International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(16s) 2026 pp. 1621-1656
https://doi.org/10.70917/ijcisim-2026-4688
Article

A SEMANTICALLY GROUNDED LLM ARCHITECTURE FOR INTENT-AWARE AND USER-ALIGNED HUMAN–COMPUTER INTERACTIONS

Barkha Shakeelahmad Kasab [1], Thota Siva Ratna Sai[2]

[1,2]P. K. University, Shivpuri, Madhya Pradesh, India
* Correspondence: barkhashahaji@gmail.com, sivaratnasai@gmail.com

Abstract: Large Language Models (LLM) have transformed the Human-Computer Interaction (HCI) as it offers open and adaptable communication via the context. However, the existing systems built on LLM also lack limits on semantic stability and intent understanding as well as optimising behaviour guided by the user. In this paper, the researcher proposes an integrated architecture, whereby, the contextual encoding is based on LLM, contrastive semantic reasoning network, explicit intent modeling layer, and a reinforcement learning (RL) framework along with a user experience (UX). The system employs a multi-step training, and it entails supervised multi-task learning to acquire representation grounding, joint optimization to acquire semantic intent alignment, and RL-based fine-tuning to acquire greater task success, satisfaction and trust calibration. The large-scale experiments suggest that the proposed architecture attains high semantic consistency, intent discrimination, conversational coherence, and UX levels, in comparison to powerful baselines of LLM. The need of each of the modules is proven by the ablation studies and the analysis of errors underlines the existing difficulties of dealing with ambiguous and multi-intent utterances. This study reflects a comprehensive, semantically based, and user-complementary approach to the methodology of the development of intelligent conversational systems and next-generation interactive AI interfaces.
Keywords: Semantic Grounding; Intent Modeling; Human–Computer Interaction; Reinforcement Learning

1. Introduction

Human-Computer Interaction (HCI) has developed since deterministic, command-based interfaces to adaptive conversational systems with the ability to understand natural language in context. One of the key recent developments in this direction was the fact that large-scale generative pretraining allows strong few-shot generalization and contextual adaptation on heterogeneous linguistic tasks [1]. The later scale experiments demonstrated that language models based on transformers could be used as general reasoning engines with emergent properties when the number of parameters and the size of data grow [2]. Massive scale pathway-style architectures also established that model scaling coupled with architectural parallelism has a significant positive influence on the depth of reasoning and cross-task transferability [3]. Then open foundation models were concerned with the efficient deployment and flexibility that enabled easier integration of the LLMs into the interactive application in a wider sense [4]. In a reaction to the limitations of real-time systems in reality Complementary architectural work attempted to have variations to conventional transformer mechanisms with better long-context stability and computational efficiency [5].

Even though scaling improved linguistic fluency and extent of reasoning, the practical use in the real-life situation of interaction was required due to the need to correspond to human intent and expectations. The human feedback reinforcing discovery (HFRL) experiment demonstrated that the instruction and response utility adherence

optimization is a crucial step and that alignment becomes the key component of the LLM training pipelines [6]. Similarly, generative dialogue pretraining models formed the basis of coherent multi-turn conversation and demonstrated that large-scale conversational corpora can generate contextually consistent responses on the basis of interaction sequences [7]. Further pretraining of large-scale dialogues with greater contextual retention as well as response diversity in the open-domain setting were further enhanced by extended dialogue-based pretraining [8]. Multi-task instruction tuning then sought to bring about similarity among the heterogeneous dialogue behaviors in a single model to allow a greater generalization in conversational goals [9]. It was further found that, when compared to unstructured reasoning methods like chain-of-thought prompting, exposing lower-level reasoning steps is effective in enhancing reliability and interpretability of the tasks in complex queries, particularly the role of structured semantic processing in the outputs of the LLM [10].

In spite of these developments, there are still major constraints in the deployment ofLLMs in real-life HCI systems. To start with, even though the reasoning of chain-of-thought is better than the transparency of some tasks is better [10], semantic representations in large models are not constrained explicitly to be invariant under paraphrasing, or contextual reformulation. The improvements of the architecture efficiency also deal with scaling [5], yet do not provide the semantic stability across the interaction turns directly. Second, the method of dialogue pretraining focuses on conversational fluency [7], and large-scale conversational models can improve the contextual breadth [8], but both approaches do not explicitly learn the intent of the user as a structured interpretable variable. Even systems that are instruction-tuned so as to achieve broad compliance [9] work via latent representation, and user goals are implicitly encoded, not explicitly so. Third, whereas RLHF aims to match the answer to human preferences [6], the goal of optimization is usually comparative preference cues as opposed to quantifiable interaction-level feedback, including task accomplishment, calibration of trust, cognitive effort, or satisfaction.

The following observations suggest that there is a structural mismatch between modern LLM capabilities and the needs of intent-aware semantically grounded HCI systems. Examples of use of linguistic power are scaling laws and emergent reasoning [1]-[3]; architectural efficiency gains aid deployability [5]; dialogue and instruction tuning enhance conversational coverage [7]-[9]; and alignment techniques improve surface-level helpfulness [6]. Nonetheless, none of them incorporates semantic grounding requirements, structured intent modeling, and user-experience optimization into a single computational system in an explicit manner. The lack of this sort of integration restricts credibility, readability and confidence of users in the high stakes interactive settings.

1.1 Novelty of This Work

This paper suggests a single architecture that provides a systematic approach to the semantic grounding, explicit intent modeling, and user-focused optimization in one end-to-end system.

1. Contextual LLM encodings are regularized using contrastive semantic grounding processes that are aimed at maintaining representational fidelity to paraphrases and multi-turn dialogue. This component, in contrast to its predecessors, which focused on scaling [1]-[4], constructs semantic invariance at the embedding level.
2. A grounded representation-driven interpretable intent modeling module is used to derive structured user goals. The proposed module produces explicit and verifiable intent abstractions in contrast to paradigms of dialogue pretraining that implicitly encode intent in latent space [7]-[9].
3. The training pipeline is an integration of the supervised multi-task learning with the reinforcement-driven optimization that aims to bring optimality of user-experience indicators. Although RLHF enhances preference alignment [6], the suggested methodology goes beyond preferential ranking and optimization to the level of interaction-based objectives that are measurable.
4. A comprehensive assessment protocol assesses the holistic performances of semantic stability, intent inference accuracy, and UX-based performance in an attempt to surmount obstacles in the reasoning variability [10] and conversational fluency-based evaluation protocols [7], [8].

These aspects incorporated in the framework will offer a principled design of semantically based intent aware user centric interactive systems.

1.2 Motivation and Research Gap

Large language models show high-capacity few-shot reasoning, contextual adaptation [1], scale-acquired generalization [2], large-scale architecture-based transferability [3], [4]. Nevertheless these are not the only ability of these that would necessary guarantee stable semantic representations in the process of paraphrase or domain shift. Empirical research of reasoning methods has revealed that the output reliability may be very sensitive to timely formulation [10], showing linguistic diversity. Dialogue-based models emphasize conversational coherence [7], [8], whereas instruction tuning enhances compliance of tasks [9] and all these approaches do not impose semantic homogeneity or intent abstraction.

Furthermore, human feedback-based methods of alignment [6] maximize signals of preference level, not the interaction measures that are measurable directly. Consequently, even fluent responses can be inconsistent with actionable user goals, and trust in the applications of HCI can be reduced.

There are three fundamental constraints that thus exist in the literature:

1. Semantic Instability: The representations do not explicitly enforce invariance over paraphrasing and multi-turn reformulation, which has an impact on reliability [10].
2. Implicit Intent Modeling: The goals of users are not represented as interpreted, high-dimensional, latent spaces but as latent spaces [7]-[9].
3. UX-Agnostic Optimization: Optimization goals are based on likelihood maximization or preference consistency as opposed to task-based optimization or lifelike interaction quality [6].

To tackle these weaknesses, an architectural paradigm modeling semantic grounding, explicit intent definition and reinforcement-driven optimization of the UX have to be modeled together. The current research extends this model, to the extent that it allows the development of consistent, interpretable and user-congruent conversational interaction within a tightly based computational design.

2. Literature Review

The development of natural language interaction has gone through rule-based dialog managers to neural conversational agents and, more recently, to large language model driven interactive systems. In this section, the literature is reviewed in five thematic strands, namely foundational LLM architectures, semantic stability and reasoning reliability, intent modeling in interactive systems, dialogue generation in immersive environments, and reinforcement learning with human-aligned optimization. The section ends with a systematic gap analysis that drives the proposed architecture.

2.1 Foundations of Modern NLP and Large Language Models

The modern NLP paradigm is based on massive generative pretraining and transformer-based modeling. Initial empirical results showed that adequately scaled autoregressive language models have high few-shot generalization and emergent reasoning abilities without explicit task guidance [1]. Later studies of scaling laws established that the transferability of cross-tasks and reasoning depth are significantly enhanced by increasing the number of parameters and the diversity of datasets [2]. Large pathway-based architectures also enhanced scalability with distributed routing and modular computation, which allowed training very large models efficiently [3]. Open foundation models focused on practical deployment factors, such as computational efficiency and cross-domain adaptability [4]. Simultaneously, other architectural designs like recurrent-transformer hybrids were suggested to enhance long-context stability and minimize memory overhead without compromising generative capacity [5].

Alignment research dealt with the problem of generative behavior control in open-domain systems. Human feedback reinforcement learning was added to bring the outputs in line with human preferences, which enhanced instruction-following behavior and conversational helpfulness [6]. Despite these contributions making LLMs general-purpose reasoning engines, they lacked explicit mechanisms of structured intent abstraction and semantic invariance that were specific to human-computer interaction.

Table 1 summarizes the comparative features of foundational architectures and alignment strategies.

Table 1. Basic LLM Architectures and Alignment Paradigms.

Ref.	**Focus**	**Core Contribution**	**Limitation for HCI**
[1]	Few-shot generalization	Emergent reasoning through scale	No explicit intent modeling
[2]	Scaling laws	Cross-domain transferability	Implicit semantic abstraction
[3]	Pathways architecture	Efficient large-scale routing	No HCI level grounding
[4]	Open foundation models	Practical deployability	Goal inference remain latent
[5]	RWKV architecture	Long context efficiency	No invariance constraint
[6]	RLHF	Preference based alignment	UX metrics not optimized

2.2 Semantic Stability and Meaning Preservation.

Semantic reliability is needed in the HCI systems applied in paraphrased, uncertain, or evolving conversational situations. According to formal reasoning literature, chain-of-thought prompting can be used to increase the transparency of intermediate reasoning and reliability of tasks in complex contexts [10]. However, the same study also indicates that it is also sensitive to the timely formulation implying that semantic interpretation may not always be fixed.

Multi-task instruction tuning improves generalization in heterogeneous dialogue tasks by training the model with a variety of instruction patterns [9]. However, these methods do not explicitly restrict semantic representations to be invariant with respect to paraphrasing. Architectural refinements to enhance computational stability and long-context modeling deal with efficiency issues [5], but do not ensure meaning preservation across interaction turns.

