Characterization and Detection of Incompleteness and Ambiguity in Multi-Turn Interactions with LLMs

Riya Naik
BITS Pilani, K K Birla Goa Campus
Goa, India
p20210056@goa.bits-pilani.ac.in

Ashwin Srinivasan
BITS Pilani, K K Birla Goa Campus
Goa, India
ashwin@goa.bits-pilani.ac.in

Swati Agarwal
PandaByte Innovations Pvt Ltd
India
agrswati@ieee.org

Estrid He
RMIT University
Melbourne, Australia
estrid.he@rmit.edu.au

Abstract

Natural language interaction with computers has been transformed by Large Language Models (LLMs), which now serve as modern-day oracles capable of answering a wide range of queries. Unlike the single-turn interaction with the Delphic oracle, LLMs support multi-turn dialogues where additional context can improve responses. This paper focuses on identifying *incompleteness* and *ambiguity* in user queries during multi-turn interactions with an LLM. Using a simple tagged message exchange model between senders and receivers, we define these properties based on the dialogue sequence. While these definitions help categorize datasets, they cannot be used directly to detect incompleteness or ambiguity. To bridge this gap, we explore the use of Embedding- and Text-based models as detectors. Our experiments on benchmark datasets show that: (a) answer correctness correlates strongly with the presence of incompleteness or ambiguity; (b) we can expect datasets with a high proportion of such questions to have longer multi-turn interactions; (c) effective detectors can be built using only the question and its context. These findings suggest that our proposed approach offers a useful mechanism for characterising datasets, and that trained detectors can be used to automatically identify queries that need to be reformulated before presenting to an LLM.

1 Introduction

Imagine this conversation taking place in 1575. Pope Gregory XIII and the physician Luigi Lilio are discussing dates for Easter:

𝔊: Tell me, Luigi, in your calculation, will next year be a leap year?
LL: Yes Your Holiness, since is divisible in four equal parts.
𝔊: I see. But then, 1500 would have been such a year.
LL: No Your Holiness. There is a correction made every century.
𝔊: Good. I assume that the same correction will be applied in 1600, and it will not be a bisexstile year?
LL: (apologetically) No, Holy Father. There is a further correction once every 400 years, and 1600 will be a bissextile year.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Multi-Turn Interactions in Large Language Models.

We do not know whether the good doctor knew then–as we do now–that if the Holy Father had gone on to ask about 4000 A.D., then a further correction would be needed. The point here is not about the accuracy of the Gregorian calendar, but that if the questions or answers required multiple caveats, then a back-and-forth dialogue would be only natural.

AI systems capable of natural conversation have long been anticipated, both in fiction and in reality. Fictional portrayals often emphasize delivery (for voice) over complexity of content ("Tea. Earl Grey. Recent advances in large language models (LLMs) have however shifted attention to the substance and nature of interactions. LLMs are now integral to many human-computer interaction systems, simulating human-like dialogue and providing intuitive responses to user queries [1].

Early conversational systems were limited to single-turn interactions and struggled with context [2]. Deep learning approaches addressed this by introducing contextual memory for multi-turn dialogue [3]. Modern LLMs retain long conversational histories [4], leading to strong single-turn performance [5, 6]. Techniques such as in-context learning allow even small amounts of feedback (clarificatory examples or domain knowledge) to substantially improve accuracy [7, 8, 9]. These advances suggest LLMs are well-suited for interactive dialogue [10].

Recent work seeks a finer-grained understanding of single vs. multi-turn interactions. For example, Burggräf et al [11] analyze which interaction style drivers find more efficient and show that multi-turn dialogue improves user satisfaction in an automotive setting. Studies emphasize the growing importance of multi-turns in QA systems across domains [12]. To address the scarcity of multi-turn data, Sorathiya et al [13] propose methods to adapt single-turn data for a multi-turn conversation format and empower the training of medical dialogue systems. Baokui et al [14] also show a similar framework that can transform data into multi-turn by linking context, replacing repeated entities with pronouns, and maintaining logical flow without expensive manual effort.

Advancing this line of research, we aim to identify domain-agnostic patterns in human–LLM multi-turn interactions. Three broad perspectives arise. A *mathematical* one views clarification as reducing information-theoretic uncertainty. A *technological* one points to model limitations (e.g., positional encoding). A *behavioral* one infers the need for feedback from the structure of the exchange. We adopt this last view, focusing on two properties of interaction: (a) *Incompleteness*—when a question lacks the information needed to provide any answer; and (b) *Ambiguity*—when a question permits multiple plausible answers.

