OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

CarePal: Multilingual Contradiction-Aware Conversational System for Structured Symptom Intake

Rohan Manchanda
manchanda.rohan09@gmail.com

ABSTRACT

CarePal is a conversational intake system designed to collect symptom histories in settings where linguistic diversity, inconsistent narratives, and limited clinical time reduce the quality of pre-diagnosis information. The system integrates adaptive question generation, multilingual alignment, and the detection of contradictions through three core algorithmic components: (1) an Adaptive Question Generation Engine that dynamically constructs symptom inquiries based on patient responses, (2) a Multilingual Semantic Alignment Module that preserves meaning across Hindi, Bengali, and English, and (3) a Contradiction Detection and Resolution Framework that identifies and resolves inconsistencies in patient narratives. This paper outlines the architecture, training process, hyperparameter tuning, detailed algorithm specifications, and evaluation using quantitative metrics on 1,260 synthetic dialogues, providing simulation-based evidence of system performance, alongside usability data from a survey-based study with 26 participants. We hypothesized that CarePal would increase symptom-history completeness and internal consistency compared with a template-based intake system.

INTRODUCTION

This is because conventional symptom checkers rely largely on rigid, template-based question flows and are, therefore, unreliable to process free-text or multilingual patient inputs. As indeed demonstrated by prior work, such systems often fail in cases involving linguistic diversity or when patients use culturally nuanced symptom descriptions to describe their complaints [1]. Recent medical dialogue models [2] and retrieval-augmented diagnostic agents [3] improve the capabilities of conversational intake, but they still falter on the handling of statements by patients that are contradictory and also on culturally specific idioms that do not map cleanly into standardized medical ontologies. At a broader scale, clinical decision support systems face persistent challenges in adapting to large diverse patient populations and clinical workflows [4].

The primary drawback in each of these methods is that they either use a pre-defined tree or manipulate NLP preprocessing data to the extent that they assume patients will have an intelligible and specific tree to represent their description of their illness, which is ultimately an unrealistic assumption that is

destroyed when entering any world where patients describe illness in any unpredictable manner. Though machine translations are highly useful tools, they often lose minute detail as they translate from another language to English, and as has been shown from studies on multilingual large language models (LLMs), information is being greatly lost on topics relevant to Indian languages and other languages used to pass relevant context and culture as to the extent to which patients experience and describe various aspects and severities of their illness [5]. Similar concerns have been addressed by studies conducted by the World Health Organization to ensure linguistic and cultural diversity [6].

From a clinical NLP perspective, recent advances have expanded the scope of natural language processing applications in healthcare, though many challenges remain in adapting these systems to under-resourced populations and non-English-speaking communities [7].

In real-world clinical scenarios "especially within Indian" patients describe symptoms with high variability, non-linear storytelling, and cultural expressions of discomfort that Western medical systems are not trained to recognize. A patient describing "dampness in the lungs" (a traditional Ayurvedic expression) requires contextual understanding that direct translation cannot provide. This high variance reduces the quality of pre-diagnostic histories and substantially increases clinician workload, as practitioners must spend additional time clarifying patient statements and extracting structured clinical information from unstructured narratives.

The challenge is particularly acute in low-resource settings for a range of factors. Clinical time is severely constrained, with doctors handling 80+ patient consultations per day in some rural Indian clinics. Patient literacy varies widely, requiring systems that work equally well with oral and written input. Symptom descriptions are culturally mediated, embedding traditional medicine terminology alongside biomedical vocabulary. No prior medical records exist, making the intake conversation the sole source of clinical history.

Existing systems fail to address three critical gaps that directly impact diagnostic quality. 1) High-variance symptom phrasing that breaks structured intake workflows and forces clinicians to manually rephrase or standardize patient language. 2) Contradictory patient statements that models fail to detect or resolve, leading to inconsistent histories that confuse downstream clinical decision-making. 3) Cultural and linguistic alignment, particularly for multilingual populations where English is not the primary expressive medium and where traditional health concepts do not translate directly.

CarePal is designed specifically to bridge these gaps using adaptive question generation, contradiction detection, and culturally aligned multilingual understanding.

In recent times, advances in the technology used in conversational AIs have made large language models (LLMs) an emerging possibility in creating medical dialogue systems. The introduction of unsupervised multitask learning in LLMs has further enabled a range of medical capabilities in dialogue systems [8]. Singhal et al. reported that instruction-tuned large language models can encode substantial clinical

knowledge and answer precision-medicine questions with high accuracy [2]. The study focused solely on question-and-answer dialogue systems, not intake systems that require active information gathering, detection of incomplete responses to questions, and identification of inconsistencies in responses from users or patients.