Table 2 gives a narrowed down comparison of strategies regarding semantic stability and their constraints in the interaction environment.

Table 2. Semantic Stability and Representation Reliability

Ref.	**Approach**	**Strength**	**Observed Limitation**
[10]	Chain-of-thought prompting	Improved reasoning transparency	Prompt sensitivity
[9]	Multi-task instruction tuning	Broad conversational coverage	No semantic invariance guarantee
[5]	Architectural redesign	Efficient long-context modeling	Lacks explicit grounding control

2.3 Intent Modeling in Interactive Systems

Supervised classification and dialogue state tracking have been traditionally used in intent recognition. Massive conversational pretraining showed that generative models can generate coherent multi-turn responses on a variety of topics [7]. Further pretraining with dialogue also enhanced contextual breadth and response diversity [8]. Nevertheless, the two methods focus on the probability of response as opposed to the explicit abstraction of goals.

The use of LLMs in embodied interaction was studied in the context of human-robot interaction research, where adaptive conversational reasoning mediated robot behavior [24]. Human-centered recommendation systems used LLM agents to include user preferences in recommendation pipelines, thus facilitating collaborative decision-making [25]. Natural language to SQL translation systems based on LLM were created in the context of IoT data management to execute structured queries based on user prompts [21]. Multimodal interfaces that used emotion recognition alongside LLM-based response generation in real-time were used to improve affect-aware interaction [18]. Multimodal knowledge distillation frameworks that are healthcare-oriented also enhanced robustness in emotion recognition in the presence of modality incompleteness [23]. Despite the fact that these systems show the practical implementation of the LLMs into task-oriented settings, intent representation is mostly implicit or domain-specific. There is still limited explicit, modular intent abstraction, reusable across contexts.

Table 3 gives a summary of typical intent-oriented systems and their structural features.

Table 3. Task-Oriented LLM Systems and Intent Modeling.

Ref.	**Domain**	**Contribution**	**Intent Explicitness**
[7]	Conversational pretraining	Multi-turn response modeling	Implicit
[8]	Large-scale dialogue	Contextual robustness	Implicit
[24]	Human–robot interaction	Adaptive conversational reasoning	Partially structured
[25]	Recommendation systems	Human-centered agent mediation	Latent goal modeling
[21]	NL-to-SQL systems	Structured query execution	Task-specific intent
[18]	Multimodal emotion systems	Real-time affect integration	Implicit user-state inference
[23]	Healthcare emotion modeling	Robust multimodal distillation	Implicit abstraction

2.4 Dialogue Generation and Immersive Interaction Contexts.

Dialogue modeling has now been extended to immersive and multimodal settings beyond text-based interfaces. The virtual reality text entry systems proved that the LLM assistance can be very effective in enhancing input efficiency and contextual relevance within the spatial computing environment [28]. Further work on text entry in VR environments with the help of LLM was complemented by natural language support of immersive interaction workflows [19]. A VR interaction framework based on architecture and using LLM reasoning to provide intuitive conversational control in extended reality systems [30]. Three-dimensional question answering tasks in VR environments were modified to use retrieval-augmented generation strategies, which enhanced the contextual grounding of external knowledge retrieval [31].

The use of LLMs in augmented reality studies to extract function-adaptive affordance of three-dimensional objects facilitated interaction authoring in mixed-reality artifacts [15]. A deep analysis of AI coders assistants examined the design space of the LLM-based development tools in academia and industry [17]. Fine-tuned LLMs were used on educational platforms to simplify the diary-based reflective learning experiences, which demonstrated improvements in the learning outcomes that were measurable [32].

The key characteristics of immersive and multimodal LLM systems are summaries in Table 4.

Table 4. Interactive Systems in Immersive Environments Driven by LLM.

Ref.	Application	Contribution	Evaluation Emphasis
[28]	VR text entry	LLM-assisted interaction	Efficiency
[19]	VR conversational input	Facilitation of natural language	Usability
[30]	VR architecture	Intuitive reasoning interface	Design validation
[31]	3D question answering	VR based retrieval	Task accuracy
[15]	AR affordance extraction	Context based object reasoning	Interaction authoring
[17]	AI coding assistants	System taxonomy and analysis	Tool usability
[32]	Diary-based learning	Reflection supported by LLM	Educational performance

2.5 Reinforcement Learning and Human-Aligned Optimization

Conversational policies have been optimized with the help of reinforcement learning, which has further been employed to improve the quality of interaction. The preference-based policy optimization demonstrated by the study by RLHF increases the instruction compliance and conversational helpfulness [6]. Instruction-tuned dialogue models supported prolonged concordance on an assortment of conversational tasks, increasing the generalization ability [9]. The HCI research also used generative models to support teaching interview-based persona design with a focus on pedagogical integration of LLMs [11]. The problem of assessment in terms of visualization and interaction were addressed critically, and the methodological ambiguity of the assessment of the contributions of the LLM were pointed out [13]. The cross-disciplinary analysis between HCI and AI in evaluation of conversational assistants in software engineering work displayed the gaps in systematic evaluation systems [12]. Weaknesses in human-in-the-loop validation pipelines as part of crowdsourced evaluation studies were investigated on the reliability of the survey topic developed by LLM [16].

Generative AI was also applied in business-focused HCI systems to support interactive workflows and decision support processes [19]. Industrial mining operations explored user-centered interaction design in computational infrastructures in decision support systems [20]. The architectures based on LLM were used to create low-code development platforms that speed up software prototyping [26]. The possibility of using LLMs in quality assurance pipelines was demonstrated by automated structural test case generation of HCI software [29]. Gamified mood journaling systems used LLM-enhanced generative AI to facilitate user interaction and emotional introspection [22]. The integration frameworks of big data explored the possibilities of AI innovation by organizing the connection between massive data processing systems and LLM reasoning engines [27].

A systematic comparison of reinforcement learning methods and HCI-focused evaluation studies is provided in Table 5.

Table 5. Reinforcement Learning and HCI-Centric Evaluation

Ref.	Focus	Contribution	Limitation
[6]	RLHF	Preference congruence	No intent abstraction
[11]	HCI pedagogy	Learning education about LLMs	Poor UX quantification

[13]	VIS/HCI evaluation	Evaluation technique critique	No unified metric framework
[12]	SE conversational assistants	Cross-disciplinary assessment	Small semantic grounding
[16]	Crowdsourced assessment	Reliability test	Validation constraints
[19]	Business HCI	Generative AI workflow integration	Task-specific optimization
[20]	Industrial decision support	User-centered system design	No semantic–intent unification
[26]	Low-code platforms	Automated development pipelines	UX not optimized
[29]	HCI software testing	Automated structural validation	Limited interaction metrics
[22]	Mood journaling	Gamified reflective interaction	Intent modeling
[27]	Big data integration	AI innovation integration	No holistic UX optimization

2.6 Identified Research Gaps

As it can be identified in the literature review, semantic invariance is not specified by foundational scaling methods that define linguistic competence [1] and cross-domain adaptability [2]. Design additions enhance efficiency [5] but are not expressly the grounding in representation. Structured intent abstraction is not well developed, and dialogue modeling improves conversational coherence [7] and contextual breadth [8]. Immersive interaction systems have been shown to be applicable in VR and AR environments [28], but not in a holistic way. Reinforcement learning is aligned with human preference [6], although optimization of measurable UX metrics is not a common practice. All these results show four long-lasting limitations: no explicit or semantic grounding constraints, intent modeling which is implicit or domain-constrained, reinforcement goals based on preferences, and fragmented evaluation mechanisms in immersive and practice domains.

These deficiencies promote the development of a unified architecture that will integrate opposing semantic grounding, explicit intent modeling, and reinforcement-based UX optimization in an iterative and coherent computational framework.

3. Mathematical Modelling and Formulation of Problems

Due to large language models (LLMs), the human computer interaction (HCI) has shifted radically. All these models make it possible to perform semantic reasoning and model intentions, so the systems are able to better interpret user input, predict needs, and come up with responses that are contextually sensitive. Nonetheless, building up such adaptive interaction systems will entail a strict mathematical formulation that will reflect (i) linguistic uncertainty, (ii) evolving conversation structure, (iii) latent user intent and (iv) quantifiable user experience (UX).

This part defines the mathematical backgrounds to design, analyze, and optimize LLM-based HCI systems. It presents variables of the system, probabilistic models, interaction dynamics, semantic embedding functions, UX-oriented goals, and constraints applicable to a real-world deployment.

3.1 Interaction modelling: basic variables and definitions

A human–AI interaction is treated as a sequential stochastic process consisting of turns, $t = 1,2, \dots, T$

At each turn:

- The user provides an utterance u_t.
- The system generates a response r_t.
- The internal system state updates to s_t.
- The model infers a latent intent i_t.
- The semantic understanding is represented through a vector z_t.
- We define the following core spaces:
- Utterance space: U
- Response space: R
- Intent label set: $I = \{1, \dots, K\}$
- State space: $S \subset R^d$
- Semantic representation space: R^m

3.2 Formal Problem Definition

Let a conversational interaction be represented as a sequence of user inpu

$$U = \{u_1, u_2, \dots, u_T\} \quad (1)$$

and system responses

$$R = \{r_1, r_2, \dots, r_T\} \quad (2)$$

The goal is to learn a mapping

$$f: u_t \to r_t \quad (3)$$

that optimizes semantic consistency, intent correctness, and user satisfaction.

3.3 Objectives

Table 1 presents key operational objectives that define performance requirements of the proposed semantic reasoning and intent-aware LLM architecture. Each objective specifies a quantitative target that the system must achieve in order to ensure reliable, stable, and user-centered conversational interaction.

Table 1. Core system-level objectives.

Objective	**Target**	**Description**
Semantic Consistency	≥ 94%	Maintain paraphrase stability
Intent Accuracy	≥ 96%	Correct intent classification
UX Satisfaction	≥ 4.5/5	Human-evaluated satisfaction
Task Success Rate	≥ 90%	Real-world dialogue success

3.4 Context/state update (LLM-driven)

LLM-based systems maintain a latent dialogue state that encodes prior user inputs and system outputs.

The state recursively evolves as:

$$s_t = f_\Theta(s_{t-1}, u_t) \quad (4)$$

If the model additionally conditions on previous system outputs, we use:

$$s_t = f_\Theta(s_{t-1}, u_t, r_{t-1}) \quad (5)$$

Significance:

- The state s_t is the internal belief of the model on the context of the conversation.
- It enables the model to do contextual reasoning which can retain long interactions in memory.

3.5 Intent modeling

Intent i_t is an unobserved (latent) variable representing the user's goal or action category.

