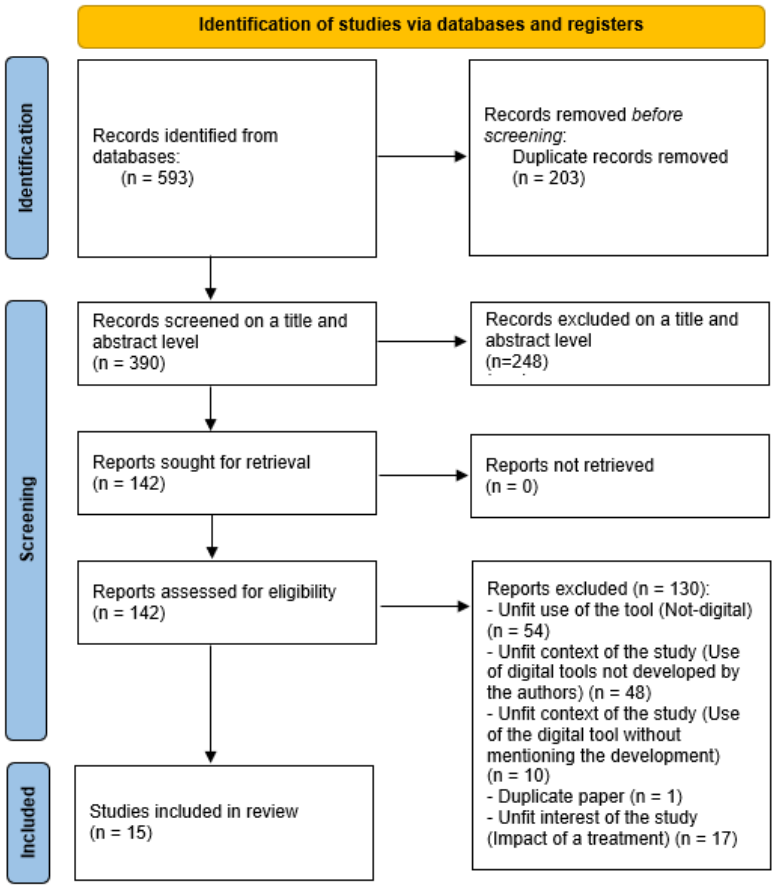


FIGURE 3.1: PRISMA 2020 flow diagram of the study selection process.

3.2 Results

A data extraction form was developed to capture key information from each included study, including: author, year, country, study design, patient population, type of technology, integration

of CTCAE/PRO-CTCAE, and main outcomes (usability findings, symptom control outcomes, etc.). For further evaluation of the articles, an assessment tool, Scale to Assess the Methodological Quality of Studies Assessing Usability of Electronic Health Products [115] was used, and the articles were evaluated on 15 questions from the assessment tool with yes/no.

Fifteen studies were included and analyzed. The studies were heterogeneous for the types of digital symptom monitoring tools evaluated, with the most common types being mobile applications web-based and Electronic Health Records (EHR) integrated platforms.

The majority of studies found high usability with System Usability Scale (SUS) scores usually between 86 and 93, which indicates good patient involvement.

Clinically, several of these tools have benefits, including better symptom management, improvement in quality of life, and decreased healthcare utilization with fewer hospital admissions and emergency department visits.

Specific study details and results are provided in Table 3.2 offering a mechanism of comparison between study designs, methods, and findings.

TABLE 3.2: Digital Tools Using CTCAE or PRO-CTCAE: Instruments and Clinically Relevant Outcomes

Study	**Country**	**Design**	**Digital Tool**	**Instrument Used**	**Main Outcomes**
Soh et al. (2018) [116]	South Korea	Interventional observational study; 203 patients	Mobile app *LifeManager*	PRO-CTCAE	QoL improved significantly (EORTC QLQ-C30, $p = 0.02$); high satisfaction (4.06/5); 86.7% completion rate.
Egbring et al. (2016) [117]	Switzerland	Randomized controlled trial; 139 breast cancer patients	Mobile/web app with ECOG	CTCAE v4.0	High symptom reporting; median survey 3.5 min; *no direct clinical outcomes reported.*
Kim et al. (2023) [118]	South Korea	Pilot study; 30 patients	Mobile app *Smart Cancer Care*	CTCAE v5.0 / PRO-CTCAE	264 alerts generated; usability 4.06/5; *positive feasibility, no QoL or efficacy data.*
Lencioni et al. (2015) [119]	USA	System evaluation; 350 patients	Web-based AERS platform	CTCAE v4.0	AE reporting ↑75%; queries ↓60%; *focus on reporting efficiency.*
Hassett et al. (2022) [120]	USA	Multicenter; >3,000 patients	Web platform *eSyM*; Epic-integrated	PRO-CTCAE	26,268 surveys; 264 alerts; *feasibility demonstrated at scale.*

Study	**Country**	**Design**	**Digital Tool**	**Instrument Used**	**Main Outcomes**
Kestler et al. (2021) [121]	Germany	Clinical trial; 66 patients	Mobile app *NEMO*	CTCAE v5.0	Significant improvement in QoL from baseline to end; high usability (SUS 92.5/100).
Lapen et al. (2021) [122]	USA	Pilot; 678 breast cancer patients	ePRO weekly surveys via portal	PRO-CTCAE	Reduction in moderate-severe toxicities ($p < 0.001$); 72% engagement; anxiety unchanged.
Gomaa et al. (2023) [123]	USA	Pilot; 34 GI cancer patients	SMS-based *RT-CAMSS* with chatbot	PRO-CTCAE	PAM score improved ($p = 0.02$); 7.4% triggered alerts; *no QoL/efficacy outcome*.
Moradian et al. (2023) [124]	Canada	Development with stakeholders	Web platform *V-Care*	CTCAE (irAEs)	High usability (SUS 86.2/100); 94% completion; *no clinical outcome measured*.
Basch et al. (2014) [47]	USA	Development and validation	Web PRO-CTCAE platform	PRO-CTCAE	Test-retest reliability $r > 0.8$; *instrument validation only*.
Grašič Kuhar et al. (2020) [125]	Slovenia	Controlled cohort; 91 patients	Mobile app *mPRO Mamma*	PRO-CTCAE	QoL improved (+10.1 and +10.6); trend toward fewer hospital visits.
Hægermark et al. (2023) [126]	Norway	Usability test; 10 patients	Web prototype with feedback	CTCAE / PRO-CTCAE	High usability (SUS 92.5/100); *no clinical effectiveness data*.
Barz Leahy et al. (2021) [127]	USA	Feasibility; 52 pediatric patients	*Pedi-PReSTO* online web-based form tool	Ped-PRO-CTCAE	92% participation; 21% of reports triggered clinician action; *positive feasibility only*.
Underwood et al. (2022) [128]	USA	Usability/feasibility; 25 patients	Mobile app *mPROS*	PRO-CTCAE	84% would use; 76% recommend; *no outcome on efficacy or QoL.*
Taramasco et al. (2023) [129]	Chile	Co-design; 12 patients + clinicians	Prototype app *+Contigo*	CTCAE / PRO-CTCAE	Identified needs (e.g., tailored language); *development-focused, no clinical results*.

Abbreviations: *CTCAE, Common Terminology Criteria for Adverse Events;PRO-CTCAE, Patient-Reported Outcomes version of CTCAE; eSyM, Electronic Symptom Management; mHealth, mobile health; ePRO, electronic patient-reported outcomes;EHR, electronic health record; QoL, quality of life; SUS, System Usability Scale; SMS, Short Message Service; AI, artificial intelligence; AE, adverse event; irAEs, immune-related adverse events; EORTC QLQ-C30, European Organisation for Research and Treatment of Cancer Quality of Life Questionnaire-Core 30; PAM, Patient Activation Measure.*

The development and use of digital devices for symptom tracking in oncology has demonstrated promising results, major obstacles, and beneficial effects on patient care. Whether they consist of mobile application (app), web-based or Electronic Medical Records (EMR) programs, these tools show the importance of user centric design, teamwork with different specialists, and extremely strict security levels in meeting the needs of oncology patients as well as healthcare providers.

3.2.1 Types of Digital Tools Developed

Based on their technological approach, the digital tools found in this scoping review can be divided into four main categories: Mobile apps, Web-Based Platforms, Short Message Service (SMS)-Based Systems, Hybrid Platforms (Mobile + Web). This classification aids in mapping the variety of digital tools available for oncology symptom monitoring.

- **Mobile apps**
  - *LifeManager*: An educational chat with to-do lists in a mobile app.
  - *Smart Cancer Care*: Android/iOS app for monitoring symptoms of cancer.
  - *NEMO*: Mobile app for offline access and data transfer with Quick Response (QR) code.
  - *mPRO Mamma*: Mobile app with feedback communications that are encrypted.
  - *mPROS*: Mobile app for capturing toxicity during radiation therapy.
  - *+Contigo*: Co-designed mobile app for tracking symptoms post breast cancer treatment with a focus on patient-centered design.
- **Web-Based Platforms**
  - *AERS*: A web-based platform for capturing adverse events integrated with EHR for seamless integration.
  - *eSyM*: platform integrated with EpicEHR.
  - *V-Care*: Web platform for the recording of irAEs.
  - *PRO-CTCAE System*: Online symptom reporting system for patients.
  - *mHealth app Prototype*: Prototype to monitor chemotherapy side effects via web interface.

  - *ePRO System*: Monitored remotely with a patient portal for acute post-radiation toxicities.

- **Hybrid Platforms (Mobile + Web)**
  - Mobile and Web App: Incorporates the ECOG with sliders symptom tracking.

- **SMS/chatbot Based Systems**
  - *RT-CAMSS*: SMS-based system with chatbot for monitoring symptoms.
  - *Pedi-PReSTO*: Monitors symptoms with online forms and reminders sent via SMS and email.

3.2.2 Symptom Monitoring According to CTCAE

All tools in the studies employed CTCAE frameworks or its patient-reported adaptation PRO-CTCAE, ensuring that the symptoms monitored were standardized and clinically relevant. This alignment allows patient-reported data to be mapped to the common grading (Grade 1–5) used by oncology professionals, facilitating clearer communication and potential integration with clinical decision-making. In practice, most digital tools simplified the way CTCAE grading was presented to patients. For example, some apps used plain-language phrasing or intuitive slider scales to represent severity, rather than medical jargon.

Several tools utilized the PRO-CTCAE item library. Many subsequent digital interventions built their symptom questionnaires from these items. For instance, the PRO-CTCAE was the basis of the survey content in the *eSyM* program, the Lapen et al. remote monitoring system for radiation oncology, and the Underwood et al. mPROS app for radiation toxicities.

A noteworthy point is that the pediatric adaptation (Ped-PRO-CTCAE) was used in the children's study. This version adjusts questions to be answerable by children (ages 7 and up) or their caregivers, reflecting differences in symptom perception and communication in younger patients. The success of that study (with high response rates from children and parents) indicates that CTCAE-based monitoring is feasible outside the adult population when properly tailored.

3.2.3 Technology and Frameworks Used

A central feature that stood out in some of these applications was the cross-platform application interfaces for building applications that can work across various operating systems. For

example, NativeScript was used in the *NEMO app*, which allowed users to track their symptoms in real time on both Android and iOS. Moreover, Node.js was also used.

For the purpose of early stage designing and prototyping, Figma's contribution was significant during the first stages of development of initiatives such as the *V-Care* platform. Because it enabled designers to design clickable demo screens to present to the users of the platform for their feedback before the actual development was done.

Integration with existing clinical systems required attention to interoperability standards. EHR standardisation was necessary in tools such as *AERS* and *eSyM* which were components of the EpicEHR. This included the ability to transfer data securely using HL7 messaging and MirthConnect. As a result of including these instruments in the current EHR systems, clinician efficiency was enhanced since symptoms of interest could be viewed in the patient's record in real-time. In terms of an adverse drug event reporting system, such integrations were key in reducing the burden of data management processes and improving the quality of data collected.

Protecting data and ensuring the confidentiality of information were given utmost consideration in the design of all the electronic systems especially because of the delicate nature of the information they handled. Systems implemented mechanisms such as encryption and OAuth 2.0 to ensure safe authorization while restricting access to patients' information during interaction. For instance, the *NEMO* application introduced a unique QR code protocol for offline data transfer, ensuring that information could be securely transferred without relying on internet connectivity.

3.2.4 Outcomes

Usability was the main strength found in the studies for these tools, and in most of the studies they scored highly. For example, the SUS score of the *NEMO* app was 92.5, indicating that elderly patients were deliberately considered in the design of the app with complete font size, high contrast, and intuitive navigation. Likewise, the *V-Care* platform had a SUS score of 86.2, indicating very good usability and retention rates (81% of eligible patients were recruited, with 85% completing weekly symptoms questionnaires over 12 weeks). The Norwegian prototype tested by Hægermark et al. also reported SUS 92% and found that the majority of tasks in a usability test were completed successfully by patients, with only minor usability issues identified. Underwood et al. did not use SUS but reported that 84% of patients would continue using their app and 76% would recommend it, reinforcing ease of use.

Beyond usability, an essential question is whether these digital monitoring interventions lead to meaningful improvements in clinical outcomes or processes. Several studies reported on symptom control, quality of life, and healthcare utilization metrics:

- **Symptom detection and alerts**: Tools like *eSyM* and *RT-CAMSS* generated alerts for severe symptoms, enabling care teams to intervene. In *eSyM*, 264 alerts (for high-grade symptoms) were triggered across six health systems. In*RT-CAMSS*, about 7.4% of weekly symptom reports led to an alert for clinical follow-up. These systems thereby captured issues that might otherwise have gone unnoticed between visits.

- **Quality of life (QoL)**: Several interventions aimed to improve patient QoL by better managing symptoms. The *mPRO Mamma* app trial reported significantly higher QoL scores in the app group; within the first week of therapy, global QoL improved by an additional 10.1 points in the intervention arm compared to controls ($p = 0.02$). By the end of treatment, the difference in QoL summary score was even larger (+10.6, $p = 0.002$). These are clinically meaningful improvements, suggesting that proactive symptom monitoring and self-management can positively impact patients' well-being during chemotherapy. Soh et al. similarly found improved QoL in patients using their *LifeManager* app (with a significant increase on EORTC QLQ-C30 scores, $p = 0.02$) along with high user satisfaction. On the other hand, some studies focused on acute toxicity management like Lapen et al. did not measure QoL directly but showed that better symptom control (fewer severe toxicities) can be achieved, which likely translates to improved overall patient comfort.

- **Healthcare utilization**: Some evidence emerged that digital monitoring can influence healthcare utilization, such as emergency visits or hospitalizations. Egbring et al. noted that routine symptom tracking did not negatively affect functional status and might help maintain it. More directly, Lapen et al. observed that among breast cancer patients undergoing radiation, those using the ePRO system had a significant reduction in moderate-to-severe toxicities over 2 months, which could translate to fewer urgent care needs. Grasić Kuhar's study did not find a statistically significant difference in hospital visit rates, but there was a trend toward fewer hospitalizations in the app group (37% had at least one hospital visit vs 54% in control, $p = 0.13$), hinting at a potential reduction in acute care utilization. As most included studies were pilot trials, were not powered to detect survival differences, but they do reinforce that patients were often able to manage symptoms better. For instance, Gomaa et al found that the patient activation measure (Patient Activation Measure (PAM))

improvement of +3.7 points in the RT-CAMSS pilot suggests patients felt more in control of their care, which in theory could reduce unnecessary healthcare use.

- **Workflow efficiency and data quality**: From the provider/system perspective, integrated tools showed gains in efficiency. The *AERS* system dramatically reduced the time needed for research staff to clarify adverse event data (60% reduction), and it cut down on missing data in trial reports. These improvements imply that digital tools can streamline processes that traditionally consume significant clinical or administrative time. Providers in the pediatric study responded that the daily symptom emails were useful 97% of the time and easy to interpret, indicating that when information is presented well, it can be readily incorporated into care decisions without adding burden.

3.3 Discussion

This scoping review discussed the present situation of the digital tools currently available to assist with symptom monitoring in oncology based on CTCAE or PRO-CTCAE frameworks. Although the majority of the tools proved to be technically feasible and highly usable, only a small fraction reported improvement of the clinical outcomes. This discussion identifies common patterns, addresses possible factors that may explain them, and outlines implications regarding development and research in the future.

3.3.1 Clinical Impact and Technology Type

Out of the 15 studies considered only 4 of them explicitly stated that there were statistically significant changes in patient outcomes in terms of quality of life or decreases in treatment related toxicities. It is important to note that each of these interventions was provided through mobile applications that allowed active symptom monitoring, personalized feedback, or education. This indicates that mobile platforms perhaps offer resources over web-based or SMS systems, which could be because of their availability, timely, and the ability to incorporate interactive features. But because many of the studies did not measure clinical impact (e.g. QoL) as an endpoint, the lack of evidence does not always mean evidence of no effect.

3.3.2 Usability and Patient-Centered Design

The majority of the tools had high scores of usability, especially those that were designed using the co-design strategies including patients in the prototyping stage. Although usability in itself is not sufficient to ensure clinical effectiveness, it could improve patient engagement, which subsequently would promote symptom reporting and prompt response.

3.3.3 Standardisation and Flexibility in Measurement of Symptoms

All the tools were based on CTCAE or PRO-CTCAE frameworks as expected on the basis of the inclusion criteria. They were, however, implemented rather differently. Some implemented the entire PRO-CTCAE item library whereas others modified the wording, included sliders, or modified phrasing to increase accessibility.

Such adaptations represent a compromise between standardisation and flexibility, which is especially significant in cases when the literacy level or age group is involved. Application of Ped-PRO-CTCAE in a single study demonstrated that reporting according to CTCAE can be successfully adapted to the paediatric populations.