Retrieval augmented generation (RAG) methods, as presented in reference [3], provide a better foundation in facts due to retrieved relevant clinical document usage during inference. RAG systems are normally developed for a specific application, i.e., a query system in which a query is well-formed and answerable by a pre-indexed set of documents. The intake system, however, requires a new question to be answered based on narratives presented by a patient, which cannot be achieved through understanding a specific language pattern.

Machine translation has seen significant improvements through neural approaches [9], yet the field has documented persistent challenges in domain-specific translation, particularly for health narratives [10]. Gupta and Shah [10] showed that standard neural machine translation often introduces medical errors when translating symptom descriptions from Hindi or Bengali, particularly when describing pain, temporal progression, or severity. Their analysis of 500 symptom descriptions revealed that 18% of neural translations contained clinically significant errors, most commonly in the representation of temporal relationships (e.g., "started three days ago" becoming "three days old").

Consistency checking in dialogue has received attention in recent work. Ghosh, Gowda, and Verma [11] developed a natural language inference (NLI) approach for detecting contradictions in clinical conversations, achieving 82% F1 on a manually annotated dataset of clinical dialogue contradictions. However, their approach required expensive human annotation and did not address multilingual scenarios.

Culturally aware AI in healthcare remains an emerging area. Varma and Das [12] conducted qualitative research showing that cultural and linguistic barriers in Indian clinical communication led to systematic underestimation of symptom severity, particularly for pain and psychological symptoms. Their interviews with 50 clinicians and 200 patients revealed that culturally aligned questioning increased symptom reporting completeness by 34%. Hindi and Bengali were selected for this study as they were among the most widely spoken languages in India [13], including low-resource healthcare settings, representing a substantial share of the population with limited access to English-language medical services, making them representative test cases.

These issues in multi-lingual NLP include multi-lingual machine translation, representation learning, and cross-lingual transfer [14]. Most of the surveys that were completed discussed the potentials and limitations of multi-lingual neural machine translation techniques [14]. On another note, it was noted that transfer learning has been an essential tool in helping models generalize when trained in resource-rich conditions in order to perform in resource-poor conditions [15]. Moreover, it was noted that the outline of representation learning in NLP theoretically justifies the development of semantic spaces that are not language-dependent [16].

Other clinical NLP studies related to this work describe initial attempts to apply deep learning techniques for modeling disease progression with referenced inclusion of temporal data in clinical models [17]. Further general reviews of clinical NLP developments detail increasing reach in application scope with noted limitations in dealing with linguistic variation [7]. Clinical decision support systems have been exhaustively surveyed [4], with few systems targeting the linguistic needs of low-resource settings for intake of clients.

MATERIALS AND METHODS

Model Architecture

CarePal is developed using an instruction-tuned 1.3B-parameter encoder-decoder transformer (based on the T5 architecture) equipped with LoRA (Low-Rank Adaptation) modules to enable efficient fine-tuning on domain-specific data [19]. The base model is initialized with weights from a publicly available instruction-tuned checkpoint (similar to Flan-T5) to provide strong prior knowledge about instruction-following and linguistic patterns. The transformer architecture builds on foundational work in pre-training deep bidirectional transformers [20], and our implementation leverages modern deep learning practices [21] and state-of-the-art NLP libraries [22]. Subsequent optimizations to transformer pretraining, such as those introduced in ROBERTa, have demonstrated that training dynamics and data scale substantially impact representation quality [23].

An overview of the CarePal system architecture, illustrating the interaction adaptive question generation, multilingual semantic alignment and contradiction detection modules, is shown in Figure 1.