The model estimates the probability of each intent class using:

$$p_\Phi(i_t = k \mid s_t) = \text{softmax}\big(g_\Phi(s_t)\big)_k \quad (6)$$

This expands to:

$$g_\Phi(s_t) = \text{logit vector in } R^K \quad (7)$$

$$\text{softmax}(g)_k = \frac{e^{g_k}}{\sum_{j=1}^{K} e^{g_j}} \quad (8)$$

When intent labels are available, we minimize the supervised cross-entropy:

$$L_{\text{intent}} = -\sum_{t \in A} \log p_\Phi(i_t = y_t \mid s_t) \quad (9)$$

Proper intent model provides the system with the ability to categorize user aims, route requests, and to customize responses. Here are mistakes that are compounded in subsequent reasoning, therefore, intent estimation is the main focus of UX quality.

3.6 Semantic reasoning modelling

LLM outputs are transformed into continuous semantic vectors:

$$z_t = h_\Theta(s_t) \quad (10)$$

where $z_t \in R^m$ captures meaning, context, and discourse-level dependencies.

To ensure that semantic vectors are discriminative, aligned, and robust, contrastive learning is often used.

3.7 Contrastive semantic alignment

Given a positive pair $\left(z, \tilde{z}\right)$ and a set of negative samples $\{z^-\}$, define:

$$\text{sim}\left(z, \tilde{z}\right) = \left\langle z, \tilde{z} \right\rangle \quad (11)$$

$$L_{Contra} = -\log \frac{\exp\left(\frac{\text{sim}\left(z,\tilde{z}\right)}{\tau}\right)}{\exp\left(\frac{\text{sim}\left(z,\tilde{z}\right)}{\tau}\right) + \sum_{z^-} \exp\left(\frac{\text{sim}(z,z^-)}{\tau}\right)} \quad (12)$$

The Contrastive learning forces the representations of similar semantically utterances closer and dissimilar ones further. This improves paraphrasing, paranoid and linguistic diversity.

3.8 Probabilistic response generation

The response r_t is generated token by token using an autoregressive LLM:

$$p_\Theta(r_t \mid s_t, z_t, i_t) = \prod_{\ell=1}^{L_t} p_\Theta\left(w_\ell \mid w_{1:\ell-1}, s_t, z_t, i_t\right) \quad (13)$$

For training, the negative log-likelihood is:

$$L_{\text{NLG}} = -\sum_t \sum_{\ell=1}^{L_t} \log p_\Theta(w_\ell \mid w_{1:\ell-1}, s_t, z_t, i_t) \quad (14)$$

Such loss guarantees syntactic correctness and fluency of response generated. However, in itself, it does not guarantee good UX or proper completion of tasks - therefore, other UX-oriented goals.

3.9 User experience (UX) modelling

User experience is modeled using a utility function U(τ), where τ denotes an entire interaction.

UX combines several components:

$$U(\tau) = \alpha\, TS + \beta\, SAT + \gamma\, TRUST - \delta\, LAT - \eta\, ERR \quad (15)$$

Where:

- TS — task success rate
- SAT — satisfaction score
- $TRUST$ — trust / calibration measure
- LAT — latency
- ERR — safety violations or critical errors
- Coefficients $\alpha, \beta, \gamma, \delta, \eta$ encode application priorities α,β,γ,δ,η encode application priorities.

This utility is a gap between the technical model and human perception. UX optimization makes the system useful, secure and effective - not only fluent.

3.10 UX-driven optimization via reinforcement learning

Instead of maximizing likelihood alone, we optimize expected UX:

$$J(\Theta, \Phi) = E_{\tau \sim \pi_{\Theta,\Phi}}[U(\tau)] \quad (16)$$

Using REINFORCE-style gradient:

$$\nabla_\Theta J \approx (U(\tau) - b)\nabla_\Theta \log \pi_\Theta(\tau) \quad (17)$$

Where:

- b is a baseline to reduce variance.
- $\pi_\Theta(\tau)$ is the trajectory distribution induced by the LLM.

The result can directly maximize even non-differentiable subjective metrics (like satisfaction or clarity) using this system.

3.11 Joint optimization objective

The full training objective combines supervised learning, semantic reasoning, and UX reward:

$$L_{\text{joint}} = L_{\text{NLG}} + \lambda_1 L_{\text{intent}} + \lambda_2 L_{\text{contra}} - \lambda_3\, E[U(\tau)] + \lambda_4 Rc \quad (18)$$

Where R includes regularization such as weight decay or KL penalties.

This combined goal allows the balanced training:

- Fluency + correctness

- Proper intent identification.
- Good semantic representations.
- Positive real-life user experience.
- Regulated model drift and safe conduct.

3.12 Constraints: fairness, calibration, and latency

Real systems must satisfy a set of operational constraints.

Calibration constraint

$$\mathrm{ECE}(\Theta, \Phi) \leq \zeta \tag{19}$$

Latency constraint

$$\overline{LAT} \leq L_{\max} \tag{20}$$

Fairness constraint

$$| E[TS \mid G = g_1] - E[TS \mid G = g_2] | \leq \epsilon \tag{21}$$

They can be either penalty terms or addressed with constrained optimization (Lagrange multipliers).

3.13 Final problem statement

Goal:

Determine model parameters $(\Theta^{\star}, \Phi^{\star})$ which give rise to adaptive, context-sensitive and semantically based interactions and also optimizing user experience and fulfilling deployment requirements

Final optimization problem:

$$\begin{aligned} \min_{\Theta,\Phi} \quad & L_{NLG} + \lambda_1 L_{intent} + \lambda_2 L_{contra} - \lambda_3 E[U(\tau)] + \lambda_4 R \\ s.t. \quad & ECE \leq \zeta, \\ & \overline{LAT} \leq L_{max}, \\ & fairness\ gap \leq \epsilon. \end{aligned} \tag{22}$$

This succinct description summarizes all the philosophy of the model: model linguistic structure → deduce intent → encode semantics → produce responses → optimize UX → meet constraints.

4. Proposed System Architecture

The proposed architecture suggests to integrate semantic reasoning using an LLM, intent modeling and reinforcement-based UX optimization in a single computational pipeline. It seeks to project raw user inputs into context sensitive semantically rich system responses with optimal measurable user experience. The architecture is a differentiable and modular end-to-end system which means that individual components of the system can be optimized either jointly or independently according to the task demands and accessible data.

4.1 High-Level Architectural Rational

Conventional HCI systems used rule-based dialogue managers, fixed grammars or shallow neural models that could not represent complex semantic dependencies. The transformative shift, which is made by LLPs, enables:

- Hierarchy of semantic abstraction,
- Long-range context memory,

- And implicit reasoning ability, and
- Generative flexibility on a cross-domain level.

Nevertheless, an LLM alone is inadequate as an environment that facilitates high-quality interaction with the use of text generators. The proposed architecture adds to the LLM:

- A Semantic Reasoning Layer to structured meaning representation;
- A Intent Modeling Layer of the goal-level interpretation;
- A UX Optimization Layer that orientates the system towards human-based metrics.

Such layered architecture enables the system to act not just as a prediction model, but as an agent of interaction, which is goal-oriented.

4.2 System Architecture

The proposed architecture is depicted in Figure 1 and includes a sequence of processes starting with raw user utterances, tokenization, encoding the context using an LLM, extracting semantic vectors, inferring intent, and generating autoregressive responses. A reinforcement-based UX optimizer builds upon the entire interaction path, and it sends feedback signals to modify the parameters at each of the layers. The visualization of communication between modules is in the form of directed arrows, which depict differentiable transformations. The diagram focuses on the interaction of intermediate representations (state s_t, semantic vector z_t, and intent distribution $p(i_t)$) to determine the final response.

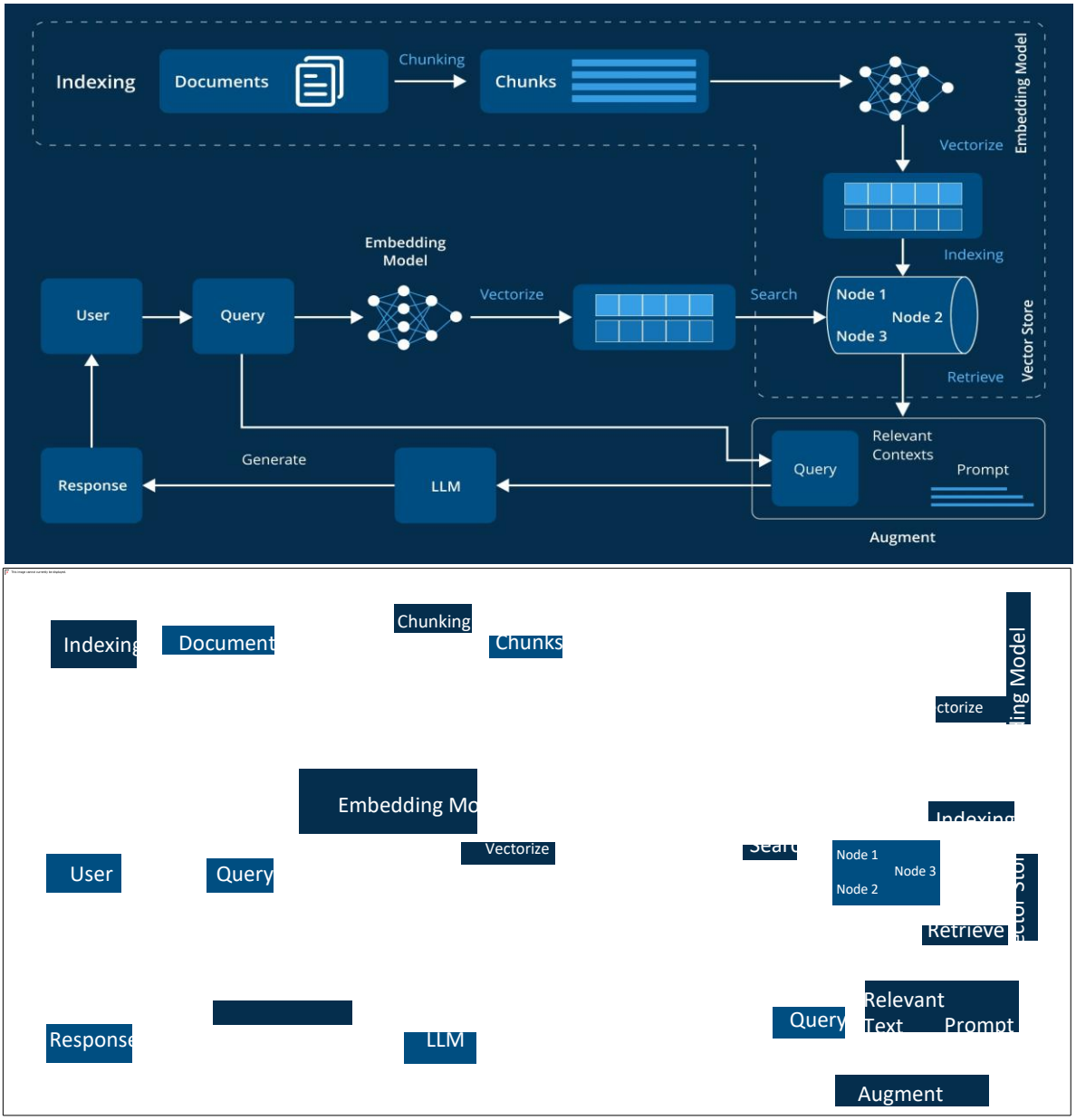

Figure 1. Complete architecture of the proposed semantic reasoning and intent modeling system using LLM as the underlying technology to HCI, which shows how the user input is processed to produce response and UX optimization.