Although such modifications can enhance ease of use, they can also cause issues with data comparability between tools. Future studies are needed to explain the process of modification and validation of PRO-CTCAE items in every digital setting.

3.3.4 Alerts and Workflow Integration

A subset of tools (e.g., *eSyM*, *RT-CAMSS*) incorporated automated alerts to notify care teams of severe symptoms. These were also among the few tools that provided clinician follow up actions and enhanced workflow efficiency. As an example, AERS platform reduced the clarification requests by 60%. These results indicate that the implementation of symptom monitoring into the current clinical practices not only enhances the quality of data but might allow earlier interventions. Nevertheless, in most studies, alerts did not directly translate to documented clinical decisions, which implies that they are not fully integrated into the processes of decision-making.

3.3.5 Gaps and Research Opportunities

The analyzed literature was mostly pilot projects, observational designs, or feasibility studies. There were few randomised controlled trials, and few reported strong clinical outcomes, like quality of life, hospitalisation or adherence to treatment, which hinders the possibility of making final

conclusions concerning effectiveness.

Another restriction is that most of the tools lacked follow-up clinical validation. Multiple interventions proved to be highly usable and feasible, yet never underwent clinical outcome evaluation rigorously. In the absence of this second step of validation, it is not clear that these digital tools have any effect at all in improving patient care.

Also, there is a variety of technologies, outcome measures, and target populations, which makes meta-analysis or benchmarking difficult. Multicentric trials with well defined primary outcomes and comparison groups should be prioritised in the future, to enable more solid conclusions regarding efficacy, cost-effectiveness, and long-term advantages.

Besides these evidence gaps, a number of methodological limitations were identified throughout the reviewed studies. These include small sample sizes, lack of control groups, and limited reporting on data security protocols. Moreover, the digital literacy and the ability to have access to internet-enabled smartphones can be a limitation to some groups of patients. Future research and implementation strategies should put these problems into consideration.

Lastly, the majority of tools found in the review were based on the static and form-based approach to reporting symptoms, which include checklists or multiple choice surveys. Even though this format is easy to implement, it might be possible that it does not perfectly match the experience of the patient and fit the variations of language and expression. Dynamic tools that capture patients' complaints like chatbots are barely investigated in this area. The review found only a single paper that adopted the use of a chatbot, and it adhered to a rule-based logic with SMS, without dynamic reasoning and generation of natural dialogue. This demonstrates an existing gap in the landscape in terms of technology and how more expressive and adaptive interaction models can be used to assist in organized symptom gathering.

3.3.6 Future Directions

In the future, the scaling and implementation of these tools in wider, multicentre investigations must be a priority where their efficacy can be evaluated in multiple populations and healthcare settings. Wider validation will be required to comprehend generalisability and feasibility in the real world other than the controlled pilot settings.

Long-term effects of these tools on clinical outcomes, healthcare utilisation and cost-effectiveness have to be evaluated. Although the feasibility and usability of the intervention is the topic of many

existing studies, future assessments are required to determine whether these digital interventions can indeed benefit patient care and resource utilization.

The utilization of innovative technologies, including LLM, opens a positive prospect. Specifically, the intersection of chatbots with LLM consistent with CTCAE guidelines may provide the opportunity to conduct smart symptom triaging during natural conversation in real-time. This can automate the process of patient reporting and cut on the workload of clinicians by limiting manual symptom reporting.

3.3.7 Conclusion

Symptom monitoring digital tools in oncology can be viewed as a potential breakthrough in the area of cancer care and health technology. In this scoping review, it was identified that since 2014 many platforms such as mobile apps, web portals as well as SMS-based systems have been designed to aid in the real-time reporting of treatment associated symptoms by the patient, according to standardised CTCAE measures. The instruments have proved to be highly usable in various patient groups and seem to enable earlier identification of adverse events.

Initial data indicate possible advantages in terms of preserving the quality of life, enhancing patient engagement, and simplifying clinical processes due to the ability to intervene immediately in reaction to extreme symptoms. However, most studies to date have been limited in scale, scope, and duration, and robust evidence of long-term clinical benefit is still emerging. There are still major issues, such as the integration into various healthcare systems and equitable access to all the patients.

In conclusion, digital symptom monitoring systems have been found to be viable and acceptable in the oncology environment and this suggests a future where digital platforms, which are aligned with CTCAE, can be a reality to be incorporated into the ongoing cancer care. Such tools can increase the effectiveness of supportive care and have a positive effect on clinical outcomes. They make it possible to report symptoms sooner to patients and provide clinicians with the opportunity to act on the information collected between visits. Future research, especially large-scale randomised trials, and implementation research in the real-life setting will be required to validate and operationalise the effect of such digital interventions fully.

Chapter 4

Developing a Clinical Chatbot for Structured Symptom Reporting in Oncology

The gaps found during the scoping review, provided in Chapter 3, where the use of conversational tools in monitoring symptoms in oncology is minimal, were used to inspire this project. Although a number of digital solutions are suggested, the majority of these tools are based on static and form-based ways of monitoring the symptoms. Conversational systems are still quite uncommon. Only a a single study was found that used a chatbot. However it was a rule-based SMS system with no sophisticated reasoning or dynamic communication.

To address these gaps, this project suggests the creation of a system consisting of a conversational chatbot that would provide patients, a way to report and monitor their symptoms, according to CTCAE guidelines, dynamically. The system's aim is to provide a more natural experience in reporting the symptoms, than the ones found in literature, while maintaining clinical rigor, by combining deterministic dialogue structure, ensuring protocol compliance, with LLM, for question generation, response evaluation, and symptom grading.

The modern TOD frameworks found in Section 2.5 were analyzed to provide the architectural direction. The fully end-to-end architectures were not considered suitable in high-stakes medical settings since they are not transparent, do not limit the reasoning path, and it is hard to validate internal decision-making. In contrast, systems like SGP-TOD and the Script-Based AI Therapist proved that combining rule-based structure with LLM enables both linguistic flexibility and control, making hybrid approaches more suitable for medical dialogue systems.

4.1 Part I: System Design

This first part presents the architectural principles of the system.

4.1.1 Overview

The system outlined in the current dissertation was inspired directly by the Script-Based AI Therapist [4]. A system that combines LLM with oriented dialogue structures, allowing patients to follow stipulated protocols.

The present system also adopts a modular, rule-guided approach aligned with LLM. Each symptom is modeled as a predetermined **path graph** in which, each node represents a clinical dimension in agreement with CTCAE requirements. The dialogue advances deterministically through these nodes, ensuring that all these dimensions are always covered. At each step, the LLM is used in a modular way: to generate context appropriate questions in natural language, to evaluate whether the patient's response satisfies the clinical objective and to produce summaries and justifications that support symptom grading.

This architecture separates language generation from dialogue control, which enables natural and flexible interactions with the patients, while preserving control of the conversation's direction.

The system is composed of different modules, with the main one being the **Dialogue Execution Module**, which manages the dialogue process, as it is in charge of walking the patient through the conversation. It's composed of three components:

1. **Node Controller**: navigates the symptom's path graph. It activates each node in the graph, which allows the conversation to flow.

2. **Question Generator (LLM)**: generates natural language questions based on the objective of the current node.

3. **Response Assessor (LLM)**: responsible for analyzing if the patient's response is sufficient to fulfill the node's objective or requires further clarification.

Additional system modules support other key tasks in the workflow:

- **Symptom Identifier**: maps additional patient-reported symptoms in free-text form to existing symptoms present in the CTCAE.

- **Node Summarisation Module**: produces a short clinical summary per node completed.

- **Grade Assigner**: assigns a CTCAE grade to each of the symptoms reported.
- **Grade Validator**: performs consistency check on the assigned grade.
- **Medical Report Generation**: provides a concise summary of the reported symptoms and aligns the all the session output in a structured medical report.

4.1.2 Symptom Modeling

All clinical relevant dimensions with respect to CTCAE grading were covered by mmodelingeach of the symptoms in the system as a path graph.

Symptom Modeling and Data Selection There are 837 adverse event terms defined in CTCAE v5.0 developed and maintained by the United States NCI. They represent a wide range of display forms, such as complaints that patients report, laboratory abnormalities, and clinical finding of healthcare professionals themselves. Grading and classification of every event can be based on one or more of these sources of information.

Nevertheless, some CTCAE terms cannot be applied in a digital environment, where the patient is responsible for reporting the symptom. A manual search of the entire CTCAE catalogue was carried out to select the subset of symptoms suitable to the structured and language-based reporting. Each AE term was individually evaluated according to two inclusion criteria:

- **Patient-observable:** The symptom should be self-observable and reportable, requiring neither clinical equipment nor clinical knowledge.
- **Language-expressible:** The symptom should be expressible in ordinary, non-technical language.

Symptoms that failed to meet either criterion were discounted. This screening produced **203** eligible symptoms, about **24%** of the CTCAE catalog.

Structured Symptom Representation

In the following section, the internal structure of mapping each CTCAE symptom into a set of clinically meaningful information nodes is described. This scheme unites official grading definitions and a detailed dissemination into discrete information nodes, which set up the systematic collection of clinically relevant information.

For each symptom, the following data elements where directly retrieved from the CTCTAE catalog:

- The official MedDRA code and SOC corresponding to the CTCAE term;
- The definition of the symtpom
- The full rubric of CTCAE grading, which include the textual definitions of each grade severity level (`Grade 1` to `Grade 4`). This definitions cover several clinical dimensions of the symptoms, like the intensity, the impact on daily tasks and self-care. *Grade 5*, which corresponds to death does not appear in the representation since it is inapplicable to patient self-reporting;

To transform each symptom to the system, every symptom is represented as:

$$S = (N, G, M)$$

where:

- $N = \{n_1, n_2, \ldots, n_k\}$ is an ordered list of information nodes, which defines the path graph, where each node, n_i, is defined as a tuple $(\texttt{label}, \texttt{description})$. The `label` representing the clinical aspect being assessed, such as "Impact on Self-Care", and the description providing the semantic basis for question generation.
- $G = \{g_1, g_2, g_3, g_4\}$ the official CTCAE grading criteria;
- $M = (MedDRA\ Code, MedDRA\ SOC, Definition)$ where the definition is what defines the symptom.

Though each node is statically characterized by its label and description, it is related, at run time, to a `state`, either `OPEN`, `CLOSED`, or `EXPAND`, that shows its status of evaluation in the ongoing dialogue process. This state is managed by the **Node Controller** as the system traverses the symptom-specific path graph.

In order to better visualize the inner composition of every symptom, Figure 4.1 illustrates how the symptom *Fatigue* is structured.

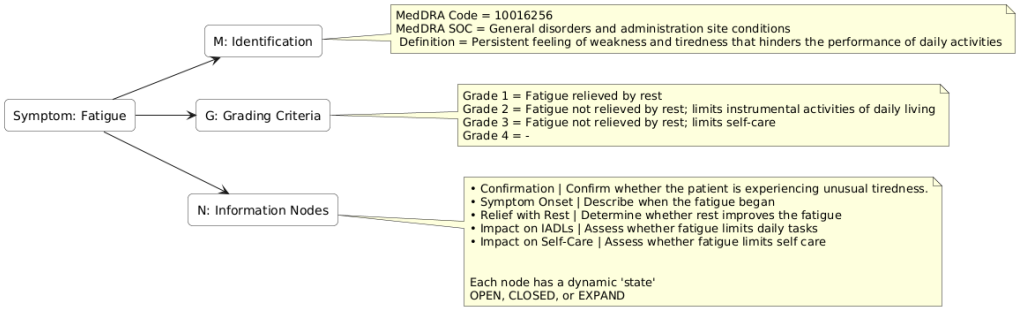


FIGURE 4.1: Structured Representation of the symptom Fatigue According to CTCAE Criteria.

The information nodes N are specific to symptoms and are defined manually and in line with a detailed decomposition of the CTCAE grading of each symptom.

The construction process of N followed a systematic methodology:

- **Grade analysis:** Grade definitions of the CTCAE (g_1 to g_4) were analyzed separately in order to determine the clinical findings that correspond to the individual levels of the grades (e.g. response to rest, effects on daily activities, symptom intensity).
- **Concept extraction:** The grading criteria was disaggregated into individual clinical concepts. Different grades may have one or several pertinent concepts as they are more or less complicated.
- **Node definition:** One node per concept was defined.

Figure 4.1 shows the symptom *Fatigue* along with the grading criteria G. These clinical requirements were broken down into specific conceptual units and related to specific information nodes. *Grade 1* is associated with fatigue that is relieved by rest, and as a result it gave rise to the creation of the node *Relief with rest* where the description of the symptom helps to infer whether the symptom gets relieved by rest or not. Grade 2 refers to Instrumental Activities of Daily Living (IADLs) limitations which led to the formation of the node *Impact on IADLs*. Grade 3 is characterized by limitations in self-care, prompting the addition of the node *Impact on Self-Care*.

Other nodes, however are not a direct outcome of the grading criteria but have significant functional or contextual roles. To give an example, the node *Confirmation* appears in each symptom as a technical prerequisite ensuring whether the symptom is present or not to continue with the assessment. In the same regard, the node *Symptom Onset* does not add an extra value to the CTCAE grading, but rather provides a good clinical context in terms of capturing the duration and temporal pattern of the symptom.

Protocol Assignment

The system includes pre-structured treatment protocols, each one linked with a curated set of symptoms. The lists of symptoms were retrieved from the official treatment documentation provided by the *Instituto Português Oncologia do Porto*, thus ensuring its conformity to the clinical monitoring requirements of each treatment protocol. Some of the protocols that are currently assessed in the system are:

- **CAPOX** (Capecitabine + Oxaliplatin)
- **FOLFOX** (Folinic Acid + Fluorouracil + Oxaliplatin)
- **FOLFIRI** (Folinic Acid + Fluorouracil + Irinotecan)

4.1.3 Dialogue Execution: Node Controller, Question Generator, and Response Assessor

The **Dialogue Execution** is responsible for the conversation flow. This module has three main components:

- **Node Controller**: a deterministic part which manages the navigation through the symptom path graph, triggering each clinical node, controlling the state of its resolution.
- **Question Generator**: a LLM based component which takes the clinical *description* of the node and generates a natural language question with the intent of retrieving the corresponding clinical information of the patient.
- **Response Assessor**: a LLM based component that assesses the answer of the patient and provides a structured decision on whether the information given meets the clinical goal of the node.

In each node n_i, the **Question Generator** takes the *description* of the node and generates a natural language question, q_i, that seeks to capture the clinical information that is represented by the node. The patient's response, r_i, is then evaluated by the **Response Assessor**, which returns a decision:

$$d_i = \texttt{Decision}(r_i, q_i) \in \{\texttt{PRUNE}, \texttt{EXPAND}\}$$

- `PRUNE`: the answer sufficiently covers the clinical aspect of a node and the **Node Controller** updates the graph by marking the node as `CLOSED`.

- `EXPAND`: the response is insufficient or ambiguous, and the **Node Controller** updates the state of the node as `EXPAND`. The **Question Generator** is responsible for rephrasing the question based of the node's original description and the gap found, to allow a clarification question to be formulated.

Every decision is followed by a justification produced by the **Response Assessor** that explains the rationale of the classification and, in the case of `EXPAND`, a short sentence providing the missing details, to provide to the **Question Generator** to reformulate the follow-up question. The system uses a redundancy filtering mechanism before asking the follow-up question to the patient, where sentence embeddings are calculated using a pretrained embedding model and cosine similarity between the candidate follow-up and each of the previous questions within the same node is computed, to avoid redundancy between the questions. The dialogue continues only when the node is marked as `CLOSED`.

This node-level decision process is summarised in Algorithm 1.

Algorithm 1 Node-Level Dialogue Execution

Require: Ordered list of information nodes $N = \{n_1, n_2, \ldots, n_k\}$
Ensure: Each node evaluated and marked as `CLOSED` when sufficient information is obtained

for each node $n_i \in N$ **do**
  Generate question $q_i \leftarrow \text{QuestionGenerator}(n_i)$
  Present q_i to the patient
  Receive response r_i
  Evaluate decision $d_i \leftarrow \text{ResponseAssessor}(n_i, r_i)$
  if $d_i =$ `PRUNE` **then**
    Mark n_i as `CLOSED`
    Proceed to the next node
  else if $d_i =$ `EXPAND` **then**
    Mark n_i as `EXPLORE`
    for each follow-up question $q_i' \in d_i$.`follow_ups` **do**
      if $q_i' \notin$ `PreviousQuestions`(n_i) **then**
        Present q_i' to the patient
        Receive updated response r_i
        Re-evaluate $d_i \leftarrow \text{ResponseAssessor}(n_i, r_i)$
        Repeat from Step 6
      end if
    end for
  end if
end for

Figure 4.2 illustrates the structured representation and dynamic evaluation of the symptom *Fatigue*. The nodes, represenitng the clinical dimension in the illustration are defined as shown in the representation of the symptom in Figure 4.1. In the process of the dialogue, every node is updated dynamically based on the response provided by the patient. The colour of a node indicates its `STATUS`: where green is `CLOSED`, red is `EXPAND` and yellow is `OPEN`.