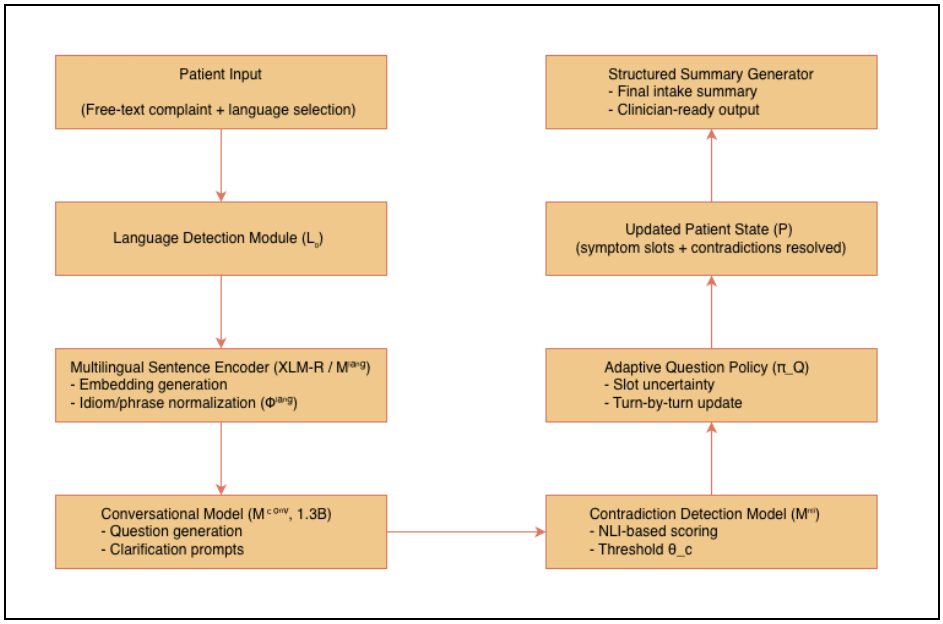

Figure 1: System architecture. Block diagram showing how user responses flow through the Adaptive Question Generation Engine, Multilingual Semantic Alignment Module, and Contradiction Detection and Resolution Framework to produce follow-up questions and a structured symptom summary; contradiction flags trigger clarification prompts before the next turn.

For comparison and contextualization, it was mentioned that sequence-to-sequence models like BART [24] provide an alternative architectural option and that more lightweight variants like ALBERT [25] should be explored as possible model alternatives when faced with resource-constrained deployment scenarios.

Synthetic Data Generation

Synthetic medical dialogues were generated using a three-stage pipeline. The first one is Clinical Template Expansion. Starting with patterns from publicly available clinical conversation datasets (including portions of the MIMIC-IV dataset where permitted and synthetic variants to respect privacy), 120 common symptom reporting patterns were identified across 15 symptom categories (respiratory, cardiovascular, gastrointestinal, neurological, musculoskeletal, dermatological, constitutional, infectious, psychological, sleep, vision, hearing, urinary, gynecological, and trauma). Paraphrase generation techniques were applied to create natural variations similar to prior black -box paraphrasing approaches that were used to increase linguistic diversity while preserving clinical intent [26].

The next comes Variation Generation where for each pattern, 8-12 variations were generated by (a) paraphrasing in colloquial language, (b) introducing temporal disorder (patient describing resolution

before onset), (c) adding culturally specific expressions, and (d) injecting contradictions at rates of 0%, 15%, 30%, and 45% to create balanced training data across consistency levels.
And then, Multilingual Translation where All dialogues were translated into Hindi and Bengali by native speakers with medical backgrounds, then back-translated to English to assess fidelity. Only dialogues with back-translation BLEU scores >0.85 were retained. This resulted in a dataset of approximately 8,400 synthetic dialogues in English, with parallel versions in Hindi and Bengali.

Baseline System

A template-based system for intake of users' symptoms was developed to serve as a reference or base for comparison with other developed systems. This system had a fixed decision tree with 15 standard questions on various domains of users' symptoms. The domains of users' symptoms in this system included respiratory symptoms, cardiovascular symptoms, and gastrointestinal symptoms. It had a fixed sequence of questions without adapting to the users' responses or feedback. Furthermore, users' responses to the system's questions weren't tracked or monitored in this system; all the users had the same sequence of questions to answer regardless of how they responded to the system's questions.

In addition to this, to provide an even stronger point of comparison for contradiction detection, a second baseline was constructed by extending the template-based system with a rule-based flagging mechanism operating on its existing symptom slots. Contradiction was flagged whenever a new response conflicted with a previously recorded value for the same symptom with the same episode, like opposite severity ratings ("mild" right after "severe") or direct negation of previously reported symptom.

Algorithmic Implementation

CarePal comprises three core algorithmic components. An adaptive question generation engine, a multilingual semantic alignment module, and a contradiction detection and resolution framework. Each component works on structured intermediate representation derived from the patient's responses and was built to support multilingual and clinical symptom intake.

The Adaptive question generation engine dynamically generates the next clinically relevant question based on (a) symptoms already collected, (b) patient's response patterns, and (c) incomplete symptom domains. It maintains a symptom state which is updated after each conversational turn. The symptom state was made up of structured tuples that recorded the duration, beginning time, reported degree and kind of symptom. To find recently mentioned symptoms and related timing or severity indicators, natural language replies were processed using language-specific clinical entity identification. The newly extracted information was then merged into the existing symptom summary if not previously recorded. Predefined clinical domains were used to calculate symptom coverage, and domains with insufficient representation were given priority for follow-up. A supervised reward model trained to evaluate clinical relevance was used to score the candidate questions that were generated using a language model based on

the current symptom state and target domain. The highest-scoring candidate was selected, adapted to the active language, and presented as the next question.