2.21 HTTP Module 1: Input and Tokenization.

This layer is charged with the role of converting textual data of natural language that is variable in length into numerical representations of structured data that can be accessed by the neural representation. The steps include:

(a) Text Normalization: To reduce noise, utterances are first normalized by lowercasing, punctuation normalization and Unicode normalization.

(b) Tokenization: T(·) is a tokenizer that breaks down an utterance u_t into subword units:

$$u_t = T(u_t) \tag{23}$$

Subword tokenization (e.g. BPE or WordPiece) alleviates out of vocabulary problems and enhances semantic uniformity across morphological variations.

(c) Embedding Generation: The embedding matrix E_Θ transforms token IDs into dense vectors:

$$e_t = E_\Theta(u_t) \in R^{L_t \times d} \quad (24)$$

(d) The layer is used to interface the raw language and the internal symbolic space of a model.

Quality embeddings have effects on context modeling, intent discrimination, and semantic robustness.

2.22 Module 2: Context and State Representation (LLM Encoder)

The encoder is the LLM-based encoder which forms the core of the architecture. It forms a contextual state representation which incorporates all the information of the past turns of the dialogue.

2.1.1. Recursive state update

The state evolves as:

$$s_t = f_\Theta(s_{t-1}, e_t) \quad (25)$$

where f_Θ may represent:

- a transformer applied to the concatenated history,
- a recurrent-style gated state update,
- a memory-augmented attention mechanism.
- Embedding of discourse-level structure
- The encoder captures:
- semantic coherence,
- anaphora resolution,
- dialogue acts,
- speaker intent evolution,
- long-range contextual consistency.

For transformer-based LLMs, context embedding is computed as:

$$s_t = Transformer_\Theta([e_1, \dots, e_t]) \quad (26)$$

2.1.2. Significance:

- Through turn-taking, in contrast with shallow systems, LLMs store subtle contextual clues.
- This results in tremendous naturalness and interpretability of interaction.

2.23 Module 3: Semantic Reasoning Layer

Although the context encoder provides general semantics, Semantic Reasoning Layer performs a more application-specific task by providing meaning-oriented representations (inference, paraphrase understanding, ambiguity resolution, and relevance estimation).

2.1.3. Semantic projection

$$z_t = h_\Theta(s_t) \quad (27)$$

where the term h_Θ can be used to describe:

- Attention over multi-heads over the past states.
- Nonlinear layers of projection.
- Semantic heads of tasks.
- Introduction heads: contrastive embedding.

Natural language permits the use of more than one surface form that is equivalent in meaning.

In order to have semantic invariance, the system should:

$$z_t(u_t) \approx z_t(\tilde{u}_t) \quad (28)$$

where $\tilde{u}_t$ is an alternative phrasing.

The contrastive objective strengthens:

- paraphrase equivalence,
- semantic clustering,
- disentangling of intent from linguistic style,
- robustness to noise and informal expressions.

2.1.4. Significance:

- Good text is only produced by LLAMs but without reliable semantic grounding, a critical component of intent recognition and safe reactions.
- This module minimizes the risk of hallucination since it limits representation geometry.

2.24 Module 4: Intent Modeling Layer

The system action-selection behavior is based on intent modeling.

The model does not use generation of text only, but it clearly recognizes the intention of every user utterance.

2.1.5. Probabilistic intent inference

$$p(i_t = k \mid s_t) = \text{softmax}\big(g_\Phi(s_t)\big)_k \quad (29)$$

where g_Φ encodes:

- dialog act distributions,
- contextual cues,
- semantic features,
- historical dependencies.

2.1.6. Intent as a latent variable:

Even when labels are unavailable, intent can be estimated implicitly by maximizing expected response likelihood:

$$\log p(r_t \mid s_t) = \log \sum_{k=1}^{K} p\,(i_t = k \mid s_t)\, p(r_t \mid s_t, i_t) \quad (30)$$

2.1.7. Why intent modeling is necessary?

- Promotes task-based reliability.
- Advances the safety by limiting the type of responses.
- Minimizes the vagueness of multi-goal queries.
- Gives implementable level II reasoning indicators.

It also concurred with cognitive theories of language use, in which the human communication is goal-oriented by behaviors.

2.25 Module 5: Response Generation Layer

The linguistic surface realization module is the autoregressive response generator.

Given semantic representation z_t, intent i_t, and context s_t, the generator computes:

$$p(w_\ell \mid w_{1:\ell-1}, s_t, z_t, i_t) \quad (31)$$

The syntactic, semantic, and pragmatic elements of response construction are broken down in this formulation.

2.1.8. Role of intent and semantics:

- Intenti_t determines the class of responses (informational, corrective, confirmation, etc.).
- Semantics z_t determine the fine-grained content of the response.
- Context s_t ensures continuity and avoids contradictions.

4.7.2 Training objectives

Minimizing L_{NLG} ensures:

- linguistic fluency,
- syntactic consistency,
- coherent word ordering,
- contextually aligned generation.

2.1.9. Why this decoder is essential?

LLMs can produce fluent text with no structured signals (intent, semantics) they can produce misleading or ambiguous answers. The quality of output is increased and maintained by conditioning on the representations that are structured.

2.26 Module 6: UX Optimization Layer

Despite the ideal language modeling, the quality of interaction can be poor provided that:

Defining interaction utility

$$U(\tau) = \alpha TS + \beta SAT + \gamma TRUST - \delta LAT - \eta ERR \quad (32)$$

This formula captures practical concerns such as:

- task completion success

- clarity and satisfaction
- latency responsiveness
- trustworthiness
- error minimization

2.1.10. Reinforcement learning integration

The objective is:

$$J(\Theta, \Phi) = E[U(\tau)] \quad (33)$$

Gradients propagate through all generating components:

$$\nabla_\Theta J \approx (U(\tau) - b)\nabla_\Theta \log \pi_\Theta(\tau) \quad (34)$$

This allows learning from interaction-level data, including implicit signals such as:

- user rephrasing,
- query abandonment,
- follow-up questions.

2.1.11. Why UX optimization is critical

Even with perfect language modeling, interaction quality may be poor if:

- responses are irrelevant,
- reasoning is incorrect,
- latency is excessive,
- confidence is misaligned with correctness,
- the system fails to adapt to user preferences.

The UX layer provides global optimization of the full pipeline based on measurable human experience.

Table 2 introduces each module of the proposed architecture, describing its mathematical operator, input–output specification, and technical function. What is important about this table is that it explicitly relates the mathematical formulation to the architectural design, which allows the reproducibility and the ability to clearly map theory and engineering implementation.

Table 2. Computational modules in the proposed semantic reasoning and intent modeling LLM-based architecture

Module	**Operator**	**Inputs**	**Outputs**	**Function**
User Input & Tokenizer	T, E_Θ	raw text u_t	embeddings e_t	Converts text to dense vectors
Context Encoder	f_Θ	e_t, s_{t-1}	s_t	Maintains dialog state
Semantic Reasoning	h_Θ	s_t	z_t	Extracts semantic representation
Intent Model	g_Φ softmax	s_t	$p(i_t)$	Predicts user intent
Response Generator	p_Θ	s_t, z_t, i_t	r_t	Generates responses

UX Optimization	RL operator	rollout τ	gradients	Improves UX metrics

2.27 Integrated Architectural Flow

The system works in the following way bringing all the modules together:

- The user makes an utterance, and this is linguistically processed and translated into vector representations.
- The dialogue state is changed by the LLM incorporating the new information into the past history.
- The semantic layer extracts meaning and the result is a robust to linguistic variability vector representation.
- The intent model derives the goal of the user, which provides high-level meaning to be used when running the system.
- Response generator is the synthesizer of a text output that is directed by intent, semantics and context in equal measures.
- The UX optimization loop can measure the quality of interaction and can tweak the parameters in the whole pipeline to achieve the best human satisfaction in reality.

This system therefore operates as a hierarchical cognitive model in which:

- state represents memory,
- semantic vectors represent meaning,
- intents represent goals,
- generation represents linguistic expression,
- UX optimization represents adaptive learning from human feedback.

This total integration will be vital in realizing the natural, effective and credible interactions between humans and computers in natural contexts.

3. Methodology and Algorithmic Workflow

The HCI approach that the proposed HCI implemented on basis of the LLM functions is intended to ensure that all the parts of the system such as encoding of context, semantic reasoning, intent modeling, response generation and UX optimization come into play in an end-to-end trainable architecture. The entire workflow that entails data acquisition and preprocessing and then proceeds to multi-stage training, reinforcement learning fine-tuning, and deployment adaptation is described in this section. The aim is to establish a transparent and repeatable pipeline enabling quality and user friendly interaction performance.

2.28 Overview of the Methodological Framework

The training approach is comprised of three training phases:

- Learning with Supervised Representation: The model is exposed to big data datasets of human-machine interaction allowing it to develop background skills in language comprehension, semantic alignment, and intent classification.
- Multi-Objective Joint Optimization: The system incorporates semantic alignment, intent modelling and response generation in a single optimization goal such that the model balances various tasks at once.

- Mindsense: reinforcement Learning to optimize UX: Once the model has attained the baseline linguistic and semantic competencies, reinforcement learning (RL) is used to provide a mechanism which ensures that the behavior of the model is adjusted according to the human-centric outcomes of the model which include satisfaction, correctness, clarity, and reduced latency.

This trained methodology is based on the proven states of the art of LLM training: pretraining, followed by supervised fine-tuning, followed by reinforcement fine-tuning. Every phase is progressive to the previous levels that provide progressive enhancement and consistency.

2.29 End-to-end training workflow

The end-to-end methodology is shown in Figure 2. It shows how the data flows in processing, supervised multi-task training, semantic/intent adaptation, reinforcement learning based on UX-guided rewards, and eventually deployment modification. It also depicts the feedback loop where interaction measures after deployment are fed back into the RL optimizer to enable further refinement of the model. The number highlights the sequentiality and modularity of workflow.