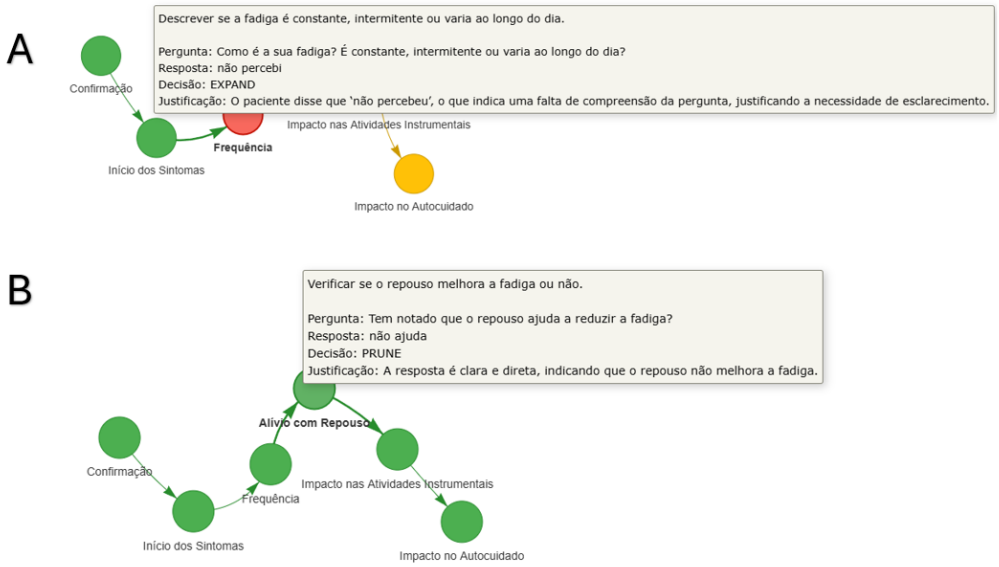


FIGURE 4.2: Symptom graph traversal for fatigue.

In Panel (A) for the node *Frequency*, the system queries: *"How is your fatigue? Is it constant, intermittent, or does it vary throughout the day?"*, to which the patient responds as *"I didn't understand"*, which in turn leads to an `EXPAND` decision with a justification indicating lack of comprehension. As a consequence the node is marked as `EXPAND`.

The node *Relief with Rest* can be seen in panel (B), with the question: *"Have you noticed whether rest helps reduce your fatigue?"*. The patient replies: *"It doesn't help"*. The response is considered sufficient, leading to a `PRUNE` decision. The node is then marked as `CLOSED`.

4.1.4 Symptom Identifier

The system has a specific module to support the reporting of the unexpected or additional symptoms that are not included in the treatment protocol of a patient.

The architecture follows a multi-stage design that combines semantic search with LLM-based reasoning. When the patient reports another symptom, the system tries to match it with one of the

symptoms in the list of CTCAE v5.0 symptoms, manually curated as explained in Section 4.1.2 using both the symptom definitions and their respective MedDRA SOC.

The identification pipeline starts with a coarse-grained classification narrowing down the search space to a specific SOC, and subsequent fine-grained semantic similarity search within the narrowed group to pick the most relevant symptom. In the event that the system cannot be confident that it found an appropriate match, a fallback mechanism is initiated: rather than attempting to assign an incorrect classification, the system instanciates a generic path graph, with pre-defined, information nodes.

These generic nodes will be able to capture baseline clinical data applicable to any symptom which includes:

- **Symptom Onset**: to determine when the symptom began.
- **Frequency**: to understand how often the symptom occurs.
- **Impact on Daily Life**: to assess whether the symptom interferes with instrumental activities of daily living.
- **Impact on Self-Care**: to check if the symptom limits self-care.

4.1.5 Node Summarization

This module transforms the interaction history of each node into a concise clinical statement after all the nodes are marked as `CLOSED`.

In summarization, the following is considered:

- the node's clinical intent;
- the full dialogue history within that node (initial and follow-up questions + answers).

The output is a short sentence, summarizing the answer given by the patient to the node, retaining the important information in relation to the node's `description`. All these summaries are represented as s_i with i being the index of the associated information node n_i.

4.1.6 Grade Assigner

The grading module operates only after all node-level summaries s_i have been generated by the **Node Summarization** module. The set of summaries for a given symptom S is represented as:

$$H_S = \{s_1, s_2, \ldots, s_k\}$$

The LLM reasons over the collected data and the official CTCAE grading rubric of the symptom, which is given as: $G = \{g_1, g_2, g_3, g_4\}$, to arrive at the final grade.

Instead of providing the whole interaction history for the symptom, the choice of using s_i relies on the fact that the LLM would not benefit of having long text-input with possibly irrelevant dialogue history.

For example, for the following node-level summaries s_i:

- Relief with rest: "The patient reports moderate fatigue not relieved with rest."
- Impact on instrumental activities: "Needs to rest during daily chores but manages to complete them."
- Impact on self-care: "No difficulty with personal hygiene or dressing."

Given the CTCAE rubric for fatigue:

- **Grade 1**: Fatigue relieved by rest.
- **Grade 2**: Fatigue not relieved by rest; limits instrumental ADL.
- **Grade 3**: Fatigue severely limits self-care ADL.

The model is expected to reason with the different grades and the summaries to conclude that the corresponding grade, in this case corresponds to *Grade 2*.

Other solutions were also discussed in the design phase such as fine-tuning or LoRA-based adaptation of a language model to directly classify symptom grades. Nevertheless, this methodology was soon disregarded because of the total lack of annotated data on patient-authored symptom descriptions that are tied to CTCAE levels. Creating such dataset would involve massive expert annotation, which is associated with data privacy issues as well as significant resource requirements. Prompt-based reasoning was thus chosen as more viable option.

4.1.7 Grade Validator

This module is responsible for ensuring that the symptom specific assigned grade, $\hat{g}$, is consistent with both:

- The grading criteria G;
- The node-level summaries H_S.

The validation mechanism, impelemented using a LLM, produces a structured output that includes not only the validation status, but also a justification for the decision:

$$\texttt{ValidateGrade}(\hat{g},\, H_S,\, G) \Rightarrow \begin{cases} \texttt{CONFIRMED}, & \text{grade is accepted, with justification} \\ \texttt{REJECTED}, & \text{grade is flagged for review, with justification} \end{cases}$$

This validation module helps in checking whether each grade assigned is in line with the information gathered in the dialogue and the grading criteria. By explicitly comparing the $\hat{g}$ against the set of (H_S) and (G), the system can detect discrepancies or unsupported classifications.

In conflict scenarios, the grade is flagged to review. This module enhances the transparency of the system since all the inputs and decisions made in the grading process are retained and traceable.

It acts as a second line of defense to determine whether the grade first set by the grading module is clinically consistent with the data obtained. It does not ensure the validation result to be always accurate, but it gives an additional chance to identify possible inconsistencies or unreasonable grading decisions.

4.1.8 Medical Report

Once the grading and validation of each symptom is completed, this module is responsible for generating a medical report, which includes:

- the name of the symptom;
- the final grade (from the grading module);
- the validation status (`CONFIRMED` or `REJECTED`);
- a clinical summary (one sentence or two sentendes) capturing the patient's overall symptom complaints;
- a set of bullet points highlighting key clinical facts;
- the justification for the validation decision.

The system generates the narrative component of the report (summary and bullet points), using a LLM as well as incorporates structured outputs of the grading and validation modules for each of the symptoms. The node-level summaries serve as the basis for generating concise, symptom-level descriptions and bullet points. The grade, validation result and the respective justification are retrieved and added to achieve complete traceability.

The report provides an interpretable and clinically meaningful overview of the session as it brings together portions of data that are contained in separate modules providing a clear picture of the session and allowing monitoring downstream clinical decision-making and auditability.

4.2 Part II: Technical Implementation

This section presents the technical implementation of the system. While Part I focused on the architectural principles and modular design, this section details how the components were executed in practice, including runtime configuration, large language model integration, data management, local deployment and data privacy.

4.2.1 Overview and Execution Pipeline

Upon run-time, the system initiates by loading the patient's protocol (e.g., FOLFOX), which contains a list of symptoms that are supposed to be evaluated (e.g., fatigue, nausea). Each of these symptoms is associated with a predetermined path graph, defined manually consisting of clinically relevant information nodes (see Section 4.1.2). Although the full graph structure is defined statically in JSON, nodes are instantiated dynamically at runtime by the **Node Controller**, which loads and activates each node sequentially as the dialogue progresses.

The first interaction with the patient occurs in the **Dialogue Execution** module, when the description of the first node is passed to the **Question Generator** component. This component will then generate, using a LLM, a natural question that is presented to the patient, aiming to infer the patient's experience within the clinical dimension of the node. When the patient provides an answer, the **Response Assessor** is then used to evaluate whether the information given is sufficient to fulfill the clinical dimension of the node or if clarification is required. If so, the LLM provides the missing detail, which is used, aligned with the initial node's description, to prompt the **Question Assessor** to generate a follow-up question. This process happens iteratively until the response is satisfactory, which is when the **Node Controller** closes the node. This guarantees that each dimension of the symptom is evaluated, having the desired information for CTCAE grading.

Once all nodes are traversed, the **Node Summarization** module generates a set of structured summaries (one per node), which will be used to infer the severity grade, by using the **Grade Assigner**. This module, using the LLM, is instructed to match the summaries of each node to the CTCAE rubric.

When all protocol symptoms have been processed, the system prompts the patient to report any additional symptoms in free-text form. Using the **Symptpom Identifier**, the system either retrieves an identified symptom from the manually curated list along with its nodes or a fallback structure is employed, and the generic nodes to collect relevant clinical information are instanciated.

In case no other symptoms are to be reported, the **Grade Validator** is run as a last step prior to the generation of reports. If any of the symptoms, which were additionally reported by patients, but weren't identified, are present, they don't undergo the process of grading or validation, as there's no grading criteria predefined for that symptom.

Once all of the validation is performed, the medical report is then generated, with a concise summary, and bullet points for each symptom, as well as the grade and its validation.

Handling of Uncertainty and Indeterminate Responses

In some cases, where patients may be unable to provide an answer to a question (e.g., "I don't know", "I can't remember"). The **Response Assessor** is designed to treat such uncertain responses as acceptable. In these cases, it issues a `PRUNE` decision with a justification explaining that no further clarification is feasible under the circumstances.

The system acknowledges that uncertainty and memory limitations are inherent to patient self-reporting. Excessive probing in such cases would not enhance data quality and may instead lead to patient fatigue or frustration.

4.2.2 Node Instantiation

All the symptoms have predetermined sets of information nodes, which are stored in structured JSON files.The ordered list of nodes of a certain symptom is presented in each JSON file.

As the conversation continues the system loads and instantiates each node dynamically creating a symptom-specific *path graph*.

During runtime, each node has the form of a Python object of the `Node` class, which is defined with `Pydantic` to achieve structure and type safety. This selection allows a smooth and secure

transformation of the predetermined node data, initially saved as JSON, into fully validated Python objects, making the dialogue running chain consistent. A node consists of an identifier, a `label`, a `description` and a `state` (`OPEN`, `EXPLORE`, or `CLOSED`) which keeps track of its progress in the dialogue.

The `label` and `description` are extracted from the JSON file. The `state` tracks the node's progress in the dialogue depending on the outcome of the **Response Assessor**.

The full path graph is thus a sequential, dynamically constructed list of such node objects, which is supervised by the `Node Controller` module.

4.2.3 Symptom Identifier Pipeline: Embeddings and LLM Verification

A hybrid RAG pipeline was implemented in case the patient wishes to report additional symptoms. It is not a usual RAG application. The solution consists of semantic similarity search (sentence embedding) and reasoning (LLM) to ensure proper identification of the symptoms.

The identification of the additional symptom is performed over the manually curated list, where they are organized by its SOC (e.g., "Gastrointestinal disorders" or "Nervous system disorders"). To allow similarity-based retrieval, semantic embeddings are calculated through the `nomic-embed-text` model.

Embedding Index Construction

Two FAISS-based semantic indices were created:

- **SOC Embedding Index:** The MedDRA SOC names were appended with their respective symptoms names. Thus, generating compound strings, which represent the semantic association between individual SOC and its symptoms.
- **Symptom Embedding Index:** The CTCAE symptoms were all embedded through concatenation of the name, the clinical definition, and the grading criteria (Grades 1-4).

Pipeline Execution

The end-to-end pipeline consists of the following steps, which put together semantic retrieval with structured LLM verification and fallback reasoning:

1. **FAISS Embedding Search:** A top-k = 4 similarity search is applied to symptom embedding index to retrieve the most semantically relevant candidate symptoms for the free-form input of the patient.

2. **LLM-Based Verification:** A large language model is used to verify each candidate by means of a structured prompt. The LLM gives an output of `YES` only when the given symptom is clinically compatible with the description.

3. **Fallback: SOC-Based LLM Classification:**

   - The LLM is prompted with a shorter list of MedDRA SOC that have not been tried.
   - It picks the most suitable SOC according to the free-text description of the patient.
   - The symptoms of that SOC are passed to the LLM which identifies the most likely candidate.
   - The selected symptom is then verified again through the same `YES/NO` process.
   - This loop repeats and updates the list of available SOCs, until a verified symptom is found, or all SOCs are done.

4. **Final Decision:** If no match is verified, the system concludes that no suitable candidate was found and returns `"not found"` as the outcome.

4.2.4 Evaluation of Symptom Retrieval System

A large language model was used to generate a tailored collection of 450 free-text patient descriptions in order to aid the analysis of symptom retrieval and recognition. These descriptions were obtained in order to represent the variability of the natural language and to mimic realistic oncological symptoms descriptions. All entries were then manually checked to ensure clinical plausibility, linguistic variation, and congruency to expected symptom terms.

Three versions of the system were tested:

- **Embedding-Only Baseline:** Similarity search over pre-computed symptom embeddings using FAISS, without any LLM reasoning during retrieval.

- **LLM-Only (SOC + Symptom Selection):** A pipeline relying solely on LLM-based reasoning to first narrow down to a System Organ Class (SOC), and then to select a symptom, without using semantic embeddings.

- **Hybrid Ensemble (Final Pipeline):** The multi-stage pipeline described, which combines embedding retrieval, SOC-based narrowing, and LLM-based verification.

Performance Results

TABLE 4.1: Accuracy comparison across retrieval strategies

Pipeline Version	**Accuracy (%)**
FAISS Embedding-Only	25.06
LLM Only	47.20
Hybrid Ensemble (Final Pipeline)	57.27

The hybrid pipeline reached the best test accuracy of 57.27% thus validating the use of integrating semantic similarity retrieval with LLM-based reasoning in symptom identification.

It was through empirical testing that the choice was made to make the symptom identification process based on SOC. Experiments done on earlier versions without using SOC, just using all the available symptoms, tended to give false positives as the system tended to equate certain similar symptoms with respect to meaning but clinically very different (e.g. abdominal pain vs. chest pain). Additionally, the retrieving of the symptom was also longer.

Regardless of the retrieval path, embeddings or fallback, all candidate symptoms undergo a final LLM verification step using the patient's original description, decreasing the chance of false positives.

4.2.5 LLM Integration

All LLM inference is performed locally using `Ollama`.

A number of open-source models were benchmarked to evaluate their applicability to the system needs, which requires structured output, natural phrasing in portuguese from Europe and good reasoning skills. The candidates tested were:

- `llama3.1:8b`, `llama3.2:latest`
- `deepseek-r1:latest`, `deepseek-r1`
- `qwen2.5:14b`, `qwen2.5:7b`

The model `qwen2.5:7b` was ultimately selected for all core reasoning tasks including question generation, node decision-making, grading and validation. It was shown to have the best tradeoffs in the quality of generation, run-time performance and structure output consistency. Specifically, the 7B version consistently produced valid and structured JSON outputs, responded within acceptable latency thresholds (<2 seconds) and handled Portuguese reliably.

Only open-source LLM were considered, as they can be deployed locally without the use of an API, thus they are not dependent on external servers.

Prompting Framework and Configuration

The entirety of language model interactions was executed through the `LangChain` framework, which features modular templates of prompts, and temperature parameter control per module, which determines how random or deterministic the output of the model can be. The lower the value of the temperatures (e.g., 0.0), the more deterministic the model will be, and it will promote similar and reproducible results. With increased values (e.g. 0.7 or greater), there is added variability and more diverse answers can be generated. This parameter is of great importance in this system, as each module require different behaviours:

- **Grading and Validation:** low temperature (`0.1`) to ensure reproducibility.
- **Question Generation:** high temperature (`0.7`) for more natural, conversational phrasing.
- **Summarisation and Reporting:** moderate temperature (`0.3`) to allow stylistic variation while preserving accuracy.

Structured Output Parsing

Due to the use of open-source models used locally the system did not benefit from function calling or strict output schema enforcement. As a result, a dedicated post-processing and validation pipeline was implemented to extract structured outputs from raw model outputs.

1. **Direct Parsing Attempt:** The raw output of the model is firstly parsed in raw shape as JSON. In case of success, no subsequent measures are needed.
2. **Regex Pre-Extraction:** If parsing fails, then regular expressions may be used to extract the most probable JSON block out of the text, compensating the possible extraneous explanations, formatting artifacts, or incomplete bracket closure which sometimes may be introduced by generative models.

3. **Pydantic Schema Validation:** That data is then validated against a `Pydantic` schema, that is specific to the particular structured output needed by the task.

4. **Retry Mechanism:** In case any of the preceding steps fail, the system will automatically re-populate model with identical data and this can be done several times (4 retries) to achieve a compliant structured answer.

5. **Failure Mechanism:** In worst case scenarios when all parsing and recovery fails, the system implements conservative fall back mechanisms in order to achieve stability of the dialog and clinical safety.

   Regarding the node-level decision modules, a fallback `EXPAND` is issued when the node should not be closed immediately, but the system should rather keep on collecting information of the patient.

   In modules like grading, validating, summaries and bullet point extraction, safe place holders are shown (e.g. summary unavailable, grade: not available), so that no clinical conclusions are made based on a missing valid structured data source.

   In free-form symptom description cases, in the case of any parsing failure in the process of candidate extraction, the instantiation of generic information nodes is triggered directly, so that the structured collection of data is also an option.

Lexical Normalisation Layer for PT-PT Compliance

A lexical post-processing layer has been deployed in order to guarantee linguistic adherence to european portuguese. All the outputs generated by the LLM are intercepted prior to presentation, and are the subject of specific replacements of frequent brazilian- portuguese derived lexical items.