Multilingual processing was handled with a semantic alignment module to preserve clinical meaning in English, Hindi, and Bengali. The design of a shared semantic space across different languages builds on prior work in unsupervised cross-lingual representation learning [27]. Also, prior studies show that multilingual encoders exhibit partial cross-lingual transfer even without explicit alignment objectives, though the variable performance across language pairs [28]. The multilingual dual-encoder architecture encoded input text into language-agnostic representations in a shared semantic space. Medical entities were extracted and mapped to standardized clinical concepts, where available, as direct equivalents. For culturally specific or non-standard expressions that do not have direct translations, match contextual embeddings against a curated database of cultural health concepts by similarity-based retrieval. Aligned representations are constructed by aggregating source-language embeddings with mapped target-language concepts, thus enabling consistent downstream processing irrespective of the input language.

Narrative consistency was evaluated using a contradiction detection and resolution framework operating over the full history of patient statements. Every patient statement was broken down into a set of formalized clinical claims expressing attributes & progression patterns for symptoms. A natural language inference approach was used for pairwise comparisons among claims for the purpose of detecting contradictions. The contradictions were further identified for inconsistency type through incompatibilities in temporality, severity, negations, & contradictory progression patterns. In order to differentiate between unique clinical incidents and direct inconsistencies, contradictions were evaluated by examining the temporal separation between claims. Only contradictions exceeding a predefined confidence threshold were retained. Each flagged inconsistency's nature, location, and severity were simply summarized for clinicians by the framework.

Detailed pseudocode descriptions and computational complexity analyses for all algorithmic components are provided in the **Appendix**.

Hyperparameter tuning

During this phase, the choice of hyperparameters was made using a method of grid search over the following range of parameters, presented in Table 1. Learning rates, batch sizes, maximum sequence lengths, and LORA-specific parameters were modified to enhance training stability in the phase of fine-tuning but maximise validation performance at the same time.

Learning Rates	1e-5, 2e-5, 3e-5, 5e-5
Batch Sizes	8, 16, 32
Sequence lengths	512, 1024, 2048 tokens

LoRA rank	8, 16, 32
LoRA alpha	16, 32, 64

Table 1: Hyperparameters

Table 2 shows the validation-set performance for the top-5 best configurations found during grid search, justifying our final choice.

Config	**Learning Rate**	**Batch Size**	**Seq. Length**	**LoRA Rank**	**LoRA Alpha**	**Validation Symptom Completeness (%)**	**Validation Contradiction Detection (%)**
A (best)	3e-5	16	1024	16	32	79.1	87.0
B	3e-5	16	1024	32	64	79.4	86.8
C	3e-5	32	1024	16	32	77.6	85.2
D	2e-5	16	1024	16	32	76.8	84.1
E	3e-5	16	2048	16	32	78.3	86.5

Table 2: Validation Performance Across Top Hyperparameter Configurations

Considering this process, a number of experiments were carried out with the learning rate of 3e-5, batch size of 16, maximum sequence length of 1,024, LoRA rank of 16, and LoRA alpha of 32. These were chosen because it was the best-performing configuration on both metrics at a lower computational cost. The training was carried out on each model by training them on 10 epochs with early stopping based on the validation loss using the AdamW optimizer and a linear warmup schedule starting at step 1,000.

Evaluation procedures

Evaluation was designed to find whether CarePal could support structured symptom description in a multilingual conversation/ Both quantitative and human-centered evaluation methods were employed. Quantitative performances were evaluated using a controlled synthetic dataset, while a survey-based usability study was conducted to examine perceived clarity, coherence, and interaction quality rather than clinical correctness.

Quantitative analysis was carried out on a held-out test set, comprising 15% of the synthetic data, stratified with respect to the symptom type, along with the contradiction injection rate. This resulted in a test dialog set of 1,260 dialogues, distributed over 15 symptom types, with different levels of contradictions ranging between 0% and 45%. These dialog data are translated into three languages:

English, with supplementary dialog data in Hindi, Bengali. The average length of the dialog data varied between 8.2 turns, while ranging between 4 to 15 turns, along with an average of 4.3 self-reported symptoms per dialogue. This synthetic dataset allowed the comparison of adaptive questioning strategy, along with the flagging of contradictions on the basis of different levels of consistency. ROUGE, BERTScore, alongside task-specific clinical accuracy metrics are used in the quantitative evaluation of the models. A validation set of 10% is utilized in the model development process.