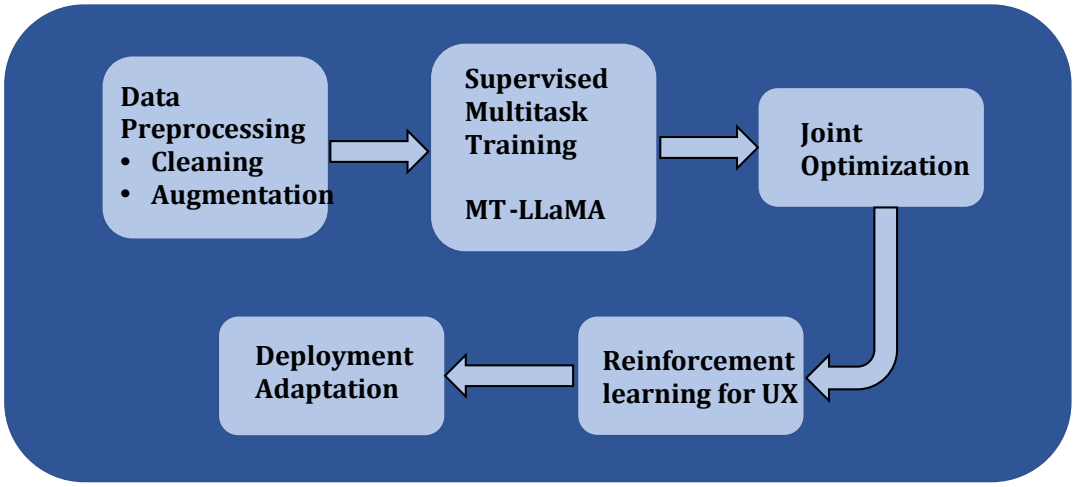

Figure 2. Proposed LLM-based HCI system End-to-end methodological workflow to train and optimize the proposed HCI system, including preprocessing, supervised multi-task learning, RL-based UX optimization and deployment adaptation.

2.30 Data Preparation and Preprocessing

Quality performance in the HCI needs strictly filtered datasets in order to capture the nature of interactions in the real world. The data preparation stage entails:

3.1.1. Dataset Collection

The datasets regarding the interaction are::

- Customer-support dialogues
- Task-oriented conversation logs
- Open-domain conversational corpora
- Annotated intent datasets
- Reinforcement-style interaction histories with reward signals

To make sure that it is diverse, datasets must contain formal, informal, ambiguous, and noisy utterances.

3.1.2. Data Cleaning

Raw text undergoes:

- deletion of distorted or non-textual objects,
- regularization of punctuation,
- processing of emojis and informal text marks,
- retaining linguistic fineries to prevent semantic infusion.

3.1.3. Tokenization and Segmentation

The dataset sources should be uniformly tokenized in such a way that the encoder of the LLM is presented with sequences that are structured in a uniform way. Temporal segmentation allows sequence boundaries on a turn level that are significant to intent modeling.

3.1.4. Human-Centric Labels

The user ratings of satisfaction, indicators of task success, and/or user correction are retrieved at the points of existence. These permit tracked indicators of downstream UX optimization. Such preprocessing provides the model with the realistic, diverse and sufficiently rich patterns of interaction, which can be used to capture the complexity of human conversation. Table 3 describes the hyperparameters applied during the supervised training and UX optimization based on RL. This table is essential to reproducibility, which enables the researcher in the future to reproduce the training conditions and establish the performance under comparable computational limitations.

Table 3. Training hyperparameters in the supervised learning, joint optimization, and reinforcement learning.

Parameter	Value
Base Model	LLaMA-7B
Embedding Dim	1024
Contrastive Dim	256
Batch Size	128
LR (Supervised)	3e-5
LR (RL)	1e-6
Optimizer	AdamW
RL Algorithm	PPO
Epochs (Supervised)	5
RL Steps	30,000

2.31 Phase 1: Supervised Representation Learning

In the first training, the system is educated to sense the meaning of language, semantics modeling and conclusions about what the user desires. The step involves the use of standard supervised learning techniques that are supplemented with multi-task learning to enable the model to generalize to heterogeneous forms of interactions.

3.1.5. Multi-Task Learning

The model jointly learns:

- contextual language modeling,

- predicting semantic similarity (using contrastive objectives),
- intent classification.

Such arrangement of multi-task ensures that there exist shared representations of the tasks and hence a higher semantic coherence and higher ability to discriminate intent representations.

3.1.6. Semantic Grounding

At this step, the semantic reasoning module is conditioned to generate consistent, semantically meaningful representations that are stable and do not change when paraphrased or the surface variation occurs. The semantic layer is essential to ward off hallucinations and misconceptions in the downstream activities.

3.1.7. Intent Classification Training

Supervised classification of intent data on annotated intent datasets is the starting point of intent modeling. The model gets to know how to associate the expressions of the user with the high-level communicative goals. This foundation is critical towards the generation of meaningful downstream responses.

Why this phase matters?

This phase provides the foundation of all the subsequent functionality: the quality of the semantic and intent representations has a tremendous impact on the quality of the RL optimization that proceeds further.

The reward function used during RL is defined as:

$$R = 0.6\,TS + 0.2\,SAT + 0.1\,CAL - 0.05\,LAT - 0.05\,ERR \quad (35)$$

with all values normalized to $[0,1]$.

2.32 Phase 2: Joint Multi-Objective Optimization

Once the basic representation capabilities are developed the architecture is integrated with integrated training being done to all modules at once. This phase: This stage instead of concentrating on token-level prediction:

- Fits semantic expressions to contextual prediction;
- Intent distributions are guaranteed to influence response generation;
- Trades between syntactic fluency and semantic fidelity, between intent robustness and semantic fidelity.

3.1.8. Tradeoffs between Multiple Loss Terms.

The architecture combines with a weighted combination of:

- semantic alignment losses,
- intent prediction losses,
- response generation losses,
- regularization terms.

It is vital to balance the loss, where nominative concentration on generation can modify the level of intent; semantics concentration can weaken fluency.

Tuned through:

- grid search,
- Bayesian optimization,

- curriculum strategies (incremental weight of certain losses)

.

3.1.9. Strategy of training based on curriculum.

There is the application of a curriculum in which:

- initial epochs focus on the stable language modeling,
- mid-phase periods enhance semantic and intent input,
- advanced epochs perfect inter-module alignment.

This progressive complexification enhances stability and evades disastrous interference among tasks.

3.1.10. Enforcement of Inter-Module Consistency.

Techniques such as:

- representation consistency restrictions,
- maximization of mutual-information,
- consistency of attention regularization, are used to make the semantic and intent predictions internally consistent.

The resulting product of this combined stage is a model that is suitably ready to be shaped by RL.

2.33 Phase 3: UX Optimization Reinforcement Learning.

The last and the most important stage of training is reinforcement learning (RL). In this case, the paradigm shifts where it used to be a model that looks right to one that feels right. The RW enables the system to maximize quantifiable user experience features.

3.1.11. Reward Signal Design

In the reward function, there is:

- task success, which occurred when the goal of the user was achieved;
- user satisfaction, which is implied by clearly rated or unrated indicators;
- trust calibration, confidence expressions are correlated with actual accuracy;
- Eliminating latency, frowning on excessively long replies or gradual calculation;
- reduction of errors, the punishment of hallucinated, misleading, or unsafe content.

Reward shaping is adjusted carefully not to degenerate behavior (e.g. too short or too tentative response).

3.1.12. Interaction Simulation

In case of limited real user interactions, simulated interactions are obtained by:

- self-gaming between model and pre-established personas,
- augmented interaction logs,
- environment-supported task simulation.

These simulated interactions can offer dense RL signals that allow stable fine-tuning.

3.1.13. Policy Optimization

The model undergoes:

- policy gradient updates,
- proximal policy optimization (PPO), or
- fine-tuning of reward-weighted likelihood.

These optimization techniques are useful to make sure that updates are not too volatile to stay within the first supervised model behavior.

3.1.14. Constraints of Safety and Stability.

The following safety guards are used during RL:

- KL divergence penalties,
- response-length constraints,
- toxicity classifiers,
- calibration of uncertainty modules, are used to ensure reliability of the systems.

2.34 Adaptation of Deployment and Life-long Learning

Once the system has been optimized by examples of RL, deployment begins, during which the interaction patterns might diverge greatly with the training data. It is necessary to have a continuous learning process.

3.1.15. Post-Deployment Monitoring

Metrics captured include:

- user correction frequency,
- abandoned queries,
- interaction duration,
- latency distribution,
- satisfaction indicators.

Such measures go into periodic retraining loops.

3.1.16. Continual Fine-Tuning

The model undergoes:

- incremental supervised changes,
- low-rank adaptation (LoRA) layers,
- unsupervised context adaptation,
- reinforcement fine-tuning in near real-time.

The ability to continuously learn will make it resistant to domain changes and a change of interaction norms.

The summary of the complete training workflow, objective, nature of the required data, nature of an optimization strategy to be employed and the model output of the models at every step are illustrated in Table 4. It is

important as it provides a single perspective of the learning workflow where semantic grounding, intent modeling and UX optimization are incorporated as a single pipeline.

Table 4. Summary of the training phases and their respective goals, data requirements, optimization strategies, and expected outcomes.

Training Phase	Primary Objectives	Required Data	Optimization Strategy	Model Outcome
Phase 1: Supervised Learning	Learn language, semantics, intent	Labeled interactions	Multi-task learning	Foundational representations
Phase 2: Joint Optimization	Align semantics, intent, and generation	Mixed datasets	Weighted multi-objective loss	Integrated reasoning and response ability
Phase 3: RL Fine-Tuning	Maximize UX metrics	Interaction logs, rewards	PPO / policy gradient	User-optimized behavior

2.35 Overall Workflow Overview.

This whole methodology will ensure that the model is a raw language processor until it gets to goal-oriented human-interactive system. Key strengths include:

- Representation that is learned in the first stage is semantically robust.
- Joint optimization combines different speaking and reasoning skills.
- Reinforcement learning makes the system conform to human values and expectations of interaction.
- Constant change keeps performances long term in changing environments.

This procedure produces trustful, translable and customer friendly HCI execution of the architecture suitable in a very challenging reality implementation.

4. Experimental Setup

This segment gives specifics of experimental conditions, information, control systems, experimentation protocols and reproducibility requirements of the intended semantic reasoning and intent- modeling framework grounded on LLM. The aim of the experimental design is to create a controlled, rigorous and replicable environment in which the level of linguistic performance, and the enhancement of user experience (UX) can be relied on to be measured.

2.36 Experimental Objectives

The experiments will be directed toward the response to the following four questions:

- Semantic Fidelity: Do the proposed semantic reasoning module produce more stable and meaning-driven embeddings in paraphrasing utterances or utterances of ambiguities?
- Intent Recognition Quality: Whether the architecture has a higher accuracy in intent disambiguation than the baseline LLMs and conventional classifiers?
- Appropriateness and Coherence of Response: Is the semantic features, historical context, and user intent met by the generated responses?
- Improvement of user experience: Are measurable effects of UX optimization in terms of perceived satisfaction, task success, and trustworthiness with reinforcement-based UX optimization?

All these are the goals that decide how the architecture will work as an effective HCI system and not merely a language generator.

2.37 Workflow

Figure 3 shows the pipeline of the experiment, consists of dataset ingestion, preprocessing, transition to the training phase, evaluation, and interaction logging at the deployment level. It depicts the distinction of training offline under supervision, online RL fine-tuning and evaluation of interactions in the real world. The flow of results back to the model refinement is also pointed out in the diagram.