It is a rule-based system with a manually created dictionary of replacements which includes many common regionalisms and clinical terms. These could be seen in the replacement of *"abdômen"* by *"abdómen"*, and *"banheiro"* by *"casa de banho"*.

4.2.6 Deployment and Runtime Environment

The whole system is implemented on a lightweight and modular stack that is focused on local execution and data privacy along with ease of deployment. It is implemented as a web application using the Flask framework, with SQLite for structured data persistence, and containerised via Docker for portability and reproducibility.

Flask and Database Backend

The application logic is realized with Flask, a lightweight Python web framework which is suitable to rapid development and flexible deployment in web systems. Flask routes are used to control user interactions at every user level (patients, clinicians and administrators). The session state and conversation history is maintained in memory at run time but regularly written back into the database.

The permanent storage of data is implemented with local **SQLite** database.

Containerisation and Local Execution

The system is packaged in a single Docker container built on the `python:3.10-slim` base image. This container combines all the in order to run the application as explained in Table 4.2.

TABLE 4.2: Main functional components within the application container

Component	**Role within the Container**
`Gunicorn`	Serves the Flask application with multiple worker processes, enabling concurrent sessions
`Ollama`	Executes the local LLM and embedding models with GPU acceleration
`SQLite`	For storing data
`Cron`	Handles scheduled tasks such as automated backups and log rotation

Startup Sequence and Hosting Infrastructure

The system is hosted on a dedicated Ubuntu 24.04.2 LTS production server with GPU support enabled via the NVIDIA Container Toolkit.

The startup process involves the following two steps to ensure all components are operational before any user session begins:

1. **Model Services:** Starts `Ollama serve` in the background and loads both `qwen2.5:7b` and `nomic-embed-text` into GPU memory.

2. **Backend Launch:** Runs the Flask application via `Gunicorn` to expose the web interface.

The container exposes the Flask server internally on port `5000`, externally mapped to `8000`.

Persistent Storage and Operational Logging

Two persistent Docker volumes are mounted:

- **Database Volume:** This stores the complete SQLite database file that the system has integrated.
- **Logs Volume:** Gathers multiple-level time-stamped logs that logs detailed events and traces of the system operations. This includes:
  - Server errors and flask backend runtime errors.
  - Failures in routes when interacting with the different users.
  - Failure traces and inference responses of the LLM.
  - Ollama server diagnostics and LLM runtime messages.
  - Errors on transcription services and failure in voice synthesis (from Edge TTS and Whisper models).

This ensures the safety of the data associated with patients and all other events of the operations as they are stored securely in case the container restarts, and enables detailed debugging of inference processes and route handling.

4.2.7 Database Architecture and Data Persistence

Based on the simplicity and native file-based persistence support, SQLite was selected to be used as the database.

Relational Structure

Every **patient** is singularly mapped with the one physician. This forms a firm one-to-many relationship.

Every session which is stored under the table `sessions` is specific to a patient.

The table **session _symptoms** is also connected directly to sessions, and it will keep information about each symptom that was assessed in the conversation with the chatbot. The entire trace of dialogue of each symptom is kept in the field `info_json`, which includes the complete dialog history for the symptom, grade assigned, node-level interaction history (including the decision/justification/ given by the LLM), structure of the executed dialogue path, the validation of the grade given by the LLM, and the generated clinical summary including bullet points.

The `patient_protocols` table contains the assigned treatment protocol, which links a particular patient with a protocol.

Automatically assigned grades which are subsequently changed manually are saved in the `grade_overrides` table, along with a justification.

Administrative users are stored separately from clinical users. Authentication credentials are securely hashed and stored in the `password_hash` field within all the users.

The Figure 4.3 represents the schema of the database.

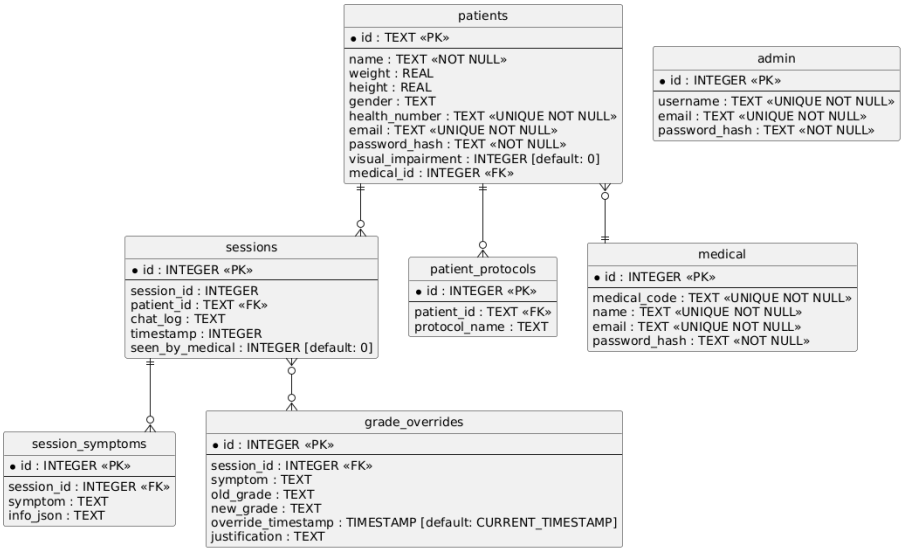


FIGURE 4.3: Relational database schema.

Persistence Mechanisms The system uses a form of incremental persistence in an active session: everything on dialogue state, symptom history and decisions is written to the database after each patient input. This guarantees that all information relating to the reporting of the symptoms is safely saved in case of system failure.

4.2.8 Data Protection, Security and Clinical Safety

The system was developed based on main principles of data protection that have been provided by General Data Protection Regulation (GDPR) [130] mainly focusing on privacy, access controls, auditability, and clinical safety. Although there is no regulatory certification, the following mechanisms are used to assure proper processing of sensitive clinical data within scope of GDPR, including Article 5(1)(f) (integrity and confidentiality), Article 5(2) (accountability), and Article 32(1) (security of processing).

Privacy and Access Control The Personal health information (PHI) such as patient identity, clinical history, and the information on symptoms are visible to users who passed the authentication and authorization procedure based on their role (patient, clinician, administrator). Hashed passwords using `bcrypt` are stored so that the risks to credential exposure are limited.

Local Data Storage The entire data is saved locally on a SQLite database wrapped within a Docker container. LLM use is local, so there is no use of API. Web Speech API has however been employed on the frontend interface which is a browser-native API for speech recognition and speech synthesis. This element can internally use third-party services (based on the browser and device settings of the user), but the backend application won't store or share any audio record of any kind.

Auditability and System Logging A logging subsystem stores the timestamps of operational events within the system. This allows traceability of the system's execution and behaviour .

Clinical Safety with Human in the Loop Oversight Any symptom classification that the LLM produces is validated in a two-tier process. Firstly, the system uses consistency checks through re-analysis with the LLM of the grading results. Then, clinicians can check and validate all classifications through the medical review dashboard and have access to the LLM justifications and session history. Any grade override is justified by the clinicician and stored for future auditing and accountability.

Session Timeout due to Inactivity As a counter measure against unauthorized access within unattended sessions, an inactivity timeout mechanism has been incorporated. The system will automatically log out a user and send him to the authentication page in case of the absence of user interaction (with a keyboard, mouse, or touch) in 8 minutes. This guarantees that sensitive data concerning remains secure.

4.2.9 User Interfaces and Interaction Design

This section shows the graphical user interfaces that have been designed and the different functions integrated. The system has three main parts that consist of patient, medical and administrator interface. Flask is used as the backend organization of all the components, and frontend

logic was developed with the help of HTML, Bootstrap, CSS and JavaScript. All the patients that are mentioned are ficticious.

Authentication and Access Control

Users are required to undertake a secure log in before they can access the patient, medical, or administrative interfaces. Every user has to use valid credentials that are a combination of unique identifier (the national health number in the case of patients, predetermined identifiers for clinicians and administrators), and a password. The login screen is shown in Figure 4.4. Since the login process is the same in any user role, only the medical view is shown.



FIGURE 4.4: Log in screen.

The system provides password reset function, through the link "Esqueceu-se da password?" to aid account recovery. This process is identical for all user roles and consists of the following steps, as illustrated in Figure 4.5.

- **Step 1:** After clicking in the link, the user will then be asked to type in their identifier;
- **Step 2:** A code will be sent to user's email, which has a 6 digit verification code, valid for 5 min;
- **Step 3:** The user has to enter the code to the interface;

- **Step 4:** After the code verification, the user can change the password and is then redirected to the login screen.

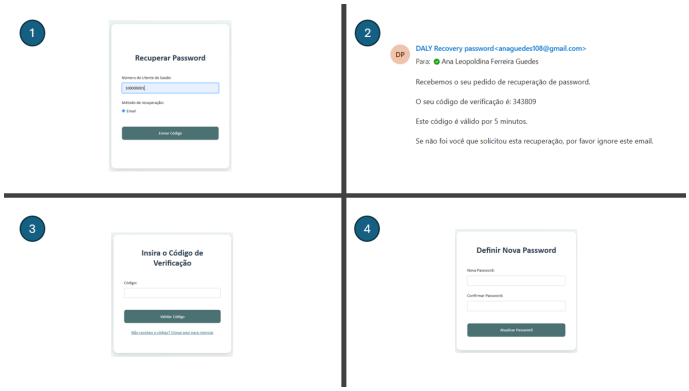


FIGURE 4.5: Password recovery process: (1) request with identifier, (2) email with 6-digit code, (3) code verification screen, (4) password reset form.

Users are only redirected to their respective interfaces (patient, clinician, or administrator) after a successful authentication and can then communicate with the system.

Patient Interface

The patient interface was designed to support a natural and accessible dialogue experience, enabling symptom reporting through both text and voice.

The design of the interface is based on the classic structure of a conversational chatbot with alternating system inquiries followed by patient answers as they occur in natural dialogue. Patients are only able to enter one response at a time, and only after the system has issued the following question.

Besides the primary conversation flow, the chat interface has a floating action button in a corner of the screen. By clicking on it, a menu will be opened that allows access to standard actions:

- **Create new chat**: instantly ends the current session and opens a new one;
- **Historic**: opens a modal window history of sessions, where it's possible to see past interactions;
- **Logout**: logs the patient out of the system.

This interface can be seen in Figure 4.6.

Voice Interaction Flow To make the system more accessible and natural to communicate with, it has an integrated voice interaction functionality. This aspect is adopted through Web Speech API and enables the patients to interact with the chatbot by responding verbally instead of using a text response. For patients that are marked with visual dificulties, the session starts with the voice mode activated. It only works when supported by the user's device and browser.

When voice mode is activated, shown by element 2 in Figure 4.6, the system follows a structured loop:

1. The question on hand is read by the system through speech synthesis;
2. Once the utterance completes, it automatically switches to speech recognition and waits to hear a response;
3. When the patient speaks, the reply is transcribed and processed just like it has been typed;
4. In the event that no response is given or detected, the system restates the question maximum 4 times before returning to typing mode;

Also, the system gives clear auditory and visual feedback to indicate the voice mode, by, when enabled, anouncing "voice mode activated", and shows on the lower corner of the screen a message informing the user that he can now speak, as illustrated by the element 3 in Figure 4.6. Likewise, turning off the voice mode does the same but announces "voice mode deactivated" and removes the visual message.

An example of the interface during active voice mode is shown in Figure 4.6.



FIGURE 4.6: Patient interface: symptom reporting session. (1) Menu of actions; (2) voice mode button; (3) voice mode status.

Session History Access Patients also have access to a session history interface which displays a chronological log of previous sessions, and upon selecting one, the history is shown. This allows users to revisit past answers. The following Figure 4.7 represents the session history (A) and chat history (B).

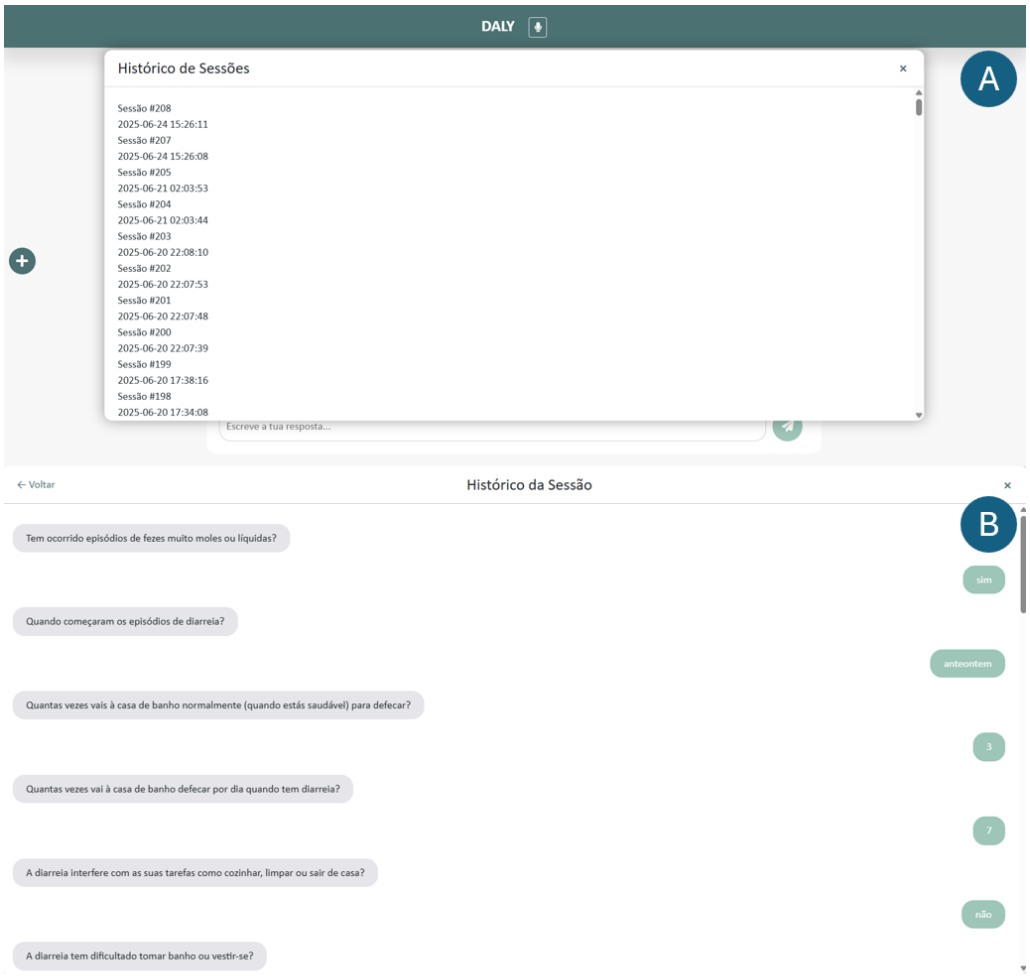


FIGURE 4.7: Patient interface: session history.

Post-Session Redirect and Main Menu Interface After completing a session, the patient is automatically redirected to the main menu interface. This interface provides access to the same features of the floating action button in the patient's chat interface.

Medical Interface

The medical interface provides a dashboard to go through the patient sessions and confirm the symptom grading. When they log in, the healthcare professionals are shown a list of all patients.

A visual alert badge is shown on each patient card when any sessions are unreviewed, with the number of sessions that lack review, and a second alert displayed when the grade is >2. The interface can be observed in Figure 4.8.

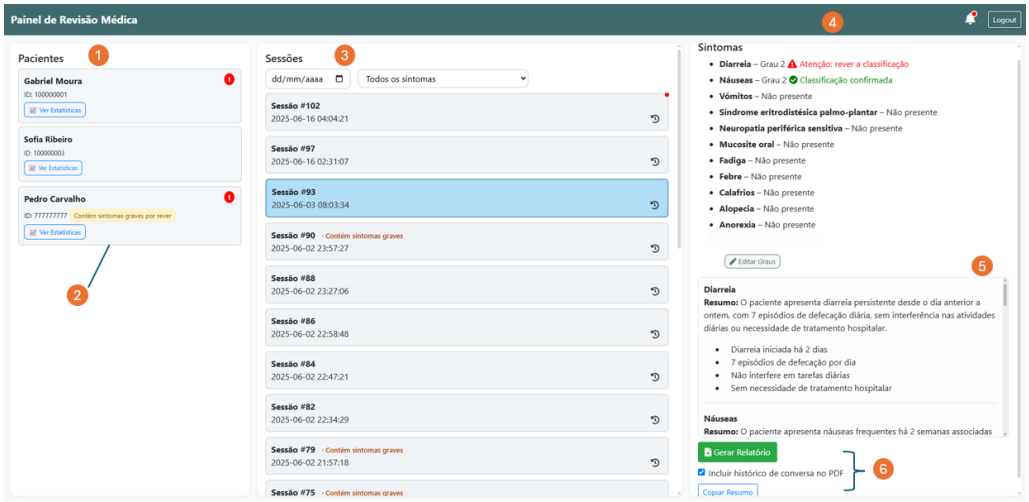


FIGURE 4.8: Medical interface. (1) Patient panel with badges where sessions are still unreviwed. (2) Patient card with a session still unreviwed and with severe symptoms. (3) Sesssions panel with date and symptoms filter. Session 102 presents a badge meaning it is still unreviwed. (4) Symptoms panel with all of the symptoms tested during the dialog with the patient, along with the validation results. The button "Editar Graus" enables the grade's override. (5) Summary and bullet points of the symptoms. (6) This section enables the exportation of the medical report including the PDF and also enables the option to copy to clipboard.

On the top of the page is a red notification bell, as seen in Figure 4.8 that highlights new or pending sessions. Clicking on it a drop down menu appears, consisting of the session information and the symptoms.

The following Figure 4.9 showcases the notification bell.