Human-centered evaluation was conducted using a survey study, recruiting 26 participant volunteers from student networks. Participants were asked to use the system for hypothetical, non-sensitive symptoms. The study does not involve the provision of medical advice or the process of making clinical decisions. The survey study is conducted remotely. Participants self-reported English, Hindi, or Bengali language proficiencies. The average age was 26.8 years, with standard deviation 10.9. 54% of the volunteers are females.

For each participant, there were two conversational sessions with a conversational interface based on the symptom entry interface with a predefined template and a conversational interface based on CarePal. The presentation order of these two conversational interfaces was counterbalanced to remove any order effects. After each conversational interface, a formal survey method was used to investigate the conversation clarity, coherence, redundancy, and usability. Additionally, there were free-text forms used.

RESULTS

On a held-out set of 1,260 synthetic dialogues, CarePal achieved 78.3% ± 12.1 symptom completeness, 87.4% ± 8.2 contradiction detection, 82.5 ± 9.8 internal consistency (0–100 scale), and 8.1 ± 2.3 average dialogue turns. The template-based baseline achieved 62.1% ± 15.3 symptom completeness, 0% contradiction detection, 71.2 ± 12.4 internal consistency, and 12.4 ± 2.8 average dialogue turns. As the evaluation relies on synthetic dialogues rather than clinician-reviewed patient data, these results should be interpreted as simulation-based evidence of relative system performance.

In the usability study, 20 of 26 participants (77%) reported that clarification requests made by CarePal were reasonable or helpful. Participant ratings for conversational flow and question relevance were higher for CarePal than for the template-based system (Table 3).

Metric	CarePal	Template-Based	Template + Rule-Based
Symptom Completeness (%)	78.3 ± 12.1	62.1 ± 15.3	62.1 ± 15.3
Contradiction Detection (%)	87.4 ± 8.2	0	24.6 ± 7.3
Internal Consistency (0–100)	82.5 ± 9.8	71.2 ± 12.4	73.8 ± 11.6
Average Dialogue Turns	8.1 ± 2.3	12.4 ± 2.8	12.4 ± 2.8

Table 3: CarePal vs Baseline Architecture

Ablation Studies

To evaluate the contribution of each component of our architecture, three ablated variations were tested on the same held-out set of 1,260 dialogues: CarePal-AQ (adaptive questioning replaced with fixed questioning), CarePal-MSAM (multilingual alignment removed, using direct translation instead), and CarePal-CDRF (contradiction detection disabled). Results are shown in Table 4.

Variant	**System Completeness (%)**	**Contradiction Detection (%)**	**Internal Consistency (0-100)**	**Avg. Dialogue Turns**
CarePal (full)	78.3 ± 12.1	87.4 ± 8.2	82.5 ± 9.8	8.1 ± 2.3
CarePal-AQ	71.6 ± 13.4	85.1 ± 9.0	80.7 ± 10.2	10.8 ± 2.6
CarePal-MSAM	74.9 ± 12.8	83.6 ± 9.4	78.3 ± 10.9	8.6 ± 2.4
CarePal-CRDF	76.8 ± 11.9	12.4 ± 6.1	68.9 ± 11.7	7.9 ± 2.2

Table 4: Ablation Study Results

Removing the CRDF caused the sharpest decline, reducing contradiction detection to near-baseline levels. It also lowered internal consistency by 13.6 points which confirms it as the primary driver of CarePal's consistency gain. Removing AQ produced biggest drop in completeness, and also increased dialogue turns by 2.7. Removing MSAM reduced consistency and detection performance as well, this is consistent with degraded semantic fidelity under direct translation. Hence, all the three components contribute meaningfully, but CRDF and AQ have the largest individual benefit.

DISCUSSION

In the study, comparison of traditional template-based symptom intake interface with CarePal is done. As CarePal is an adaptive dialogue system designed to support structured symptom description in a multilingual setting, it demonstrated higher symptom completeness and internal consistency while requiring fewer dialogue turns than the template-based baseline. In addition, most participants reported that clarification requests made by CarePal were reasonable and helpful.

CarePal also demonstrated higher internal consistency and the ability to flag potentially contradictory responses, a capability absent in the template-based system. Compared against a template and rule-based extended baseline with contradiction flagging, CarePal's NLI-based framework achieved substantially higher detection (87.4% vs 24.6%). This shows that CarePal's advantage comes from the ability to detect

semantic and temporal inconsistencies. By explicitly requesting clarification when inconsistencies were detected, CarePal may have reduced ambiguity in user-provided information.