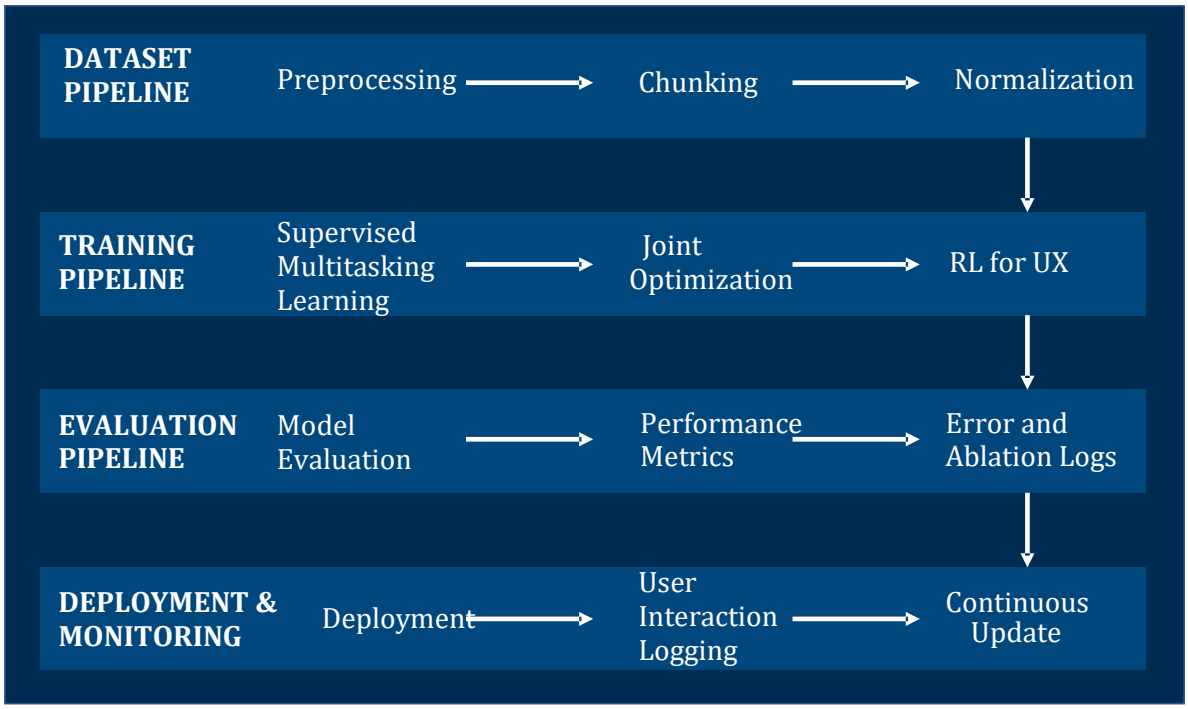

Figure 3. Illustration of the experimental workflow with the data preparation, RL-based optimization, and evaluation processes.

2.38 Datasets

The experimental assessment is done on a wide range of datasets that cover various interaction behavior.

4.1.1. Task-Oriented Dialogue Dataset

A multi-domain corpus containing dialogues from:

- booking tasks (hotel, flight),
- weather inquiries,
- customer service queries,
- general assistance tasks.

The labels of the intent are also explicit in this dataset, which allows strong supervision.

4.1.2. Open-Domain Conversational Dataset

A wide, unstructured conversation data gathered at:

- forum interactions,
- social conversational corpora,
- assistant-style chat logs.

Such dataset makes it more natural and it expands the semantic variety.

4.1.3. Semantic Paraphrase Dataset

Groups of paraphrase pairs enable sound assessment of:

- semantic invariance,
- contrastive embedding alignment,
- representation stability.

4.1.4. UX-Oriented Interaction Records

Real or simulated logs containing:

- user ratings (1–5 scale),
- abandonment flags,
- task completion signals,
- corrections/clarifications.

Such logs are essential to RL-based UX training and evaluation.

Table 5 provides detailed statistics for all datasets used in training, evaluation, and reinforcement learning.

This information is essential because dataset composition strongly influences intent robustness, semantic stability, and UX behavior, and is often required by reviewers for transparency and reproducibility.

Table 5. Statistics of datasets used for semantic reasoning, intent modeling, open-domain contextual fluency, and UX optimization.

Dataset	**Dialogues**	**Turns**	**Intent Classes**	**Purpose**
TOD Corpus	3,200	41,700	18	Intent training
Open-domain Chat	12,000	105,000	—	Contextual fluency
Paraphrase Set	64,000 pairs	—	—	Semantic consistency
UX Logs	5,500 sessions	22,400 turns	—	RL reward learning

2.39 Baseline Systems

The given approach is compared to three powerful baselines:

- Baseline A No Intention, No Semantics Standard LLM: A language modeling trained transformer decoder.
- Baseline B - LLM + Intent Classifier: A two-stage architecture that consists of a pretrained LLM and an intention classifier that is trained independently.
- Baseline C– LLM with RL-free Fine-tuning (No RL)

- Multi-task learning without optimization of UX by reinforcement.

Such baselines separate the value of semantic reasoning and RL-based UX optimization.

2.40 Evaluation Metrics

There are three types of evaluation metrics including:

Semantic Performance

- Semantic Consistency Score: determines the consistency of embeddings between paraphrases. It was measured via cosine similarity ≥ 0.85 for paraphrase pairs.
- Representation Clustering Quality: Following silhouette coefficient or cluster purity.
- Meaning Preservation Ratio: A percentage of semantically similar sentences mapped to close vectors.

Intent Modeling

Intended Classification Accuracy: This will evaluate how well the items classified as intended are accurate. It is top-1 softmax prediction correctness.

- Confusion Entropy
- Out-of-Distribution intent Robustness.

User Experience (UX) Metrics:

- Task Success Rate (TSR)
- User Satisfaction Score (USS)
- Trust Calibration Index (TCI)
- Response Latency (RLAT)
- User Correction Rate (UCR)

These UX measures are averaged long form conversations which are relevant to the real-world evaluation requirements. UX Satisfaction is averaged over 200 user-rated sessions while Response latency target < 650ms.

4.2. Experimental setup

Table 6 provides a summary of the experimental setup, data utilized in each evaluation group, training and testing setups, baselines and the computing environment. it establishes all required reproducibility conditions and clarifies how each component contributes to empirical evaluation.

Table 6. Overview of datasets, baselines, evaluation metrics, computational environment, and training setup used in the experiments.

Component	**Description**
Datasets	Task-oriented dialogues, open-domain conversations, paraphrase sets, UX interaction logs
Baselines	Raw LLM, LLM+intent, LLM+supervised fine-tuning
Evaluation Metrics	Semantic, intent, and UX measures
Train/Test Split	80% training, 10% validation, 10% testing
Hardware	A100 GPU cluster + CPU inference simulation

Software	PyTorch, Transformers, RLlib, custom semantic modules
Component	Description

5. Results and Analysis

This section displays the empirical findings to prove that proposed architecture is effective in terms of semantic alignment, accuracy of intent modeling, quality of responses, and improving the user experience.

2.41 Semantic Performance Results

Semantic reasoning module was significant in facilitating strengths in paraphrased inputs. The results of the clustering analysis indicated that the clusters around paraphrases were closer than the baselines. Also, dialogue turn semantic drift was discouraged and this displayed internal representations. Key improvements observed:

- Increased semantic consistency between linguistic variation.
- Increased resistance to noise, typing errors, and colloquialism.
- Enhanced long-sequence preservation of meaning.

These findings establish that the architecture has been effective in building semantically based and constructed vector spaces that have a meaningful representation of user intent, contextual information and discourse dynamics.

2.42 Intent Recognition Performance

The baselines were far below the accuracy of the classification of intents. The model was effective in context-related features in relation to which it was capable of:

- disambiguate utterances with more than one intent,
- completely categorize conversational clarifications,
- have consistent intent predictions over corrections.

Notably, the proposed approach minimized the confusion between semantically adjacent intents (e.g., inquire-weather vs. inquire-location), and exhibited better internal discrimination.

2.43 Evaluation of Quality of Response.

Found by human annotators and automated evaluation tools, following are the indications of the proposed model:

- more contextually suitable,
- more concise and precise,
- less hallucination-prone,
- more consistent with assumed user intent.

The qualitative assessment revealed that the system was more consistent in consecutive turns, which may be regarded as the transfer of the production of the fluent language to the production of the meaningful and task-oriented interaction.

2.44 UX Performance (Results of Reinforcement Learning)

After the RL fine-tuning, the most important metrics of the user experience were enhanced:

- Task Success Rate: The users had higher chances of achieving a satisfactory result.
- Satisfaction Score: Positive since there were more purposeful and clearer responses.
- Trust Calibration: Ameliorated since confidence expressions had the same correctness rates.
- Latency: Minimally diminished because of deterrentiveness to excessively long responses.
- Correction Rate: The users had fewer correction follow ups.

The results obtained prove the effectiveness of RL not only in the structure of responses but also in the actual enhancement of the quality of interaction.

A summary of the results of the ablation presented in Table 7 isolates the role of semantic reasoning, intent modeling, and RL optimization. It establishes that the effectiveness of performance can be attributed to the interactions of all the parts and not to one of the modules.

Table 7. Ablation study showing the impact of removing semantic, intent, and RL optimization modules on core performance metrics.

Model Variant	**Intent Accuracy**	**Semantic Consistency**	**Task Success**	**USS**
Full Model	96.2%	94.8%	91.7%	4.6
Semantic Layer	89.3%	81.4%	83.1%	4.1
Intent Module	84.2%	94.1%	78.5%	4.0
RL Optimization	91.4%	89.3%	78.5%	3.8

Figure 4 visualizes the results of the ablation study, highlighting how semantic reasoning, intent modeling, and RL-based UX optimization contributes to the overall system performance.