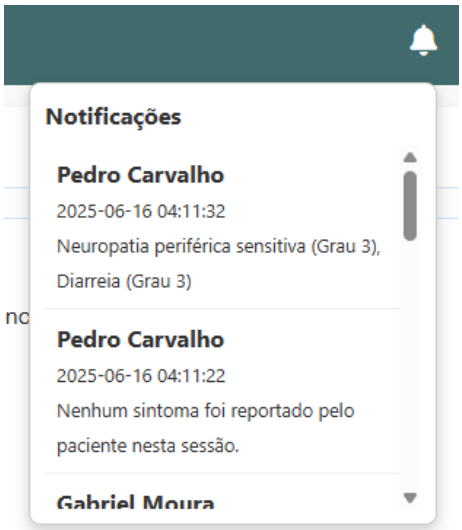


FIGURE 4.9: Notification bell.

Session Navigation and Symptom Overview When a session is chosen, a structured summary of the all reported symptoms appears in the right panel, shown in Figure 4.8(4). Every symptom consists of:

- The CTCAE grade assigned by the LLM;
- A summary of the conversation;
- A list of bullet points;

Clinicians have access to the full dialogue history rendered in the same format used in the patient interface.

There is an edit button through which they can automatically override the suggested grade and a reason as to why it is being changed. This is shown in Figure 4.10.



FIGURE 4.10: Editing the grade and justification provider.

Clinical Analytics View The analytics panel is also involved in the dashboard, through which clinicians have an overview of statistical symptom frequency and grade distribution. They are shown in form of charts which can be seen in the Figure 4.11. This includes:

- **Bar chart**: frequency of each grade for the selected symptom;
- **Line chart**: symptom severity evolution across sessions.

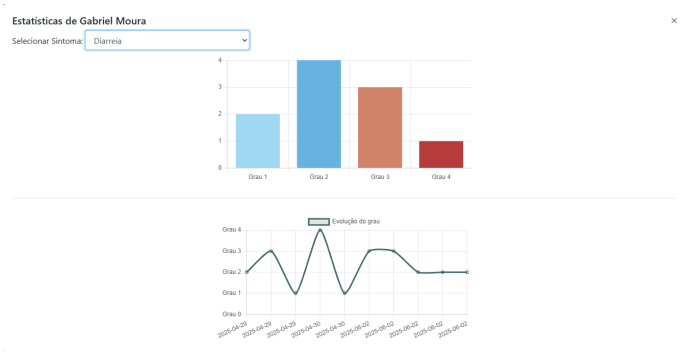


FIGURE 4.11: Bar and line charts.

Structured Summary and Export Options For each symptom, the summary includes the final CTCAE grade, a concise summary, bullet-pointed facts extracted from the dialogue, the LLM validation status, and an explanatory justification for the grade validation.

Whenever the status is `REJECTED` a warning icon appears next to the symptom, prompting the clinician to review, and when the status is `CONFIRMED` a validating icon also appears, as shown in Figure 4.8.

Clinicians can export this structured information in two ways:

- **A report exported in PDF**:
- **Clipboard Copy:** A modal interface that allows copying the report via two formats:
  - *Plain text*: readable information for direct review;
  - *JSON*: structured data format suitable for integration with external systems.

An example of the medical report is shown in the Appendix B, and the copy functionality is illustrated in the Figure 4.12.

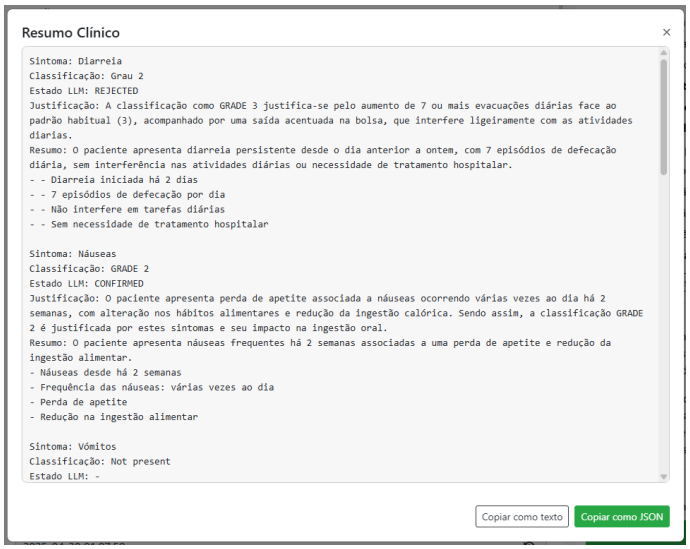


FIGURE 4.12: Copy to clipboard functionality.

Administrator Interface

The administrator interface allows complete access to the patient's and medical's management, as shown in Figure 4.13.

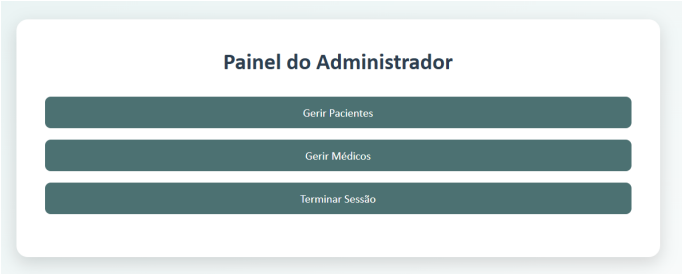


FIGURE 4.13: Administrator panel.

After selecting one of the options, the administrator is present with list of all registered users. After that the available functionalities include:

- **Registration:** Create new patients and physicians.
- **Editing and Deletion:** Edit or delete patients and physicians by clicking in the user.
- **Search and Filtering:** Search and filter patients and physicians by name, health number (for patients), medical code (for physicians) or email.
- **Physician and protocol Assignment:** Associate the physician and protocol to the patient when creating a patient.

The patient's and physician's management interface is alike, therefore only the patient's management interface will be displayed in Figure 4.14.

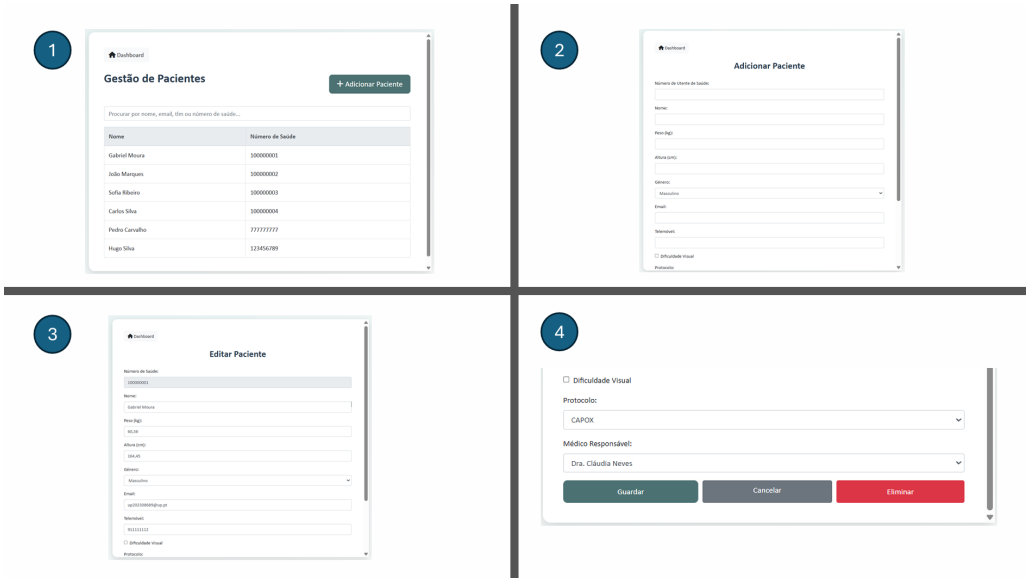


FIGURE 4.14: Patient's management.interface. (1) List of all the patients, including a search bar and an add patient button. (2) Patients' registration form, where the demographic and also the clinical information is entered. It's in this process that the physician and protocol are assigned. (3) Patients' edit form. (4) Continuation of the edit form with the option to delete de user.

Chapter 5

Results and Discussion

A clinical assessment was conducted with an oncologist, to assess the system efficacy and accuracy.

5.1 Evaluation Methods

Evaluation Procedure

The assessment was performed with one oncologist (female, 37 years old) working in the Unidade Local de Saúde do Alto Ave, having 10 years of clinical practice. The participant self-evaluated her digital skill as moderate and had no prior exposure to the system.

The assessment included:

- For the patient interface:
  - Performing a dialogue with the chatbot, simulating a patient interview.
  - Assessing the free-text symptom recognition by reporting additional symptoms.
  - Viewing the session history.
- For the medical interface
  - Grading the symptoms assessed during the dialog.
  - Reviewing the medical report.
  - Assess the static charts.

The Appendix A provides the full question set for each of the instruments as well as for the qualitative questions.

NASA-TLX

The perceived workload was measured by the use of National Aeronautics and Space Administration (NASA)-Task Load Index (TLX). The Physical Demand dimension was avoided because of virtual environment. The participant rated the following dimensions on a scale of 1-10, where 1 is strongly disagree and 10 is strongly agree.

1. Mental Demand
2. Temporal Demand
3. Effort
4. Frustration
5. Performance

The Performance dimension was reformulated to fit each context of the assessment: in the medical interface, it expressed how well the interface assisted the physician in making decisions, whereas, during the patient interface, it denoted the clinical data gathered during the dialog.

The items were **positively framed**. Therefore, instead of asking "How mentally demanding was the task?", the question was rephrased as "Was clinical reasoning easy during the interview?". Because of that higher values indicate a more positive experience.

To stay true to the original NASA-TLX Score, each item was **inverted** before calculation using:

$$\text{Inverted Score}_i = 11 - \text{Response}_i$$

All five inverted scores were scaled to a 0–100 range and averaged using:

$$\text{NASA-TLX Score} = \frac{1}{5}\sum_{i=1}^{5}(10 \times \text{Inverted Score}_i)$$

The five dimensions were then scaled to a 0-100 range and averaged:

$$\text{NASA-TLX Score} = \frac{1}{5} \times 10 \times \sum_{i=1}^{5}\text{Dimension}_i$$

System Usability Scale

SUS assessed usability across 10 statements ranging from 1 (strongly disagree) to 5 (strongly agree). Scores were adjusted as follows:

- Odd-numbered items: subtract 1 from the score.
- Even-numbered items: subtract the score from 5.

The total was multiplied by 2.5 to yield a score from 0 to 100:

$$\text{SUS Score} = 2.5 \times \left(\sum_{i \in \{1,3,5,7,9\}} (SUS_i - 1) + \sum_{j \in \{2,4,6,8,10\}} (5 - SUS_j) \right)$$

Questionnaire for User Interaction Satisfaction

QUIS measured satisfaction across 10 items rated from 1 (strongly disagree) to 9 (strongly agree). The average score was calculated as:

$$\text{QUIS Score} = \frac{1}{10} \sum_{i=1}^{10} \text{Question}_i$$

Higher scores are an indicator of better usability.

Additional Specific Questions

Besides the generic evaluation tools, there was also a short list of targeted questions that helped collect more dedicated feedback on the system functionality and integration in clinical practice. The scale ranged from 1 (strongly disagree) to 5 (strongly agree).

Appendix A is the list of all of the questions posed to the physician.

5.2 Evaluation Results

Symptom Grading and Additional Symptom Assessment

The oncologist fully agreed on all the severity grades determined by the system. Also, the system was able to recognize and process free-text symptoms proposed during the session,, which were “urinary tract pain” and “laryngopharyngeal dysesthesia”, demonstrating robustness in both protocol symptoms and additional symptoms.

In the Table 5.1 are presented the symptoms that were evaluated during the dialogue.

TABLE 5.1: Grades assigned by the system and confirmed by the oncologist

Symptom	**System Grades**	**Oncologist Grades**
Nausea	3	3
Oral Mucositis	3	3
Fever	1	1
Fatigue	2	2
Constipation	1	1
Urinary Tract Pain	1	1
Laryngopharyngeal Dysesthesia	2	2

5.2.1 Physician Interaction: Patient Interface

NASA-TLX scores

TABLE 5.2: NASA-TLX scores: Patient Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Score**
Physician 1	40	10	20	20	20	22

The average NASA-TLX score of the LLM-guided interview was **22.0**, which demonstrated a low perceived workload. It implies that the interaction was easy and effective, and it did not require much thinking.

Temporal Demand (Q2, 10) demonstrated the lowest score on workload dimension, which proves that the speed of the interview was comfortable.

Effort, Frustration, and Performance (Q3-Q5, each 20), were slightly higher, however, remained within the range of acceptable values, showing that the dialogue occurred without causing discomfort or overload, and covered the necessary clinical dimensions.

The dimension with the highest score was Mental Demand (Q1, 40), which indicates that there was an inherent requirement of a certain level of clinical reasoning involved within the interview.

Qualitative Evaluation

TABLE 5.3: Qualitative 5-point scores: Patient Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**
Physician 1	4	5	3	-	5

The qualitative evaluation of the interview questions by the physician was overall positive with the scores ranging from 3 to 5.

Q1 (clarity and adequacy) scored 4 which indicates that the dialogue was mostly well-set with a space for improvement. Q2 (use of simple language) was the highest scored, showing that the system did not use medical jargon.

Q3 (objectivity) was rated 3, which refers to general ambiguity, as the system struggled with temporal questioning (e.g., "Do you have or have you ever had this symptom?"). Q4 (ability to reformulate) was omitted because this behaviour was not present.

The Q5 (clinical logic) had a score of 5, indicating that clinical logic in question flow was satisfactory.

System Usability Scale

TABLE 5.4: SUS scores: Patient Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Score**
Physician 1	4	1	5	2	5	2	4	2	5	1	87.5

Patient interface recorded a score of **87.5** This means that in the eyes of the physician, the patient experience would be accessible and direct. It received high ratings in the following domains of ease of use (Q3), feature integration (Q5), user confidence (Q9), and learnability (Q7). In the aspects of unnecessary complexity (Q2) and required prior knowledge (Q10) the scores were satisfactory. These findings imply that the interface is self-explanatory, well organized, and is suitable for first-time users. Slightly lower scores were observed in dimensions like perceived consistency (Q6), complexity (Q8), and need of support (Q4), which implies that in general the system is easy to navigate through, yet there are certain aspects of it which the user may require slight support. The frequency of use (Q1) was also rated lower, implying potential to enhance engagement.

Questionnaire for User Interaction Satisfaction

TABLE 5.5: QUIS scores: Patient Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Score**
Physician 1	7	8	8	9	8	7	8	-	8	7.88

The score on the QUIS questionnaire is **7.88** out of 9, which indicates a high level of user satisfaction in the patient interface and across all dimensions measured.

There was a high score on the visual appearance and layout (Q4), interface consistency (Q5), system response time (Q2), ease of navigation (Q3) and satisfaction with the interface (Q10), which implies the interface was perceived as attractive, fast and easy to navigate. The functionality (Q7) and accessibility (Q8) sections also showed high satisfaction, implying that the main aspects of the given system were highly accessible.

The slightly lower results were recorded in the clarity of messages and warnings (Q6) and ease of learning (Q1), which might indicate that information or instructions were sometimes displayed ambiguously.

The error handling and recovery options (Q9) is not present because there were no errors that happened throughout the evaluation session.

5.2.2 Physician Interaction: Medical Interface

NASA-TLX

TABLE 5.6: NASA-TLX scores: Medical Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Score**
Physician 1	8	10	10	10	10	4.8

The NASA-TLX is **4.8**, which represents a low workload perceived value during medical review task. The mental demand level (Q1) was the lowest score, which means that even though there was a low mental workload needed, some cognitive effort was still required. The results for the other dimensions (Q2, Q3, Q4, Q5), such as temporal demand, the level of effort, frustration, and perceived performance, indicate that the process was stress-free, effortless, and the provided information was adequate for decision-making in clinical environment.

Qualitative Evaluation

TABLE 5.7: Qualitative 5-point scores: Medical Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**
Physician 1	5	5	5	5	5

The physician found all the five components of the qualitative medical review interface with the highest rating of 5, which states that the user experience is overall good.

The summaries generated by the system were considered objective and clinically helpful (Q1) and the bullet points were a true representation of the described dialogue (Q2). The information provided (Q3) was clinical relevant and regarding the clarity and organisation (Q4), the report could be viewed as well organised and easy to interpret. The interface was also classified to be compatible within clinical workflows (Q5).

System Usability Scale

TABLE 5.8: SUS scores: Medical Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Score**
Physician 1	5	1	5	1	5	1	5	1	5	1	100

The medical interface scored **100.0** in SUS, indicating great usability. The rating of all positive framed items, including frequency of use, (Q1) ease of use (Q3), integration (Q5), learnability (Q7) and confidence (Q9) were rated at the maximum value of 5. At the same time, all negatively framed questions, referring to unnecessary complexity, support, complexity and learning curve (Q2, Q4, Q6, Q8, Q10), were scored at the minimum possible, which confirms that, overall the interface was perceived to be intuitive and easy to use.

Questionnaire for User Interaction Satisfaction

TABLE 5.9: QUIS scores: Medical Interface

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Score**
Physician 1	9	9	9	9	9	9	8	8	-	9	8.78

The medical interface received an average QUIS score of **8.78** which indicates a very high satisfaction level of the interface among the various dimensions.

Most dimensions were rated at the maximum: ease of learning (Q1), system responsiveness (Q2), navigation (Q3), visual layout (Q4), interface consistency (Q5), clarity of system message (Q6), and overall satisfaction (Q10) received maximum ratings. The general functionality (Q7) and accessibility (Q8) were rated slightly lower, although still remained high.

The item related to error handling and recovery (Q9) was not scored, as no errors were encountered during the evaluation.