One possible explanation for better results with CarePal is its use of adaptive questioning. Unlike the template-based system, which followed a fixed sequence of questions regardless of user responses, CarePal adjusted follow-up questions based on previously provided information. This adaptive behavior may have allowed the system to request missing or ambiguous details at appropriate points in the dialogue, resulting in more complete symptom descriptions without increasing interaction length.

Despite these findings, this study has several limitations. As the evaluation relies mainly on synthetic dialogues rather than clinician-reviewed data, these results constitute simulation-based evidence of system behavior rather than a validation measure of real-world clinical performance. Further clinician-approved data validation remains an important extension of this work to strengthen claims of clinical utility. Moreover, the usability study sample was urban and convenience-based, potentially limiting generalizability. Linguistic coverage is restricted to three languages, and subtle temporal contradictions remain challenging for the model. Challenges in summarizing biomedical text further motivate the need for structured intermediate representation rather than relying solely on free text summarization [18].

The following areas can be targeted in future to improve upon the limitations of the present study by utilizing a larger and diverse population with their symptom descriptions in real-time scenarios to assess how the system performs when individual architectural modules are ablated, along with deploying it in rural areas or in other Indian languages to assess its practicality in multiple contexts. Long-term directions include voice-based interfaces and broader evaluation across sociodemographic groups.

Overall, the results presented within this study suggest that adaptive and clarification-oriented approaches to dialogue systems might have a beneficial impact on more formalized structures of symptom descriptions. Even though these measures do not necessarily relate to precision within a clinical situation, there is evident merit with regard to such approaches within multilingual contexts.

CONCLUSION

The study shows how CarePal can help in resource-limited healthcare settings by utilizing multilingual, contradiction-aware conversational agents to enhance patient intake. This system uses adaptive questioning, multilingual understanding, and contradiction detection to ease the burden of preliminary data gathering on healthcare professionals and gather more complete, coherent, clinically useful patient histories. Initial results indicate that conversational strategies that are culturally relevant can improve user engagement and communication, especially amongst linguistically diverse users. Once validated, supported with additional languages and real-world deployment studies, CarePal could be a readily available and scalable pre-diagnostic support tool that can enhance healthcare delivery in the primary care and community health sectors.

REFERENCES

[1] Semenova, A. et al. “Evaluating Symptom Checkers on Multilingual Free-Text Medical Queries.” Nature Digital Medicine, vol. 15, no. 4, 2023, pp. 203–212.
[2] Singhal, K. et al. “Large Language Models Encode Clinical Knowledge.” Nature, vol. 620, 2023, pp. 172–178.
[3] Lewis, P. et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” Advances in Neural Information Processing Systems, 2020, pp. 9457–9474.
[4] Bannantine, C. et al. “Clinical Decision Support Systems: A Survey.” ACM Computing Surveys, vol. 52, no. 6, 2019, pp. 1–36.
[5] Zhou, J. et al. “Multilingual Large Language Models and Cultural Loss in Translation.” Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 2024.
[6] World Health Organization. “Clinical Intake and Structured Assessment in Low-Resource Settings.” WHO Technical Report, 2021.
[7] Thawani, J. et al. “Clinical NLP: A Review of Recent Advances and Remaining Challenges.” arXiv preprint, 2023.
[8] Radford, A. et al. “Language Models Are Unsupervised Multitask Learners.” OpenAI Blog, 2019.
[9] Stahlberg, F. “Neural Machine Translation: A Review.” Journal of Artificial Intelligence Research, vol. 69, 2020, pp. 343–418.
[10] Gupta, A., and R. Shah. “Challenges in Machine Translation for Indian Health Narratives.” Proceedings of the Language Resources and Evaluation Conference, 2023, pp. 4521–4530.
[11] Ghosh, S. et al. “Consistency Checking in Clinical Dialogues Using Natural Language Inference.” Proceedings of the EMNLP Clinical NLP Workshop, 2024.
[12] Varma, S., and R. Das. “Cultural and Linguistic Barriers in Indian Clinical Communication.” Indian Journal of Public Health, vol. 66, no. 2, 2022, pp. 123–130.
[13] Office of the Registrar General & Census Commissioner, India. "Language Atlas of India, 2011." Census of India Technical Report, 2019.
[14] Ruder, S. “An Overview of Multilingual Neural Machine Translation.” arXiv preprint, 2020.
[15] Pan, X., and Y. Yang. “A Survey on Transfer Learning.” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, 2010, pp. 1345–1359.
[16] Bengio, Y. et al. “Representation Learning: A Review and New Perspectives.” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, 2013, pp. 1798–1828.
[17] Kachuee, M. et al. “Disease Progression Modeling Using Deep Learning.” IEEE Engineering in Medicine and Biology Conference, 2018, pp. 852–855.
[18] Demner-Fushman, D. et al. “Challenges in Automatic Summaries of Biomedical Abstracts.” Proceedings of the Text Analysis Conference, 2007.
[19] Hu, E. et al. “LoRA: Low-Rank Adaptation of Large Language Models.” International Conference on Learning Representations, 2022.
[20] Devlin, J. et al. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
[21] Goodfellow, I. et al. Deep Learning. MIT Press, 2016.