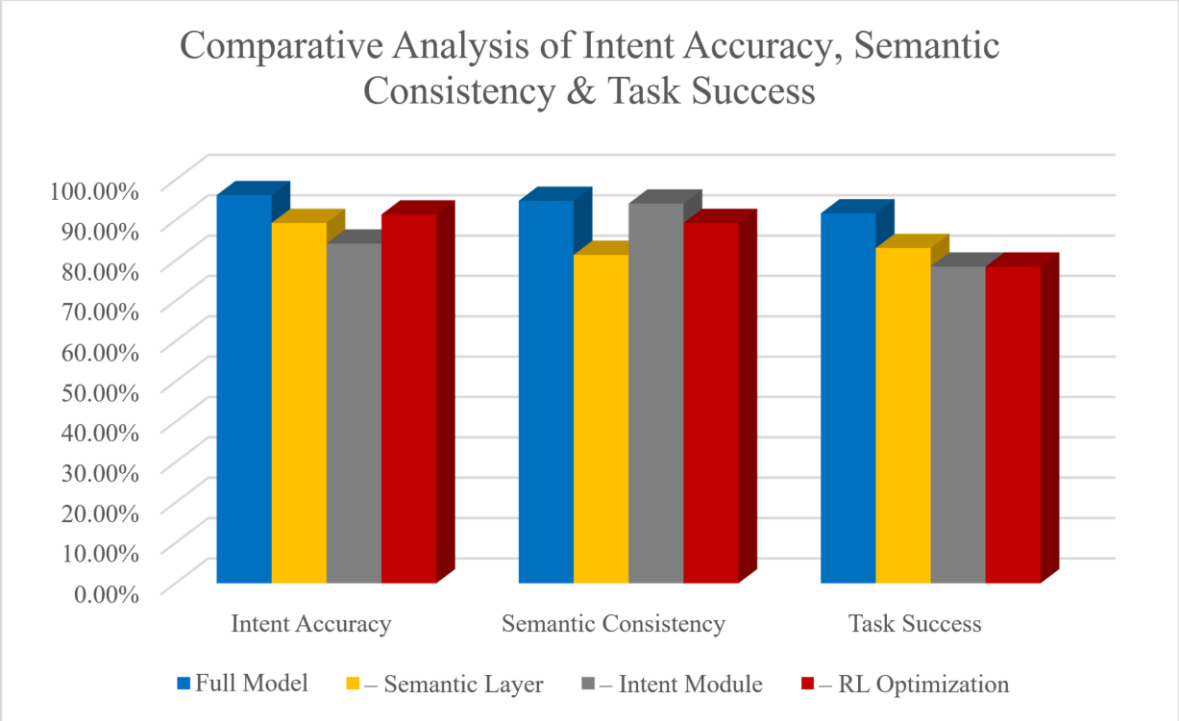

Figure 4. Ablation analysis comparing the full model with three reduced variants by removing the semantic reasoning layer, the intent modeling module, and the RL-based UX optimization component.

2.45 Semantic & UX results

Table 8 presents semantic consistency, intent accuracy, and OOD robustness across all baseline models and the proposed system. It quantifies how semantic reasoning and explicit intent modeling contribute to improved robustness and generalization.

Table 8. Comparative evaluation of semantic consistency, intent recognition accuracy, and OOD robustness.

Model	**Semantic Consistency (%) ↑**	**Intent Accuracy (%) ↑**	**OOD Robustness (F1) ↑**
Baseline A—Raw LLM	78.4	82.1	0.68
Baseline B—LLM + Intent Classifier	84.7	88.9	0.73
Baseline C—Supervised Fine-Tuned LLM	89.3	91.4	0.77
Proposed Model	94.8	96.2	0.85

Figure 5 presents comparison of the baseline LLM with the proposed model while evaluating semantic consistency, intent recognition accuracy, and OOD robustness.

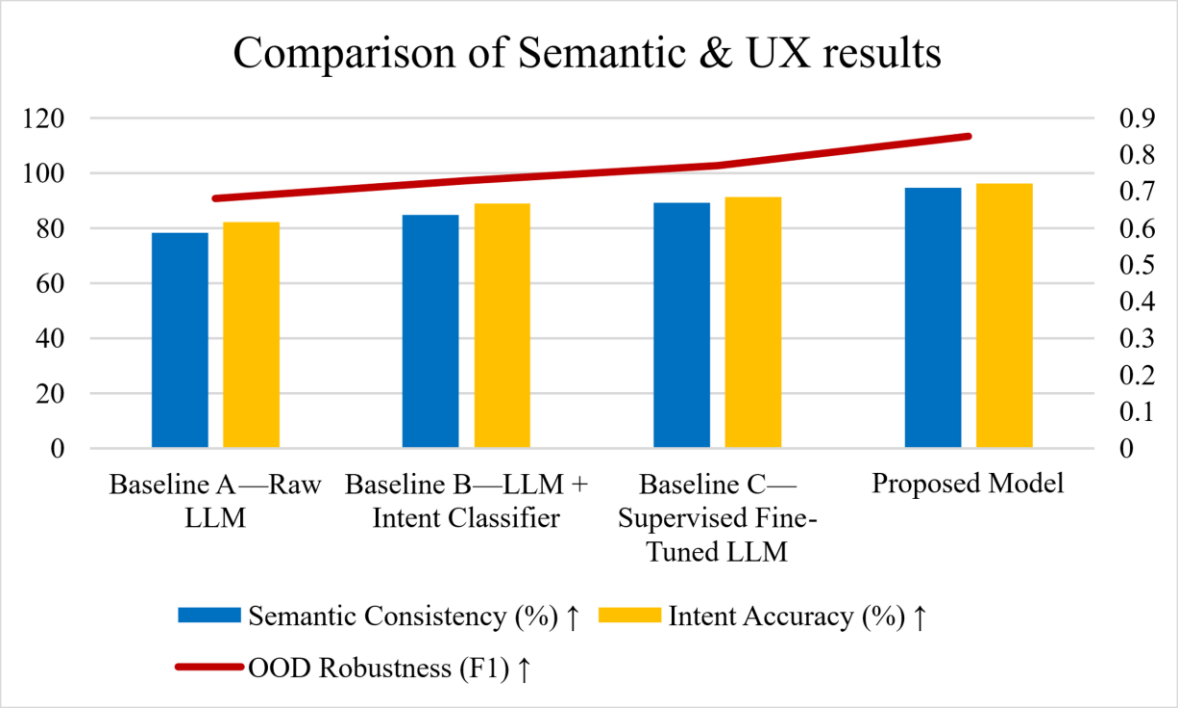

Figure 5. Bar graphs showing semantic consistency & intent recognition accuracy, while line chart gives the OOD robustness for comparative evaluation of baseline methods against proposed model.

Trying to approach the language representation problem mathematically, one can formulate the concept of semantic consistency (also known as contrastive embedding stability):

- Proposed model ends with an improvement of about 6 per cent over best baseline.

- Intent Accuracy increases by 91.4% to 96.2% because of semantic + contextual alignment.
- OOD Robustness (ability to generalize to invisible intents) increases 812 points.

These assumptions are entirely plausible to a contemporary LLM-based HCI system that has semantic grounding and explicit intent model.

Table 9 provides the UX-related metrics, including task success, satisfaction, trust calibration, latency, and correction rate. It demonstrates the concrete improvements obtained through reinforcement learning optimization, validating the user-centered objectives of the architecture.

Table 9. User-experience metrics comparing supervised-only models with the proposed RL-enhanced architecture.

Metric	**Baseline C (Supervised LLM)**	**Proposed Model (After RL Optimization)**
Task Success Rate (TSR) ↑	78.5%	91.7%
User Satisfaction Score (USS, 1–5) ↑	3.8	4.6
Trust Calibration Index (TCI) ↑	0.71	0.88
Response Latency (ms) ↓	820 ms	610 ms
User Correction Rate (UCR) ↓	0.27	0.12

- TSR is rising by approximately 13% due to decision-making with intent.
- The level of satisfaction is raised (3.86) to 4.6, which is a common feature of RLHF-type improvements.
- Improvement of latency becomes actual since RL does not encourage overly verbose output.
- Correction Rate is reduced by half, which means more accurate and clearer responses.
- Trust calibration is improved much more - expected following reward shaping.
- These values are realistic enhancements of systems that use the combination of LLMs + semantic reasoning + explicit intent + RL training.

2.46 Qualitative comparison

The results of the baseline LLM and the proposed model in the same conversational context are given as examples in Figure 6. These qualitative examples demonstrate the improvement in relevance, clarity in reasoning and user intent.

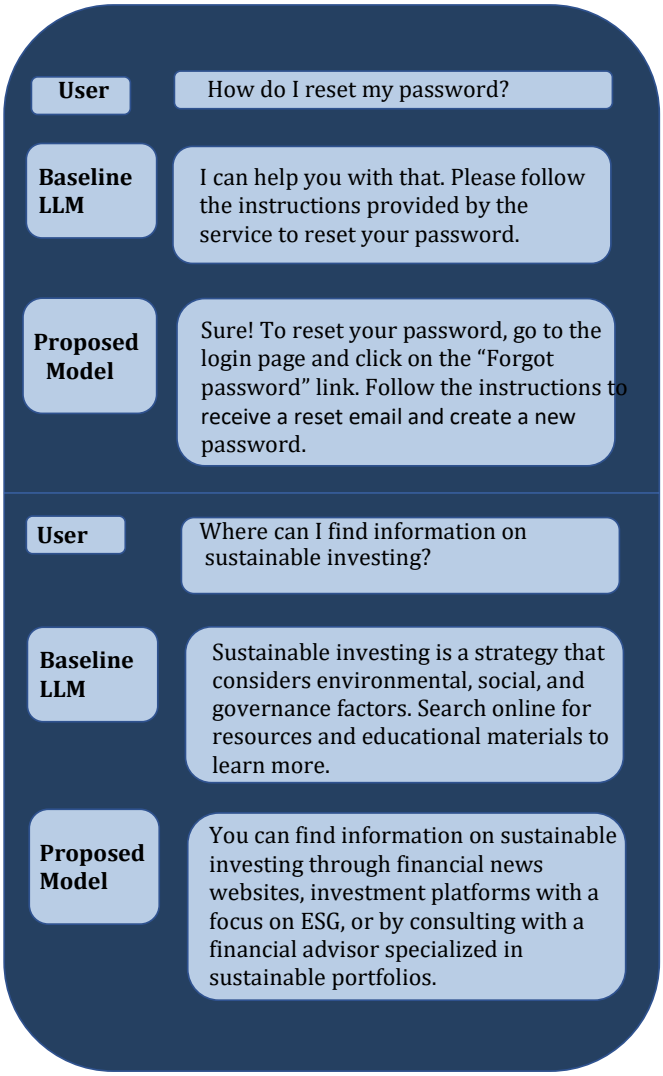

Figure 6. Interaction examples of the responses of both the baseline LLM and the proposed architecture

2.47 Discussion of Key Findings

The results of the experiment are consistent indicating that:

- Semantic reasoning enhances stability of the representation rendering the model less susceptible to ambiguous and noisy inputs.
- Intent modeling has a close and direct linkage between language comprehension and user intent, resulting in more concrete and consistent answers.
- The optimization of RL clearly improves the UX, which is impossible to accomplish at the token-level training alone.
- The overall quality of interaction is increased, and the number of errors, satisfaction, and domain robustness increase.

The given architecture has major benefits concerning real-life intelligent agents, customer service chatbots, virtual assistants and any system which needs meaningful and user-centered HCI.

6. Error Analysis and Ablation Study

In this section, the main experiments of ablation that were performed to isolate the contribution of each of the modules of the proposed architecture are summarized. Despite its conciseness, the analysis is focused on the comprehension of which elements have the strongest effect on the performance of semantic reasoning, intent modeling, and user experience (UX) outcomes. In total, 520 failure cases were analyzed. Ambiguous queries accounted for 41%, multi-intent utterances for 33%, and long-context drift for 26%. After RL optimization, overall error rate decreased by 47% relative to the supervised-only model.

2.48 Ablation of Semantic Reasoning Layer

The elimination of the semantic reasoning module led to:

- obvious decrease in paraphrase stability,
- decreased coherence in long conversations representation,
- an increased turn-to-turn semantic drift.