Chapter 6

Conclusions and Future Work

This dissertation described the design, deployment and initial assessment of a clinical chatbot system that can perform structured symptom assessment. The objective was to explore whether chatbot using a large language model was suitable to gather and grade clinically relevant symptom data according to the CTCAE criteria.

The results of the clinical evaluation, though restricted to one oncologist, showed encouraging outcomes, as the grades given by the system corresponded completely to the grades given by the physician. In addition, the dialogue structure and reasoning capabilities were perceived favorably, and patient and physician interfaces had a low workload and scored high in usability and satisfaction using multiple validated instruments (NASA-TLX, SUS, and QUIS).

The hypothesis of the research is preliminarily confirmed as the findings imply that a chatbot guided by a large language model is able to conduct structured interviews, retrieving the symptoms and grading them according to the CTCAE guidelines.

However, this study has a number of limitations that need to be taken into consideration. First, the clinical assessment was carried out by a single oncologist, which limits the validity of the results, also the assessment did not involve any actual patient interactions, as the chatbot was tested in simulated conditions only. Third, the system's performance has not yet been evaluated across different user profiles or healthcare settings. These restrictions reduce the robustness of the existing conclusions and underline the necessity of wider validation.

Further research ought to focus on the enhancement of the clinical validation. The next step is to conduct assessments with more oncologists, in order to validate the system more thoroughly and to prepare it for real world settings through a large-scale trial in different clinical contexts. With this broader testing, it would be possible to expose the system to a wider range of user behaviours and clinical scenarios.

Another important direction involves capturing the clinician feedback when overrides to the symptom's grading occur. Collecting these justifications could enhance the system performance in grading the symptoms, either through direct fine-tuning or via reinforcement learning methods such as RLHF.

Finally, it will be critical to construct a high-quality benchmark dataset that includes patient responses, generated questions and clinician-approved grades, as such a resource would enable future research on training models for reliable symptom grading.

Bibliography

[1] X. Zhang, B. Peng, K. Li, J. Zhou, and H. Meng, "SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting," in *Findings of the Association for Computational Linguistics: EMNLP 2023* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 13348–13369, Association for Computational Linguistics, Dec. 2023.

[2] L. Cao, "Diaggpt: An llm-based chatbot with automatic topic management for task-oriented dialogue," *arXiv.org*, vol. abs/2308.08043, 2023.

[3] S. Sohn, Y. Lyu, A. Liu, L. Logeswaran, D.-K. Kim, D. Shim, and H. Lee, "Tfow: Modeling the structure of task-oriented dialogues," pp. 3355 – 3371, 2023.

[4] R. Wasenmüller, K. Hilbert, and C. Benzmüller, "Script-based dialog policy planning for llm-powered conversational agents: A basic architecture for an "ai therapist"," 2024.

[5] I. Soerjomataram and A. Jemal, "Global cancer statistics 2022: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 74, no. 3, pp. 229–263, 2024.

[6] L. Stark, C. Tofthagen, C. Visovsky, and S. C. McMillan, "The symptom experience of patients with cancer," *Journal of Hospice and Palliative Nursing: JHPN: The Official Journal of the Hospice and Palliative Nurses Association*, vol. 14, no. 1, pp. 61–70, 2012.

[7] T. L. Deshields, P. Potter, S. Olsen, J. Liu, and L. Dye, "Documenting the symptom experience of cancer patients.," *The journal of supportive oncology*, vol. 9 6, pp. 216–23, 2011.

[8] E. Spichiger, C. Müller-Fröhlich, K. Denhaerynck, H. Stoll, V. Hantikainen, and M. J. Dodd, "Prevalence of symptoms, with a focus on fatigue, and changes of symptoms over three months in outpatients receiving cancer chemotherapy.," *Swiss medical weekly*, vol. 141, p. w13303, 2011.

[9] E. Avalos-Reyes, D. Parke, K. Wallace, A. N. Freeman, and K. A. Johnson, "Characteristics of patients hospitalized through the emergency department with an oncology drug-related side effect.," *Journal of Clinical Oncology*, 2021.

[10] S. Lim and M. Yi, "Study on cancer patients who visited an emergency department with the side effects of chemotherapy," 2014.

[11] C. Caminiti, G. Maglietta, L. Arenare, R. D. Liello, G. Migliaccio, D. Barberio, M. D. Laurentiis, F. D. Rella, F. Nuzzo, C. Pacilio, G. Iodice, M. Orditura, F. Ciardiello, S. D. Bella, L. Cavanna, C. Porta, F. Giovanardi, C. Ripamonti, D. Bilancia, G. Aprile, T. Ruelle, F. Diodati, M. C. Piccirillo, E. Iannelli, C. Pinto, and F. Perrone, "Psychometric properties of patient-reported outcomes common terminology criteria for adverse events (pro-ctcae®) in breast cancer patients: the prospective observational multicenter vip study.," *The Breast*, 2024.

[12] M. F. Günther, L. Hentschel, M. Schulte, K. Schütte, Y. Ko, I. G. Schmidt-Wolf, and U. Jaehde, "Developing tumor-specific pro-ctcae item sets: analysis of a cross-sectional survey in three german outpatient cancer centers," *BMC Cancer*, 2023.

[13] C. T. E. Program, "Common terminology criteria for adverse events v3.0 (ctcae)," 2003.

[14] A. J. Nashwan and A. A. AbuJaber, "Harnessing the power of large language models (llms) for electronic health records (ehrs) optimization," *Cureus*, vol. 15, p. e42634, Jul 29 2023.

[15] L. Fan, L. Li, Z. Ma, S. Lee, H. Yu, and L. Hemphill, "A bibliometric review of large language models research from 2017 to 2023," *ACM Trans. Intell. Syst. Technol.*, vol. 15, Oct. 2024.

[16] M. Cascella, F. Semeraro, J. Montomoli, V. Bellini, O. Piazza, and B. Elena, "The breakthrough of large language models release for medical applications: 1-year timeline and perspectives," *Journal of Medical Systems*, vol. 48, 02 2024.

[17] A. R. Yeung, S. L. Pugh, A. H. Klopp, K. M. Gil, L. Wenzel, S. N. Westin, D. K. Gaffney, W. J. Small, S. Thompson, D. E. Doncals, G. H. C. Cantuaria, B. P. Yaremko, A. Chang, V. Kundapur, D. S. Mohan, M. L. Haas, Y. B. Kim, C. L. Ferguson, S. Deshmukh, D. W. Bruner, and L. A. Kachnic, "Improvement in patient-reported outcomes with intensity-modulated radiotherapy (rt) compared with standard rt: A report from the nrg oncology

rtog 1203 study," *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, vol. 38, no. 15, pp. 1685–1692, 2020.

[18] N. E. Billingy, C. J. G. van den Hurk, V. N. M. F. Tromp, and et al., "Patient- vs physician-initiated response to symptom monitoring and health-related quality of life: The symprolung cluster randomized trial," *JAMA Network Open*, vol. 7, p. e2428975, Aug 2024. Published 2024 Aug 1.

[19] L. R. Narra, N. Verdini, K. Lapen, R. Nipp, and E. F. Gillespie, "Patient-reported outcomes in clinical trials: From an endpoint to an intervention in cancer care," *Seminars in Radiation Oncology*, vol. 33, no. 4, pp. 358–366, 2023.

[20] S. G. Brearley, Z. Stamataki, J. Addington-Hall, C. Foster, L. Hodges, N. Jarrett, A. Richardson, I. Scott, M. Sharpe, D. Stark, C. Siller, L. Ziegler, and Z. Amir, "The physical and practical problems experienced by cancer survivors: a rapid review and synthesis of the literature," *European Journal of Oncology Nursing: The Official Journal of European Oncology Nursing Society*, vol. 15, no. 3, pp. 204–212, 2011.

[21] L. Barbera, C. Taylor, and D. Dudgeon, "Why do patients with cancer visit the emergency department near the end of life?," *CMAJ: Canadian Medical Association journal = journal de l'Association médicale canadienne*, vol. 182, no. 6, pp. 563–568, 2010.

[22] R. Barkley, M. J. Soobader, J. Wang, S. Blau, and R. D. Page, "Reducing cancer costs through symptom management and triage pathways," *JCO Oncology Practice*, vol. 15, no. 2, pp. e91–e97, 2019.

[23] N. R. Handley, L. M. Schuchter, and J. E. Bekelman, "Best practices for reducing unplanned acute care for patients with cancer," *JCO Oncology Practice*, vol. 14, no. 5, pp. 306–313, 2018.

[24] E. Basch, A. C. Dueck, L. J. Rogak, S. A. Mitchell, L. M. Minasian, A. M. Denicoff, J. K. Wind, M. C. Shaw, N. Heon, Q. Shi, B. Ginos, G. D. Nelson, J. P. Meyers, G. J. Chang, H. J. Mamon, M. R. Weiser, T. Kolevska, B. B. Reeve, D. W. Bruner, and D. Schrag, "Feasibility of implementing the patient-reported outcomes version of the common terminology criteria for adverse events in a multicenter trial: Ncctg n1048," *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, vol. 36, no. 31, p. JCO2018788620, 2018. Advance online publication.

[25] A. Anagnostou, C. Warren, J. Dantzer, A. D. Galvin, E. J. Phillips, D. Khan, and A. Banerji, "Patient-reported outcome measures in food and drug allergy.," 2024.

[26] A. Dinkel and M. Jahnen, "[patient-reported outcomes-the role of the patient's subjective perspective for research and clinical care].," 2024.

[27] M. Kosinski, L. Nelson, R. H. Stanford, J. D. Flom, and M. Schätz, "Patient-reported outcome measure development and validation: A primer for clinicians," *The Journal of Allergy and Clinical Immunology: In Practice*, 2024.

[28] E. Basch, L. Barbera, C. L. Kerrigan, and G. Velikova, "Implementation of patient-reported outcomes in routine medical care," *American Society of Clinical Oncology educational book. American Society of Clinical Oncology. Annual Meeting*, vol. 38, pp. 122–134, 2018.

[29] J. Chen, L. Ou, and S. J. Hollis, "A systematic review of the impact of routine collection of patient reported outcome measures on patients, providers and health organisations in an oncologic setting," *BMC Health Services Research*, vol. 13, p. 211, 2013.

[30] R. Mercieca-Bebber, M. T. King, M. J. Calvert, M. R. Stockler, and M. Friedlander, "The importance of patient-reported outcomes in clinical trials and strategies for future optimization," *Patient Related Outcome Measures*, vol. 9, pp. 353–367, 2018.

[31] G. Kotronoulas, N. Kearney, R. Maguire, A. Harrow, D. Di Domenico, S. Croy, and S. MacGillivray, "What is the value of the routine use of patient-reported outcome measures toward improvement of patient outcomes, processes of care, and health service outcomes in cancer care? a systematic review of controlled trials," *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, vol. 32, no. 14, pp. 1480–1501, 2014.

[32] S. Jarosławski, M. Toumi, C. Chouhaid, B. Fallissard, and P. Auquier, "Patient reported outcomes (pro) in oncology, what beyond randomized controlled trials?," *Value in Health*, vol. 21, p. S25, 2018.

[33] A. Nikolic, O. A. Botrugno, J. Urosevic, and N. Tosic, "Cancer monitoring methods," *BioMed Research International*, vol. 2014, p. 981495, 2014.

[34] J. Meirte, N. Hellemans, M. Anthonissen, L. Denteneer, K. Maertens, P. Moortgat, and U. Van Daele, "Benefits and disadvantages of electronic patient-reported outcome measures: Systematic review," *JMIR Perioperative Medicine*, vol. 3, no. 1, p. e15588, 2020.

[35] M. C. O'Leary, E. Kwong, C. Cox, A. Gentry, A. M. Stover, M. B. Vu, J. Carda-Auten, J. Leeman, and G. N. Mody, "Patient motivators of postoperative electronic patient-reported outcome symptom monitoring use in thoracic surgery patients: a qualitative study," *Journal of Patient-Reported Outcomes*, 2024.

[36] H. Yu, C. Lei, W. Xing, Y. Wang, W. Xu, L. Tang, W. Dai, J. Liao, P. Yang, R. Gong, X. Su, Q. Yu, J.-Y. Zhang, L. Zhang, Y. Huang, X. Zhuang, J. Bai, Z. Wang, Q. Li, and Q. Shi, "Electronic symptom monitoring after lung cancer surgery: establishing a core set of patient-reported outcomes for surgical oncology care in a longitudinal cohort study," *International Journal of Surgery*, 2024.

[37] C. W. Noel, Y. J. Du, E. Baran, D. Forner, Z. Husain, K. M. Higgins, I. Karam, K. K. W. Chan, J. Hallet, F. Wright, N. G. Coburn, A. Eskander, and L. Gotlib Conn, "Enhancing outpatient symptom management in patients with head and neck cancer: A qualitative analysis," *JAMA Otolaryngology–Head & Neck Surgery*, vol. 148, no. 4, pp. 333–341, 2022.

[38] J. Gill, M. Pucci, A. Samudio, T. Ahmed, R. Siddiqui, N. Edwards, R. M. Marticorena, S. Donnelly, C. Lok, K. Wentlandt, K. T. Wolofsky, and I. Mucsi, "Self-reported measurement of physical and psychosocial symptoms response tool (support-dialysis): systematic symptom assessment and management in patients on in-centre haemodialysis – a parallel arm, non-randomised feasibility pilot study protocol," *BMJ Open*, vol. 14, 2024.

[39] "Longitudinal subgrouping of patients with cancer based on symptom experiences: An integrative review," *Oncology Nursing Forum*, 2022.

[40] S. B. Clauser, P. A. Ganz, J. Lipscomb, and B. B. Reeve, "Patient-reported outcomes assessment in cancer trials: evaluating and enhancing the payoff to decision making.," *Journal of Clinical Oncology*, vol. 25, pp. 5049 – 5050, 2007.

[41] C. Xiao, R. C. Polomano, and D. W. Bruner, "Comparison between patient-reported and clinician-observed symptoms in oncology.," *Cancer Nursing*, vol. 36, 2013.

[42] D. F. Lobach, M. E. Cooley, B. Halpenny, H. D. Yackel, A. Boxwala, and J. L. Abrahm, "1. capturing the patient's voice to inform advanced clinical decision support for guideline-based cancer symptom management.," *JCO oncology practice*, 2023.

[43] L. Lizán, L. Pérez-Carbonell, and M. Comellas, "1. additional value of patient-reported symptom monitoring in cancer care: A systematic review of the literature.," *Cancers*, 2021.

[44] E. M. Basch, A. M. Deal, M. G. Kris, H. I. Scher, C. A. Hudis, P. J. Sabbatini, L. J. Rogak, A. V. Bennett, A. C. Dueck, T. M. Atkinson, J. F. Chou, D. Dulko, L. Sit, A. Barz, P. J. Novotny, M. Fruscione, J. Sloan, and D. Schrag, "Symptom monitoring with patient-reported outcomes during routine cancer treatment: A randomized controlled trial.," *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, vol. 34 6, pp. 557–65, 2016.

[45] A. M. Trotti, A. D. Colevas, A. Setser, and E. M. Basch, "Patient-reported outcomes and the evolution of adverse event reporting in oncology.," *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, vol. 25 32, pp. 5121–7, 2007.

[46] B. Movsas, "Proceeding with the patient-reported outcomes (pros) version of the common terminology criteria for adverse events," *JAMA Oncology*, vol. 1, pp. 1059 – 1060, 2015.

[47] E. Basch, B. B. Reeve, S. A. Mitchell, S. B. Clauser, L. M. Minasian, A. C. Dueck, T. R. Mendoza, J. Hay, T. M. Atkinson, A. P. Abernethy, D. W. Bruner, C. S. Cleeland, J. A. Sloan, R. Chilukuri, P. Baumgartner, A. Denicoff, D. St Germain, A. M. O'Mara, A. Chen, J. Kelaghan, and D. Schrag, "Development of the national cancer institute's patient-reported outcomes version of the common terminology criteria for adverse events (pro-ctcae)," *Journal of the National Cancer Institute*, vol. 106, no. 9, p. dju244, 2014.

[48] T. Yamaguchi, T. Kawaguchi, and T. Miyaji, "Patient reported outcome(pro)in oncology," *Gan to kagaku ryoho. Cancer & chemotherapy*, vol. 46, pp. 1345 – 1356, 2019.

[49] L. M. Minasian, A. M. O'Mara, and S. A. Mitchell, "Clinician and patient reporting of symptomatic adverse events in cancer clinical trials: Using ctcae and pro-ctcae® to provide two distinct and complementary perspectives," *Patient Related Outcome Measures*, vol. 13, pp. 249 – 258, 2022.

[50] A. Pradhan, P. Dwived, P. Pareek, S. Kalra, S. Singh, S. B. Varthya, A. D. Goel, M. A. Shamim, J. R. Vishnoi, P. P. Sharma, A. Pradhan, S. Ambwani, and S. Misra, "Electronic-pro measures for adverse events of treatment in oncology (e-promotion)," *International Journal of Medical Informatics*, 2023.

[51] D. V. P. de Almeida, J. M. Anderson, D. C. Danila, M. J. Morris, S. F. Slovin, W. Abida, E. Cohn, R. Baser, H. I. Scher, and K. A. Autio, "Evaluating immune-related adverse events using pro-ctcae in a phase ii study of ipilimumab for hormone-sensitive prostate cancer," *Journal of immunotherapy and precision oncology*, 2023.

[52] S. J. Ivatury, S. J. Ivatury, S. L. Wong, S. L. Wong, and D. Schrag, "Aso author reflections: Patient-reported outcomes-bench to bedside," *Annals of Surgical Oncology*, vol. 27, pp. 725 – 726, 2020.

[53] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2024.