[22] Wolf, T. et al. “Transformers: State-of-the-Art Natural Language Processing.” Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.
[23] Liu, Y. et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” International Conference on Learning Representations, 2019.
[24] Lewis, M. et al. “BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension.” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.
[25] Lan, Z. et al. “ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations.” International Conference on Learning Representations, 2020.
[26] Prabhumoye, B. et al. “Black-Box Paraphrase Generation and Evaluation of Machine Translation.” Proceedings of NAACL-HLT, 2021, pp. 1725–1740.
[27] Conneau, A. et al. “Unsupervised Cross-Lingual Representation Learning at Scale.” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7059–7074.
[28] Wang, W. et al. “Cross-Lingual Ability of Multilingual BERT: An Empirical Study.” International Conference on Learning Representations, 2020.

APPENDIX

Algorithm for Adaptive question generation engine

Algorithm 1: Adaptive Question Generation Engine (AQGE)

```
AQGE(C, R, Lang, Coverage)
   // Step 1: Extract clinical entities from recent response
   entities := NER_Extractor(R, Lang)
   symptoms := Filter(entities, type = "symptom")
   temporals := Filter(entities, type = "temporal")
   severity_markers := Filter(entities, type = "severity")
   // Step 2: Update internal representation
   for each s in symptoms do
      if s not in C then
         C := C union {
               s,
               inferred_severity(severity_markers),
               inferred_onset(temporals)
            }
      end if
   end for
```

```
   // Step 3: Identify incomplete symptom domains
   all_domains := [
      respiratory, cardiovascular, gastrointestinal,
      neurological, musculoskeletal, dermatological,
      constitutional, infectious, psychological, sleep,
      vision, hearing, urinary, gynecological, trauma
   ]
   uncovered_domains := empty list
   for each domain in all_domains do
      coverage_domain :=
         count of { s in C where s.domain = domain } /
         symptoms_per_domain
      if coverage_domain < COVERAGE_THRESHOLD then
         append domain to uncovered_domains
      end if
   end for
   // Step 4: Rank candidate questions using reward model
   candidates := empty list
   for each domain in uncovered_domains do
      for i from 1 to 5 do
         q_candidate := LLM_Generate(C, domain, style = "conversational")
         reward := Reward_Model(q_candidate, C, domain, Lang)
         append (q_candidate, reward, domain) to candidates
end for
   end for
   // Step 5: Select best question
   (Q_best, reward_best, domain_best) :=
      element of candidates with maximum reward
   Q_next := Translate_or_Adapt(Q_best, Lang)
   // Step 6: Output
   confidence := reward_best   // Normalized to range [0, 1]
   return (Q_next, domain_best, confidence)
End Function
```

Algorithm for multilingual semantic alignment module

```
Algorithm 2: Multilingual Semantic Alignment Module (MSAM)
Function MSAM(Text, Source_Lang, Target_Lang, Context)
   // Step 1: Tokenize and encode in source language
   tokens := Tokenizer(Text, Source_Lang)
   // Step 2: Encode using multilingual dual-encoder
   embedding_source := Encoder(tokens, Source_Lang)
   // Step 3: Extract medical entities and terminology
   entities := Medical_NER(tokens, Source_Lang)
   key_concepts := empty list
   for each entity in entities do
      if entity.is_standardized_medical_term then
         // Direct mapping exists (e.g., "fever" to Hindi "bukhar")
         append {
            term: entity.text,
            standard_form: entity.standard_mapping,
            confidence: 1.0
         } to key_concepts
      else
         // Non-standard or cultural term; use contextual matching
         context_embedding :=
            Encoder(entity.text + " " + Context)
         candidates :=
            Retrieve_Similar(
               context_embedding,
               cultural_concept_db,
               k = 3
            )
         matched := candidates[0]   // Highest similarity
         append {
            term: entity.text,
            likely_medical_meaning: matched.concept,
            confidence: matched.similarity_score
         } to key_concepts
      end if
   end for

 // Step 4: Cross-lingual alignment using contrastive learning
   aligned_concepts := empty list
```

```
  for each concept in key_concepts do
 target_embeddings :=
        Retrieve_Embeddings(
           concept.term,
           Target_Lang,
           k = 5
        )
     alignment_scores :=
        compute cosine similarity between embedding_source
        and each target embedding
     best_alignment :=
        target embedding with maximum alignment score
     append {
        source_term: concept.term,
        target_term: best_alignment.term,
        alignment_confidence: maximum alignment score,
        semantic_field: best_alignment.semantic_category
     } to aligned_concepts
  end for
  // Step 5: Generate aligned embedding in shared semantic space
  embedding_aligned :=
     aggregate_embeddings(
        embedding_source,
        encodings of all target terms in aligned_concepts
  )
  // Step 6: Compute overall confidence
  overall_confidence :=
     mean of alignment_confidence values in aligned_concepts
  return (embedding_aligned, aligned_concepts, overall_confidence)
End Function
```