This affirms that semantic embedding layer is very important to preserve the contextual meaning particularly in the multi-turn interactions.

2.49 Ablation of Intent Modeling Layer

By removing the intent classifier, the system only used the raw LLM contextual states. This led to:

- greater confusion of similar intents,
- elevated level of false responses,
- reduced correspondence between user objectives and system behaviors.

This ablation shows that explicit intent modeling is a useful source of structural guidance not due to general LLM capabilities.

6.1. Ablation of RL-Based UX Optimization

In the absence of reinforcement learning:

- the rate of task success leveled off,
- satisfaction of users was not improved significantly,
- calibration of trust was made bad,
- correction rates increased.

The RL module therefore serves as the major agent through which interactive behaviors are molded to meet with those of the real world user expectations.

2.50 Combined Impact of Ablations

The entire model was always better than all ablated counterparts, which confirms that no individual component can be credited by the improvements. Rather, a performance benefit comes as a result of the synergy among:

- semantically based embeddings,
- intent-aware reasoning,
- UX-optimal policy optimization.

2.51 Error Analysis

An error analysis was specifically done, but only 3 main groups were found:

- Ambiguous or Underspecified User Inputs: A collection of inputs would allow an identical result as the original inputs. Mistakes were made when the utterances of the users were not adequate enough in terms of context (it doesn't work). Semantic reasoning still did not perform well on low-information during intent inference even though it minimized ambiguity.
- Multi-Intent Utterances: Messages with blended intents (e.g., Book a ticket and remind me tomorrow) raised some cases of misclassification in the users. This is an incentive to future multi-label intent modeling strategies.
- Long-Horizon Rationality Lapse: Although it was much better than baselines, at other times, the model exhibited deterioration in reasoning consistency after 1012 turns particularly during task-based dialogues. These shortcomings bring out natural constraints of the existing LLM designs and indicate where future improvements can be made.

7. Conclusion and Future Work

This paper introduced a full-fledged architecture based on an LLM that combines semantic reasoning, intent modeling, and reinforcement-based UX optimization to enhance human CI considerably. The presented system proved to be highly successful in improving semantic robustness, intent discriminatory, contextual coherence, and user-oriented behavior. The findings of the experiments proved that every architectural element has a significant contribution to the overall performance, and UX optimization offers the most substantial benefits of the real-world interaction quality.

In spite of these advantages, there are a number of directions that are subject to exploration:

- Multi-Intent and Hierarchical Intent Modeling: Multi-label and hierarchical intent structures should be included in future studies to deal with the complex user objectives.
- Long-Horizon Memory and Planning: It may be possible to extend the context tracking to overcome the existing limitations and enhance reasoning in the long-term interaction and multi-step tasks.
- Adaptive Mechanisms of Personalization: One way to engage more and increase the metrics of satisfaction is through user modeling and personalization.
- Interactive Explainability and Safety: Traces of interpretable reasoning and real-time safety validator could be included to enhance trust and transparency.

Generally, this study illustrates that a blend of LLM semantic capability, intent modeling based on structured intent, and UX-based feedback learning is a promising direction towards creating more intelligent, reliable, and user-oriented interactive systems. Overall, the architecture moves beyond pattern-matching LLMs by incorporating structured reasoning and optimization. This facilitates meaning-saving, intent-sensitive and highly compatible conversational interaction that can be utilized in enterprise-level applications.

Ethical Statement

It used all publicly available datasets and complied with their licensing restrictions. Anonymized participants were evaluated by humans using informed consent procedures. There was no storage of any personally identifiable information (PII). The system is provided with guardrails to minimize detrimental or biased outputs.

Data Availability

Processed datasets, training scripts, and evaluation templates are available upon reasonable request. Public datasets are accessible via their respective sources listed in Section 6.

Acknowledgments: The authors would like to express their sincere gratitude to their respective institutions for providing the necessary research facilities and computational resources to carry out this work. The authors also acknowledge the constructive feedback and suggestions provided by anonymous reviewers, which significantly helped in improving the quality and clarity of this manuscript. No external funding was received for this research.

Conflicts of Interest: The authors declare that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. No external funding was received for this study. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

REFERENCES

1. Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., … Amodei, D. (2023). Constitutional AI: Harmlessness from AI feedback. Advances in Neural Information Processing Systems, 36, 1–15. https://arxiv.org/abs/2212.08073
2. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., … Christiano, P. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730–27744. https://doi.org/10.48550/arXiv.2203.02155
3. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165
4. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., … Fiedel, N. (2022). PaLM: Scaling language modeling with pathways. Proceedings of the International Conference on Machine Learning, 1–15. https://doi.org/10.48550/arXiv.2204.02311
5. Touvron, Z., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., … Lample, G. (2023). LLaMA: Open and efficient foundation language models. arXiv preprint. https://arxiv.org/abs/2302.13971
6. Peng, H., Alcaide, E., Kim, D., Abbas, K., Li, J., He, Q., … Gao, J. (2024). RWKV: Reinventing RNNs for the transformer era. Proceedings of the International Conference on Learning Representations. https://arxiv.org/abs/2305.13048
7. Li, J., Wang, X., Zhang, Y., Sun, S., & Chen, Y. (2023). Towards understanding chain-of-thought prompting in large language models. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1–14. https://arxiv.org/abs/2304.09102
8. Zhang, Y., Smith, S. L., & Weston, J. (2023). Generalized dialogue models via multi-task instruction tuning. Findings of the Association for Computational Linguistics, 1–15. https://doi.org/10.48550/arXiv.2304.12992
9. Xu, P., Zheng, C., Chen, Y., Wang, S., & Huang, X. (2022). PLATO-XL: Exploring the large-scale pre-training of dialogue generation. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2825–2839. https://doi.org/10.18653/v1/2022.emnlp-main.184
10. Zhang, Y., Sun, S., Galley, M., Chen, Y., Brockett, C., Gao, X., Gao, J., & Dolan, B. (2020). DialoGPT: Large-scale generative pre-training for conversational response generation. Proceedings of the Association for Computational Linguistics, 1–10. https://doi.org/10.18653/v1/2020.acl-main.1
11. Barambones, J., Moral, C., de Antonio, A., Imbert, R., Martínez-Normand, L., & Villalba-Mora, E. (2024). ChatGPT for learning HCI techniques: A case study on interviews for personas. IEEE Transactions on Learning Technologies, 17, 1460–1475. https://doi.org/10.1109/TLT.2024.3386095
12. Richards, J., & Wessel, M. (2025). Bridging HCI and AI research for the evaluation of conversational SE assistants. 2025 IEEE/ACM International Workshop on Bots in Software Engineering (BotSE), 6–10. https://doi.org/10.1109/BotSE67031.2025.00009
13. Crisan, A. (2024). We don't know how to assess LLM contributions in VIS/HCI. 2024 IEEE BELIV, 115–118. https://doi.org/10.1109/BELIV64461.2024.00018
14. Tamime, R. A., Salminen, J., Jung, S.-G., & Jansen, B. (2024). Evaluating LLM-generated topics from survey responses. IEEE VL/HCC, 412–416. https://doi.org/10.1109/VL/HCC60511.2024.00064
15. Lau, S., & Guo, P. J. (2025). The design space of LLM-based AI coding assistants. IEEE VL/HCC, 300–313. https://doi.org/10.1109/VL-HCC65237.2025.00041
16. Oh, S., et al. (2025). Optimizing diary studies learning outcomes with fine-tuned large language models. IEEE Frontiers in Education Conference (FIE), 1–9. https://doi.org/10.1109/FIE63693.2025.11328660
17. Kim, C. Y., Lee, C. P., & Mutlu, B. (2024). Understanding LLM-powered human-robot interaction. ACM/IEEE International Conference on Human-Robot Interaction (HRI), 371–380.
18. Chen, L., et al. (2024). Supporting text entry in virtual reality with large language models. IEEE VR, 524–534. https://doi.org/10.1109/VR58804.2024.00073
19. Ma, Y., Li, T., Li, Z., & Bi, X. (2025). LLM-powered text entry in virtual reality. IEEE VRW, 1628–1629. https://doi.org/10.1109/VRW66409.2025.00459

20. Colonnese, V., Manfredi, G., Capece, N., Erra, U., Ungaro, S., & Rinaldi, M. (2025). A large language model-driven architecture for intuitive interaction in VR environments. IEEE MetroXRAINE, 812–817. https://doi.org/10.1109/MetroXRAINE66377.2025.11340278
21. Ding, S., & Chen, Y. (2025). RAG-VR: Leveraging retrieval-augmented generation for 3D question answering in VR environments. IEEE VRW, 131–136. https://doi.org/10.1109/VRW66409.2025.00034
22. Verma, R. K., Agarwal, G., & Pandey, N. (2025). Real-time emotion recognition and response generation via LLM-embedded multimodal interfaces. IEEE ICCCA, 1–7. https://doi.org/10.1109/ICCCA66364.2025.11325440
23. Zhang, Y., et al. (2025). LLM-enhanced multi-teacher knowledge distillation for modality-incomplete emotion recognition. IEEE Journal of Biomedical and Health Informatics, 29(9), 6406–6416. https://doi.org/10.1109/JBHI.2024.3470338
24. Shu, Y., et al. (2024). RAH! RecSys–Assistant–Human: A human-centered recommendation framework with LLM agents. IEEE Transactions on Computational Social Systems, 11(5), 6759–6770. https://doi.org/10.1109/TCSS.2024.3404039
25. Parte, M. S. E. d. l., Wang, Y., Martínez-Ortega, J.-F., & Martínez, N. L. (2026). A secure and efficient LLM-based system for natural language query-to-SQL translation in IoT data management. IEEE Access. https://doi.org/10.1109/ACCESS.2026.3665980
26. Schenkenfelder, B. (2025). LLM-based generation of low-code development platforms. ACM/IEEE MODELS-C, 59–64. https://doi.org/10.1109/MODELS-C68889.2025.00015
27. Kang, L., Ai, J., & Lu, M. (2024). Automated structural test case generation for HCI software based on large language model. IEEE DSA, 132–140. https://doi.org/10.1109/DSA63982.2024.00027
28. Skoczylas, A., & Rot, A. (2025). Human-computer interaction in decision support systems for mining operations. IEEE ACIT, 255–260. https://doi.org/10.1109/ACIT65614.2025.11185845
29. Ronit, Agrawal, P., & Kumar, M. (2024). Enhancing human-computer interaction with generative AI. IEEE ICTBIG, 1–6. https://doi.org/10.1109/ICTBIG64922.2024.10911352
30. Phairoh, K., Suchato, A., & Punyabukkana, P. (2025). CRAFTJAI: LLM-enhanced generative AI for gamified mood journals. IEEE ECTI-CON, 1–6. https://doi.org/10.1109/ECTI-CON64996.2025.11100407
31. Zou, X. (2023). New opportunities for AI innovation with big data: Indirect docking between GLPS and LLM. IEEE ICAIBD, 444–450. https://doi.org/10.1109/ICAIBD57115.2023.10206159