[54] W. Du, T. Luo, Z. Qiu, Z. Huang, Y. Shen, R. Cheng, Y. Guo, and J. Fu, "Stacking your transformers: A closer look at model growth for efficient llm pre-training," 2024.

[55] M. Škorić, M. Utvić, and R. Stanković, "Transformer-based composite language models for text evaluation and classification," *Mathematics*, vol. 11, no. 22, 2023.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[57] M. Fujitake, "Dtrocr: Decoder-only transformer for optical character recognition," *arXiv.org*, vol. abs/2308.15996, 2023.

[58] J. Roberts, "On the computational power of decoder-only transformer language models," *CoRR*, vol. abs/2305.17026, 2023.

[59] G. Lee, J. H. Park, D. S. Papailiopoulos, and K. Lee, "Prompted llms as chatbot modules for long open-domain conversation," *arXiv.org*, vol. abs/2305.04533, 2023.

[60] Z. Li, Z. Z. Chen, M. Ross, P. Huber, S. Moon, Z. Lin, X. L. Dong, A. Sagar, X. Yan, and P. A. Crook, "Large language models as zero-shot dialogue state tracker through function calling," 2024.

[61] S. Vatsal and H. Dubey, "A survey of prompt engineering methods in large language models for different nlp tasks," *arXiv.org*, vol. abs/2407.12994, 2024.

[62] M. F. Russe, M. Reisert, F. Bamberg, and A. Rau, "Improving the use of llms in radiology through prompt engineering: from precision prompts to zero-shot learning.," 2024.

[63] Y. Xia *et al.*, "Understanding the performance and estimating the cost of llm fine-tuning," August 2024.

[64] V. B. Parthasarathy, A. Zafar, A. Khan, and A. Shahid, "The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities," October 2024.

[65] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," April 2021.

[66] A. Storhaug and J. Li, "Parameter-efficient fine-tuning of large language models for unit test generation: An empirical study," November 2024.

[67] K. Yao *et al.*, "Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models," November 2024.

[68] K. Galim, W. Kang, Y. Zeng, H. I. Koo, and K. Lee, "Parameter-efficient fine-tuning of state space models," October 2024.

[69] J. Koo, M. Jang, and J. Ok, "Towards robust and efficient federated low-rank adaptation with heterogeneous clients," October 2024.

[70] L. Mi *et al.*, "V-lora: An efficient and flexible system boosts vision applications with lora lmm," November 2024.

[71] Y. Hou, H. Dong, X. Wang, B. Li, and W. Che, "Metaprompting: Learning to learn better prompts," 2022.

[72] G. Qin and J. Eisner, "Learning how to ask: Querying lms with mixtures of soft prompts," *ArXiv*, vol. abs/2104.06599, 2021.

[73] J. Wu, T. Yu, R. Wang, Z. Song, R. Zhang, H. Zhao, C. Lu, S. Li, and R. Henao, "Infoprompt: Information-theoretic soft prompt tuning for natural language understanding," vol. 36, pp. 61060–61084, 2023.

[74] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.

[75] Z. Chen, K. Zhou, B. Zhang, Z. Gong, W. X. Zhao, and J.-R. Wen, "Chatcot: Tool-augmented chain-of-thought reasoning on chat-based large language models," November 2023.

[76] Z. Sprague *et al.*, "To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning," October 2024.

[77] Z. Jin and W. Lu, "Self-harmonized chain of thought," September 2024.

[78] N. Rawal, P. Tavva, and P. Selvakumar, "Enhancing large language model performance with reinforcement learning from human feedback: A comprehensive study on q&a, summarization, and classification," pp. 1 – 6, 2024.

[79] R. Kirk, I. Mediratta, C. Nalmpantis, J. Luketina, E. Hambro, E. Grefenstette, and R. Raileanu, "Understanding the effects of rlhf on llm generalisation and diversity," *arXiv.org*, vol. abs/2310.06452, 2023.

[80] X. Li, M. Gao, Z. Zhang, J. H. Fan, and L. Tian, "Data-adaptive safety rules for training reward models," 2025.

[81] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2024.

[82] S. Vakayil, D. S. Juliet, A. J, and S. Vakayil, "Rag-based llm chatbot using llama-2," in *2024 7th International Conference on Devices, Circuits and Systems (ICDCS)*, pp. 1–5, 2024.

[83] A. Khan, M. T. Hasan, K. Kemell, J. Rasku, and P. Abrahamsson, "Developing retrieval augmented generation (rag) based llm systems from pdfs: An experience report," 10 2024.

[84] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers," *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 155–161, 2024.

[85] X. Zhao *et al.*, "Agentigraph: An interactive knowledge graph platform for llm-based chatbots utilizing private data," October 2024.

[86] X. Tan, X. Wang, Q. Liu, X. Xu, X. Yuan, and W. Zhang, "Paths-over-graph: Knowledge graph empowered large language model reasoning," October 2024.

[87] K. Qu, K. C. Li, B. T. M. Wong, M. M. F. Wu, and M. Liu, "A survey of knowledge graph approaches and applications in education," *Electronics*, vol. 13, p. Art. n.o 13, January 2024.

[88] M. M. Akter and M.-M. Rahoman, "A systematic review on knowledge graphs classification and their various usages," *Artificial Intelligence Evolution*, pp. 187–215, November 2023.

[89] D. F. Daza Cruz, *Exploiting Subgraphs and Attributes for Representation Learning on Knowledge Graphs*. PhD thesis, Vrije Universiteit Amsterdam, 2024.

[90] L. Aryan, L. Thingbaijam, K. Palle, P. V. Prasad, B. Mallala, and S. Patil, "Incorporating knowledge graphs in semantic search," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–6, June 2024.

[91] Z. Huang, S. Gutierrez, H. Kamana, and S. MacNeil, "Memory sandbox: Transparent and interactive memory management for conversational agents," August 2023.

[92] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "Memorybank: Enhancing large language models with long-term memory," May 2023.

[93] B. Kynoch, H. Latapie, and D. van der Sluis, "Recallm: An adaptable memory mechanism with temporal understanding for large language models," October 2023.

[94] A. Rezazadeh, Z. Li, W. Wei, and Y. Bao, "From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms," December 2024.

[95] Y. Hou, H. Tamoto, and H. Miyashita, "Integrating temporal representations for dynamic memory retrieval and management in large language models," October 2024.

[96] Y. Cheng, K. Mao, Z. Zhao, G. Dong, H. Qian, Y. Wu, T. Sakai, J.-R. Wen, and Z. Dou, "Coral: Benchmarking multi-turn conversational retrieval-augmentation generation," October 2024.

[97] N. Alonso, T. Figliolia, A. Ndirango, and B. Millidge, "Toward conversational agents with context and time sensitive long-term memory," June 2024.

[98] Y. Li, "Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering," April 2023.

[99] Y. Li, B. Dong, C. Lin, and F. Guerin, "Compressing context to enhance inference efficiency of large language models," October 2023.

[100] R. Ribeiro and L. Coheur, "Sumbot: Summarizing context in open-domain dialogue systems," October 2022.

[101] M. T. Ansari, "Dialogue management as graph transformations," pp. 219 – 227, 2022.

[102] K. A. S. Factor and L. J. Langley, "Digraphs with isomorphic underlying and domination graphs: Pairs of paths," *The journal of combinatorial mathematics and combinatorial computing*, 2010.

[103] T. H. Cormen, "Directed acyclic graphs," *IEEE Xplore*, pp. 71–89, Jan. 2013.

[104] N. Apollonio and L. Balzotti, "Two new characterizations of path graphs," *Discrete Mathematics*, vol. 346, p. 113596, 2023.

[105] D. Ham, J.-G. Lee, Y. Jang, and K.-E. Kim, "End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2," *Meeting of the Association for Computational Linguistics*, pp. 583 – 592, 2020.

[106] H.-D. Xu, X. Mao, P. Yang, F. Sun, and H. Huang, "Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent," pp. 2748 – 2763, 2024.

[107] Y. Xu, "Finite-state machines," pp. 373 – 388, Elsevier eBooks, 2023.

[108] A. Boronat, R. Bruni, A. L. Lafuente, U. Montanari, and G. Paolillo, "Exploiting the hierarchical structure of rule-based specifications for decision planning," vol. 6117, pp. 2 – 16, Springer-Verlag, 2010.

[109] J. Cuadrado, S. Pérez-Soler, E. Guerra, and J. de Lara, "Automating the development of task-oriented llm-based chatbots," 2024.

[110] O. Seminck, "Conversational ai: Dialogue systems, conversational agents, and chatbots," *Computational Linguistics*, vol. 49, pp. 257 – 259, 2023.

[111] E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher, "A simple language model for task-oriented dialogue," *arXiv: Computation and Language*, 2020.

[112] N. Walker, S. Ultes, and P. Lison, "Graphwoz: Dialogue management with conversational knowledge graphs," *arXiv.org*, vol. abs/2211.12852, 2022.

[113] C. Chun, D. Rim, and J. Park, "LLM ContextBridge: A hybrid approach for intent and dialogue understanding in IVSR," in *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track* (O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert, K. Darwish, and A. Agarwal, eds.), (Abu Dhabi, UAE), pp. 794–806, Association for Computational Linguistics, Jan. 2025.

[114] M. D. J. Peters, C. Godfrey, P. McInerney, Z. Munn, A. C. Tricco, and H. Khalil, "Chapter 11: Scoping reviews (2020 version)," in *JBI Manual for Evidence Synthesis* (E. Aromataris and Z. Munn, eds.), JBI, 2020.

[115] A. G. Silva, P. Simões, R. Santos, A. Queirós, N. P. Rocha, and M. Rodrigues, "A scale to assess the methodological quality of studies assessing usability of electronic health products and services: Delphi study followed by validity and reliability testing," *J Med Internet Res*, vol. 21, p. e14829, Nov 2019.

[116] J. Y. Soh, W. C. Cha, D. K. Chang, J. H. Hwang, K. Kim, M. Rha, and H. Kwon, "Development and validation of a multidisciplinary mobile care system for patients with advanced gastrointestinal cancer: Interventional observation study," *JMIR mHealth and uHealth*, vol. 6, p. e115, May 2018.

[117] M. Egbring, E. Far, M. Roos, M. Dietrich, M. Brauchbar, G. A. Kullak-Ublick, and A. Trojan, "A mobile app to stabilize daily functional activity of breast cancer patients in collaboration with the physician: A randomized controlled clinical trial," *Journal of Medical Internet Research*, vol. 18, p. e238, Sep 2016.

[118] C. Sin, H. Kim, H. S. Im, *et al.*, "Development and pilot study of "smart cancer care": a platform for managing side effects of chemotherapy," *BMC Health Services Research*, vol. 23, p. 922, 2023.

[119] A. Lencioni, L. Hutchins, S. Annis, *et al.*, "An adverse event capture and management system for cancer studies," *BMC Bioinformatics*, vol. 16, no. Suppl 13, p. S6, 2015.

[120] M. J. Hassett, C. Cronin, T. C. Tsou, J. Wedge, J. Bian, D. S. Dizon, H. Hazard-Jenkins, R. U. Osarogiagbon, S. Wong, E. Basch, T. Austin, N. McCleary, and D. Schrag, "esym: An

electronic health record–integrated patient-reported outcomes–based cancer symptom management program used by six diverse health systems," *JCO Clinical Cancer Informatics*, no. 6, p. e2100137, 2022. PMID: 34985914.

[121] A. M. R. Kestler, S. D. Kühlwein, J. M. Kraus, J. D. Schwab, R. Szekely, P. Thiam, R. Hühne, N. Jahn, A. Fürstberger, N. Ikonomi, J. Balig, R. Schuler, P. Kuhn, F. Steger, T. Seufferlein, and H. A. Kestler, "Digitalization of adverse event management in oncology to improve treatment outcome—a prospective study protocol," *PLOS ONE*, vol. 16, no. 6, p. e0252493, 2021.

[122] K. Lapen, C. Sabol, A. L. Tin, K. Lynch, A. Kassa, X. Mabli, J. Ford, E. Cha, M. B. Bernstein, L. Z. Braunstein, O. Cahlon, B. M. Daly, K. Sandler, S. A. McCloskey, A. J. Vickers, A. J. Khan, and E. F. Gillespie, "Development and pilot implementation of a remote monitoring system for acute toxicity using electronic patient-reported outcomes for patients undergoing radiation therapy for breast cancer," *International Journal of Radiation Oncology, Biology, Physics*, vol. 111, no. 4, pp. 979–991, 2021.

[123] S. Gomaa, J. Posey, B. Bashir, A. Basu Mallick, E. Vanderklok, M. Schnoll, T. Zhan, and K. Y. Wen, "Feasibility of a text messaging-integrated and chatbot-interfaced self-management program for symptom control in patients with gastrointestinal cancer undergoing chemotherapy: Pilot mixed methods study," *JMIR Formative Research*, vol. 7, p. e46128, 2023.

[124] S. Moradian, S. Ghasemi, B. Boutorabi, Z. Sharifian, F. Dastjerdi, C. Buick, C. T. Lee, S. J. Mayo, P. P. Morita, and D. Howell, "Development of an ehealth tool for capturing and analyzing the immune-related adverse events (iraes) in cancer treatment," *Cancer Informatics*, vol. 22, p. 11769351231178587, 2023.

[125] C. Grašič Kuhar, T. Gortnar Cepeda, T. Kovač, M. Kukar, and N. Ružić Gorenjec, "Mobile app for symptom management and associated quality of life during systemic treatment in early stage breast cancer: Nonrandomized controlled prospective cohort study," *JMIR mHealth and uHealth*, vol. 8, no. 8, p. e17408, 2020.

[126] E. A. Hægermark, N. Kongshaug, S. X. Raj, E. Hofsli, and A. Faxvaag, "Design, development, and evaluation of an mhealth app for reporting of side effects during cytostatic treatment: Usability test and interview study," *JMIR Formative Research*, vol. 7, p. e47374, 2023.

[127] A. B. Leahy *et al.*, "Electronic symptom monitoring in pediatric patients hospitalized for chemotherapy," *Cancer*, vol. 127, pp. 2980–2989, Aug 2021.

[128] J. Underwood *et al.*, "Developing a mobile patient-reported outcomes version of the common terminology criteria for adverse events administration system to capture postradiation toxicity in oncology: Usability and feasibility study," *JMIR Formative Research*, vol. 6, p. e27775, Apr 2022.

[129] C. Taramasco, C. Rimassa, R. Noël, M. L. B. Storm, and C. Sánchez, "Co-design of a mobile app for engaging breast cancer patients in reporting health experiences: Qualitative case study," *Journal of Medical Internet Research*, vol. 25, p. e45968, Nov 2023.

[130] "General Data Protection Regulation (GDPR), Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016." `https://eur-lex.europa.eu/eli/reg/2016/679/oj`, 2016.

[131] S. Grüne, "Anamnese und körperliche untersuchung [anamnesis and clinical examination]," *Deutsche medizinische Wochenschrift (1946)*, vol. 141, no. 1, pp. 24–27, 2016.

[132] C. J. Hermansen-Kobulnicky, "1. symptom-monitoring behaviors of rural cancer patients and survivors," *Supportive Care in Cancer*, 2009.

[133] A. C. Tricco, E. Lillie, W. Zarin, K. K. O'Brien, H. L. Colquhoun, D. E. Levac, D. Moher, M. D. Peters, T. Horsley, L. Weeks, S. Hempel, E. A. Akl, C. Chang, J. McGowan, L. A. Stewart, L. Hartling, A. Aldcroft, M. Wilson, C. M. Garritty, S. Lewin, C. Godfrey, M. Macdonald, E. V. Langlois, K. Soares-Weiser, J. Moriarty, T. Clifford, zge Tunalp, and S. E. Straus, "Prisma extension for scoping reviews (prisma-scr): Checklist and explanation," *Annals of Internal Medicine*, vol. 169, pp. 467–473, 2018.

[134] E. Miller and H. L. Colquhoun, "The importance and value of reporting guidance for scoping reviews: A rehabilitation science example," *Australian Journal of Advanced Nursing*, vol. 37, 2020.

[135] S. M. Chang, "Scoping reviews and systematic reviews: Is it an either/or question?," *Annals of Internal Medicine*, vol. 169, pp. 502–503, 2018.

[136] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, *et al.*, "The prisma 2020 statement: an updated guideline for reporting systematic reviews," *BMJ*, vol. 372, p. n71, Mar 29 2021. [FREE Full text].

[137] Z. Nie, R. Zhang, and Z. Wu, "A text is worth several tokens: Text embedding from llms secretly aligns well with the key tokens," 2024.

[138] E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner, "Do nlp models know numbers? probing numeracy in embeddings," pp. 5306–5314, Association for Computational Linguistics, 2019.

[139] P. Dufter, M. Schmitt, and H. Schütze, "Position information in transformers: An overview," 2021.

[140] P.-C. Chen, H. Tsai, S. Bhojanapalli, H. W. Chung, Y.-W. Chang, and C.-S. Ferng, "A simple and effective positional encoding for transformers," 2021.

[141] M. Tao, Y. Feng, and D. Zhao, "A frustratingly easy improvement for position embeddings via random padding," *arXiv.org*, vol. abs/2305.04859, 2023.

[142] L. Ruan and Q. Jin, "Survey: Transformer based video-language pre-training," 2021.

[143] A. Hernández and J. Amigó, "Attention mechanisms and their applications to complex systems," *Entropy*, vol. 23, p. 283, 02 2021.

[144] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[145] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," 2021.

[146] R. K. Mahabadi, Y. Belinkov, and J. Henderson, "Variational information bottleneck for effective low-resource fine-tuning," 2021.

[147] E. Rusli, D. Wujcik, and A. Galaznik, "2. remote symptom alerts and patient-reported outcomes (pros) in real-world breast cancer practice: Innovative data to derive symptom burden and quality of life," *Bioengineering*, 2024.