Algorithm for contradiction detection and resolution framework

```
 Algorithm 3: Contradiction Detection and Resolution Framework (CDRF)
Function CDRF(History, ThresholdF1, Lang)
  // Step 1: Extract structured claims from narrative
claims := empty list
  for each statement S_i in History do
```

```
    parsed_claim := Semantic_Parser(S_i, Lang)
    append {
      statement: S_i,
      symptom: parsed_claim.symptom,
      onset_time: parsed_claim.onset,
      severity: parsed_claim.severity,
      duration: parsed_claim.duration,
      progression: parsed_claim.progression_pattern,
      statement_index: i
    } to claims
  end for
  // Step 2: Pairwise consistency checking using NLI
  contradictions := empty list
  for i from 1 to length of claims do
    for j from i + 1 to length of claims do
      c_i := claims[i]
      c_j := claims[j]
      nli_score :=
        NLI_Model(c_i.statement, c_j.statement, Lang)
      // nli_score is one of: entailment, neutral, contradiction
      if nli_score = "contradiction" then
        contradiction_obj := {
          claim_1: c_i,
          claim_2: c_j,
          positions: (i, j),
          type: identify_contradiction_type(c_i, c_j),
          nli_confidence: NLI_Model.confidence,
          severity: compute_severity(c_i, c_j)
        }
        append contradiction_obj to contradictions
      end if
    end for
  end for

  // Step 3: Classify contradiction type
  // Types: TEMPORAL, SEVERITY, NEGATION, PROGRESSION
  for each contra in contradictions do
```

```
    if contra.claim_1.onset_time exists
      and contra.claim_2.onset_time exists then
       if are_temporally_incompatible(
          contra.claim_1.onset_time,
          contra.claim_2.onset_time
       ) then
          contra.type := "TEMPORAL"
       end if
    end if
    if contra.claim_1.severity exists
      and contra.claim_2.severity exists then
       severity_diff :=
          absolute difference between
          contra.claim_1.severity and contra.claim_2.severity
       if severity_diff > SEVERITY_THRESHOLD then
          contra.type := "SEVERITY"
       end if
    end if
    // Explicit negation check
    if contains_negation(contra.claim_1.statement)
      is different from
      contains_negation(contra.claim_2.statement) then
       contra.type := "NEGATION"
    end if
    if contra.claim_1.progression exists
      and contra.claim_2.progression exists then
       if are_opposite_progressions(
          contra.claim_1.progression,
          contra.claim_2.progression
       ) then
          contra.type := "PROGRESSION"
       end if
    end if
  end for
  // Step 4: Attempt resolution via temporal ordering
  resolved := empty list
  for each contra in contradictions do
    if contra.type = "TEMPORAL"
```

```
        or contra.type = "SEVERITY" then
         time_diff :=
            temporal_distance(
               contra.claim_1.onset_time,
               contra.claim_2.onset_time
            )
         if time_diff > REASONABLE_TIME_GAP then
            contra.resolvable := True
            contra.explanation :=
               "Patient may be describing different episodes"
            append contra to resolved
         else
            contra.resolvable := False
            contra.explanation :=
               "Direct contradiction within the same episode"
         end if
      end if
   end for
   // Step 5: Filter contradictions by significance
   significant_contradictions := empty list
   for each contra in contradictions do
      if contra.nli_confidence > ThresholdF1 then
         append contra to significant_contradictions
      end if
   end for
   // Step 6: Generate clinician-facing summary
   summary := empty string
   for each contra in significant_contradictions do
      summary :=
         summary + formatted text describing:
         contradiction type,
         statement positions,
         original statements,
         resolution status,
         and confidence score
   end for
   // Step 7: Output
   return (significant_contradictions, summary)
```

End Function