[148] B. Stollnitz, "The transformer architecture of gpt models," 2024. Accessed: Mar. 27, 2024.

[149] Y. Feng, Z. Lu, B. Liu, L. Zhan, and X.-M. Wu, "Towards llm-driven dialogue state tracking," 2023.

[150] H. Li, C. Yang, A. Zhang, Y. Deng, X. Wang, and T.-S. Chua, "Hello again! llm-powered personalized agent for long-term dialogue," June 2024.

[151] A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y. Fang, "Evaluating very long-term conversational memory of llm agents," February 2024.

[152] D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D. Yu, "Longmemeval: Benchmarking chat assistants on long-term interactive memory," October 2024.

[153] F. Hohman, M. B. Kery, D. Ren, and D. Moritz, "Model compression in practice: Lessons learned from practitioners creating on-device machine learning experiences," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, (Honolulu, HI, USA), pp. 1–18, ACM, May 2024.

[154] G. Yang *et al.*, "Llmcbench: Benchmarking large language model compression for efficient deployment," October 2024.

[155] H. Li, J. Hong, Y. Wu, S. Adbol, and Z. Li, "Continuous approximations for improving quantization aware training of llms," October 2024.

[156] "Compact language models via pruning and knowledge distillation," 2024. Acedido: 7 de janeiro de 2025.

[157] X. Yao, H. Qian, X. Hu, G. Xu, and Y. Liu, "Theoretical insights into fine-tuning attention mechanism: Generalization and optimization," October 2024.

[158] M. Paraschiv, "A topology-based directed acyclic graph generation strategy," Jan. 2010. Exact publication venue not specified.

[159] M. Mirtaheri, N. Varghese, C. Khatri, and A. Kelkar, "Towards automatic evaluation of task-oriented dialogue flows," *arXiv preprint*, vol. arXiv:2411.10416, Nov. 2024. [Online; accessed 2025-06-11].

[160] F. Yang, "Directing the flow of conversation in task-oriented dialogue," Master's thesis, Oregon Health and Science University, Jan. 2008. [Online; accessed 2025-06-11].

[161] Z. Hao, J. Zhang, B. Xu, Y. Wang, G. de Melo, and X. Li, "Intentdial: An intent graph based multi-turn dialogue system with reasoning path visualization," *CoRR*, vol. abs/2310.11818, Oct. 2023. [Online; accessed 2025-06-11].

[162] K. Xu, Y. Cheng, W. Hou, Q. Tan, and W. Li, "Reasoning like a doctor: Improving medical dialogue systems via diagnostic reasoning process alignment," in *Findings of the Association for Computational Linguistics: ACL 2024* (L.-W. Ku, A. Martins, and V. Srikumar, eds.), (Bangkok, Thailand), pp. 6796–6814, Association for Computational Linguistics, Aug. 2024. [Online; accessed 2025-06-11].

[163] M. Malakhaltsev, E. Osipov, I. Kozlov, and L. Sheremetov, "A generalized dialogue graph construction and visualization based on a corpus of dialogues," *Prikladnaia Diskretnaia Matematika (Applied Discrete Mathematics)*, vol. 59, pp. 111–127, dec 2023. [Online; accessed 2025-06-11].

[164] S. S. Ray, "Trees and fundamental circuits," pp. 35 – 47, Springer, India, 2013.

[165] S. Bhaskar and A. J. Kienzle, "Algorithmic traversals of infinite graphs," *arXiv: Logic*, 2018.

[166] H.-J. Kreowski and S. Kuske, "Modeling and analyzing graph algorithms by means of graph transformation units.," *The Journal of Object Technology*, vol. 19, 2020.

[167] J. Mok, M. Kachuee, S. Dai, S. Ray, T. Taghavi, and S. Yoon, "Llm-based frameworks for api argument filling in task-oriented conversational systems," 2024.

[168] B. Zhou, D. Geißler, and P. Lukowicz, "Misinforming llms: vulnerabilities, challenges and opportunities," 2024.

[169] S. Banerjee, A. Agarwal, and S. Singla, "Llms will always hallucinate, and we need to live with this," 2024.

[170] S. Ahuja, Z. Tan, H. Sidahmed, J. Xie, G. Schubiner, A. Patel, H. Lara, B. Chu, Z. Chen, and M. Tiwari, "Leveraging large language models in conversational recommender systems," *arXiv.org*, vol. abs/2305.07961, 2023.

[171] U.S. Department of Health and Human Services, National Institutes of Health, National Cancer Institute, "Common terminology criteria for adverse events (ctcae) version 5.0." `https://ctep.cancer.gov/protocoldevelopment/electronic_applications/ctc.htm#ctc_50`, 2017. Accessed June 2025.

[172] C. Strathearn and D. Gkatzia, "Chefbot: A novel framework for the generation of commonsense-enhanced responses for task-based dialogue systems," in *Proceedings of the*

14th International Conference on Natural Language Generation (A. Belz, A. Fan, E. Reiter, and Y. Sripada, eds.), (Aberdeen, Scotland, UK), pp. 46–47, Association for Computational Linguistics, Aug. 2021.

Appendix A

Evaluation Instruments: Full Question Set

A.1 Patient Review Interface Evaluation

Simplified NASA-TLX

Participants rated agreement (1–10) for the following statements:

1. Was clinical reasoning easy during the interview?
2. Was the pace comfortable and free of time pressure?
3. Was it easy to follow the question sequence?
4. Did you feel comfortable and not frustrated during the interview?
5. Did the system collect sufficient clinical information to support evaluation?

Qualitative Evaluation

Participants rated agreement (1–5) for the following statements:

1. Were the questions clear and appropriate?
2. Did the system use simple, patient-friendly language (no medical jargon)?
3. Were the questions objective and unambiguous?
4. Did the system handle vague/confusing responses effectively?
5. Did the question flow follow a logical clinical sequence?

A.2 Medical Review Interface Evaluation

Simplified NASA-TLX

Participants rated agreement (1–10) for the following statements:

1. Was reviewing the report simple and not mentally demanding?
2. Was the review process fast and efficient?
3. Was it easy to analyze and confirm the information?
4. Was the information clear and free of errors or inconsistencies?
5. Was the report clinically adequate to support decision-making?

Qualitative Evaluation (1 to 5)

Participants rated agreement (1–5) for the following statements:

1. Was the medical summary objective and useful?
2. Did the bullet points correctly reflect the reported elements?
3. Was the included information clinically relevant and sufficient?
4. Was the report well organized and clear?
5. Was the format compatible with clinical workflows?

A.3 Usability Evaluation (SUS and QUIS)

System Usability Scale (SUS)

Participants rated agreement (1–5) for the following statements, for both the patient and physician interfaces:

1. I would use this system frequently.
2. The system is unnecessarily complex.
3. The system is easy to use.
4. I would need the support of a technical person to use this system.

5. The functionalities are well integrated.
6. There is too much inconsistency in the system.
7. Most people would learn to use it quickly.
8. The system is overly complicated.
9. I felt confident using the system.
10. I had to learn a lot before using the system.

User Interaction Satisfaction (QUIS)

Participants rated satisfaction (1–9) for both interfaces on:

1. Ease of learning
2. System response time
3. Ease of navigation
4. Visual appearance and layout
5. Interface consistency
6. Clarity of messages and warnings
7. Overall functionality
8. Accessibility
9. Error handling and recovery options
10. Overall satisfaction with the interface

Appendix B

Medical Report

The following pages present a full example of the automatically generated clinical report produced by the system after patient interaction and symptom evaluation.

Relatório da Sessão

Paciente: Hugo Silva

ID da Sessão: 109

Data: 2025-06-16 17:43:49

Sintomas Avaliados

Neuropatia periférica sensitiva – Não presente

Resumo: O paciente apresentou formigueiro e dormência nos primeiros 5 dias, principalmente ao entrar em contato com objetos frios.

- Formigueiro
- Dormência
- Sintomas durante os primeiros 5 dias
- Agravado por contato com objetos frios

Diarreia – Não presente

Resumo: O paciente não apresenta diarreia.

- O paciente negou episódios de fezes moles ou líquidas.

Náuseas – Grau 3 (Rever)

Resumo: O paciente apresenta náuseas frequentes ocorrendo várias vezes ao dia durante a primeira semana após o início do tratamento, com redução do apetite que se resolvia após essa semana.

- Náuseas frequentes (várias vezes por dia) desde o primeiro dia após o início do tratamento
- Redução do apetite durante a primeira semana

Justificação do LLM: A classificação GRADE 2 reflete a frequência diária das náuseas (várias vezes por dia) e a perda de apetite com redução da ingestão

oral sem comprometer significativamente o peso ou levando à necessidade de alimentação assistida.

Vómitos – Não presente

Resumo: O paciente não relata episódios de vómitos.

- Não há história de vómitos.

Mucosite oral – Grau 3 (Confirmado)

Resumo: O paciente apresenta mucosite com dor moderada que iniciou no quarto dia após o início do tratamento e resultou em dificuldade para mastigar, limitando a ingestão a gelatina, sopa fria e líquidos por uma semana.

- Mucosite
- Dor moderada
- Início na 4ª dia após o início do tratamento
- Dificuldade em mastigar
- Limitação da alimentação a gelatina, sopa fria e líquidos

Justificação do LLM: A classificação como GRADE 3 justifica-se pela presença de dor moderada e por interferência significativa na ingestão oral, necessitando a paciente de uma dieta modificada.

Fadiga – Não presente

Resumo: O paciente não apresenta sintomas de fadiga ou cansaço extremo.

- Negativa para fadiga
- Negativa para cansaço extenuante

Alopecia – Não presente

Resumo: O paciente não apresenta perda capilar excessiva.

- Perda capilar: negativa

Rash pustular – Não presente

Resumo: O paciente apresenta lesões cutâneas sem evidência de úlceras elevadas com pus.

- Feridas elevadas com pus negativo
- Lesões cutâneas presentes

Febre – Grau 1 (Confirmado)

Resumo: A paciente apresenta febre persistente com picos de até 38.3°C há dois dias, ocorrendo em episódios diários com intervalos de 8 horas.

- Fevereiro persistente
- Picos máximos: 38.3°C
- Duração: 2 dias
- Episódios diários com intervalos de 8 horas

Justificação do LLM: A febre foi classificada como GRADE 1 por estar dentro do intervalo de 38,0 a 39,0 ºC e durar mais de 24 horas.

Cefaleia – Não presente

Resumo: O paciente não apresenta dor de cabeça.

- Dor de cabeça negativa.

Calafrios – Não presente

Resumo: O paciente não apresenta sintomas de calafrios ou sensação de frio com tremores.

- Negativo para calafrios
- Ausência de sensação de frio com tremores

Obstipação – Grau 1 (Confirmado)

Resumo: O paciente apresenta constipação desde 5 dias após o início do tratamento, com esforço defecatório ligeiro. Utiliza lactulose três vezes ao dia sem necessidade de clisteres ou evacuação manual.

- Constipação há 5 dias
- Esforço defecatório ligeiro
- Uso de lactulose 3x/dia

Justificação do LLM: O grau foi classificado como 'GRADE 1' devido à presença de sintomas intermitentes relacionados à dificuldade na defecação que iniciou após o início do tratamento e a utilização pontual de lactulose. Não há evidência de limitações significativas nas atividades instrumentais da vida diária ou necessidade de evacuação manual.

Dor no trato urinário – Grau 1 (Confirmado)

Resumo: O paciente apresenta uma dor ligeira no trato urinário há 3 dias que interfere com a sua rotina, mas não causa dores no bexiga, uretra, flanco, abdómen inferior ou lombar e não afeta significativamente a mobilidade, sono, alimentação ou autocuidado.

- Dor ligeira no trato urinário há 3 dias
- Interfere com a rotina
- Sem dores no bexiga, uretra, flanco, abdómen inferior ou lombar
- Não afeta significativamente a mobilidade, sono, alimentação ou autocuidado

Justificação do LLM: A dor é classificada como GRADE 1 pois é ligeira e interfere com as atividades instrumentais da vida diária, mas não limita o autocuidado.

Disestesia laringofaríngea – Grau 2 (Rever)

Resumo: O paciente apresenta uma sensação de desconforto na laringe e parte superior do esófago há 6 horas após o tratamento, com leve dificuldade respiratória. Não há dificuldades significativas nas atividades diárias ou necessidade de ansiolítico ou tratamento urgente.

- Sensação de desconforto na laringe e parte superior do esófago
- Dificuldade leve em respirar
- Não afeta as atividades diárias
- Sem necessidade de ansiolítico ou tratamento urgente

Justificação do LLM: CLASSIFICAÇÃO: GRADE 1 - Sintomas ligeiros com leve sensação de dificuldade respiratória, sem dispneia, ansiedade ou dificuldades significativas nas atividades diárias.

Histórico de Conversa

Assistente: Sentes atualmente algum formigueiro ou dormência?

Paciente:
Agora não, senti durante os primeiros 5 dias, sobretudo quando tocava em objetos frios.

Assistente: Tem ocorrido episódios de fezes muito moles ou líquidas?

Paciente: Não.

Assistente: Tem sentido náuseas ou sentiu enjoo?

Paciente: Senti enjoos durante a primeira semana.

Assistente: Quando começaram as náuseas?

Paciente: No dia seguinte ao inicio de otratamento.

Assistente:
Como descreve a frequência com que ocorrem as náuseas (ex: raramente, várias vezes por dia)?

Paciente: Senti-me enjoado durante a maior parte do dia.

Assistente:
Como costuma acontecer o enjoamento? É algo que sente raramente ou várias vezes por dia?

Paciente: Várias vezes por dia.

Assistente:
Tem perdido apetite recentemente? Isto tem afectado como se alimentas?

Paciente:
Sim, durante a primeira semana do tratamento, ao final de uma semana volta ao normal.

Assistente:
Tem comido ou bebido menos do que o habitual devido às náuseas?

Paciente: Sim.

Assistente: Tem episódios de vómitos?

Paciente: Não.

Assistente: Tem alguma dor, ferida ou inflamação na boca?

Paciente: Sim, tive.

Assistente: Quando começaram as feridas?

Paciente: Ao 4º dia após o inicio do tratamento

Assistente: Qual é a intensidade da sua dor? (leve, moderada ou intensa)

Paciente: moderada

Assistente:
Tem dificuldade em comer ou beber normalmente devido à mucosite?

Paciente:
sim, não conseguia mastigar. So consegui ingerir gelatina, sopa fria e liquidos durante 1 semana

Assistente: Tem sentido algum fadiga ou cansaço extremo?

Paciente: não

Assistente: Tem perdido cabelo mais do que de costume?

Paciente: não

Assistente: Tem feridas elevadas com pus na pele?

Paciente: não

Assistente: Tem febre?

Paciente: Sim.

Assistente: Quando começou a febre?

Paciente: há dois dias

Assistente:
Qual é o valor mais alto medido e por quanto tempo ele se manteve?

Paciente: 38.3ºC, 2 dias

Assistente: Tem a febre ocorrido uma vez ou em vários episódios?

Paciente: episodios de 8/8h

Assistente: Tem dor de cabeça?

Paciente: não

Assistente: Tem calafrios ou sente-se muito frio com tremores?

Paciente: não

Assistente: Tem dificuldades em defecar?

Paciente: sim

Assistente:
Quando começou o seu sintoma? É um problema recente ou regular?

Paciente: 5 dias após o inicio do tratamento

Assistente:
Como descreve o esforço ou a dor ao defecar? É ligeiro, persistente ou grave?

Paciente: ligeiro

Assistente:
Tem utilizado amaciadores das fezes, laxantes, clisteres ou precisou de evacuação manual?

Paciente: Sim, lactulose 3x dia

Assistente:
Tem dificuldades em cozinhar, limpar ou deslocar-se em casa devido ao seu sintoma?

Paciente: não

Assistente:
Tem dificuldades em realizar tarefas básicas como higiene, banho ou vestir-se?

Paciente: não

Assistente: Gostaria de reportar um novo sintoma?

Paciente: Dificuldade em urinar

Assistente: Por favor, descreva o sintoma com o máximo de detalhe.

Paciente: Dor ao urinar e urinar pequenas quantidade varias vezes ao dia.

Assistente: Sentes alguma dor ou desconforto no trato urinário?

Paciente: sim

Assistente: Quando começou a dor?

Paciente: ha 3 dias

Assistente:
Como está a dor? É ligeira, moderada ou intensa, e interfere com a sua rotina?

Paciente: ligeira, interfere

Assistente: Tem dores no bexiga, uretra, flanco, abdómen inferior ou lombar?

Paciente: não

Assistente:
A dor interfere com a sua mobilidade, sono, alimentação ou autocuidado?

Paciente: não

Assistente: Gostaria de reportar um novo sintoma?

Paciente: Sim

Assistente: Por favor, descreva o sintoma com o máximo de detalhe.

Paciente:
Dificuldade em articular palavras e sensaçao de "garganta presa" no final do dia do tratamento.

Assistente:
Sentes algum desconforto persistente na garganta, laringe ou parte superior do esófago?

Paciente: Senti

Assistente:
Quando começou a sensação anormal ou desconfortável na garganta?

Paciente: 6h apos o tratamento

Assistente:
Tem apenas desconforto ou ansiedade leve, ou sente dificuldade em respirar ou engolir?

Paciente: Leve sensaçao de dificuldade a respirar

Assistente:
Tem alguma dificuldade em comer, falar, sair de casa ou trabalhar?

Paciente: Falar

Assistente: Tem alguma coisa que te dificulta comer, lavar-te ou falar?

Paciente: Nao

Assistente:
Tem necessidade de ansiolítico, observação médica ou tratamento urgente?

Paciente: Não

Assistente: Gostaria de reportar um novo sintoma?

Paciente: Não.

Assistente: Obrigado. Vamos finalizar a sessão.