Although LLMs generate fluent text, they often misinterpret context, requiring clarificatory feedback. Determining what clarification is needed often needs detecting incompleteness or ambiguity or both, which current LLMs are not inherently designed to model [15]. To address incompleteness, Addlesee and Damonte [16] design repair pipelines based on human recovery strategies, while Kumar and Joshi [17] define incomplete questions as those missing topic, adjective, or interrogative components. Ambiguity has been tackled through multiple approaches: graphical representations of query–answer similarity, operator-scope overlap [18], span classification with RoBERTa [19], and LLM-based injection of ambiguous patterns into relational tables [20]. Others treat it as uncertainty in intent [21]. Even so, language models lack a definitive pattern to detect these deficiencies as the context increases. In this paper, we symbolically examine these two deficiencies in a question as properties deducible from the messages exchanged between the human and LLM. For this, we need to first clarify the interaction model we adopt.

2 A Simple Messaging System for Interaction

In this section, we describe a message-based communication mechanism between a pair of agents. Communication between agents consists of messages from a sending agent to a receiving agent.

Def. 1 (Messages) *A message is a 3-tuple of the form* (a, m, b) *where* a *is the identifier of the sender;* b *is the identifier of the receiver; and* m *is a finite length message-string.*

There can be several further categories of message-strings; we list the prominent ones in Table 1. In the rest of the paper, we omit the identifier n as long as the context is clear. In the above definitions, s represents a sequence of message strings. However, in this paper, in $?(s)$ $|s| = 1$ (that is, only 1 question is allowed at a time). There can be multiple answers or even no answers for a question. That is, in $!(s)$, $|s| \geq 0$. Additionally, in $\top(s)$, we will require $|s| \geq 1$. If ordering is unimportant, we will

sometimes show s as a set instead of a sequence. In all cases, if $|s| = 1$, we will simply denote the message by the singleton element, dispensing with the sequence (or set) notation.

Table 1: Categories of message strings and their descriptions

Category	Message String	Description
Termination	$m = \square$	Sender is terminating communication with the receiver.
Question	$m = ?_n(s)$	Sender sends question s with identifier n to the receiver.
Answer	$m = !_n(s)$	Sender sends answer s with identifier n to the receiver.
Statement	$m = \top(s)$	Sender sends a statement s to the receiver.

A pair of back-and-forth message exchanges makes up a turn, and one or more such turns between two agents form an interaction.

Def. 2 (Interaction) *A (1-step) turn from agent a to agent b is the pair of messages (M_1, M_2), where $M_1 = (a, m_1, b)$, $M_2 = (b, m_2, a)$, and $m_1 \neq \square$. A k-step turn is the sequence $\langle T_1, T_2, \cdots, T_k \rangle$, where T_i $(1 \leq i \leq k)$ is a 1-step turn from a to b. Similarly for 1-step and k-step turns from b to a. We will call a sequence of 1 or more turns between a and b an interaction between a and b.*

We note that each turn consists of a sequence of 2 messages. Thus, with every interaction consisting of k turns $\langle T_1, \cdots, T_k \rangle$ there exists a corresponding sequence $\langle M_1, M_2, \cdots, M_{2k-1}, M_{2k} \rangle$ messages and $\langle m_1, m_2, \cdots, m_{2k-1}, m_{2k} \rangle$ message-strings. By construction the sequence $\langle m_1, m_3, \ldots, m_{2k-1} \rangle$ will be from agent a to agent b, and $\langle m_2, m_4, \ldots, m_{2k} \rangle$ will be from agent b We denote these as $\langle m_{ab} \rangle$ and $\langle m_{ba} \rangle$ for short. These interaction sequences allow us to define the *context* for an agent. We assume any agent has access to a (possibly empty) set of prior statements, which we call *background knowledge*.

Def. 3 (Context) *Let a and b be agents with background knowledge B_a and B_b respectively, prior to any interaction. Without loss of generality, let $(T_1, T_2, \cdots, T_k)$ be a k-step interaction from a to b. We denote the context for a on the i^{th} turn T_i as $C_{a,i} = B_a \cup \{m_1, m_2, \cdots, m_{2i-2}\}$, and the context for b on the i^{th} turn $C_{b,i} = B_b \cup \{m_1, m_2, \cdots, m_{2i-1}\}$.*

In this paper, we are interested in question-answer sequences occurring in an interaction. These are obtainable by examining the messages exchanged.

Def. 4 (Questions and Answers) *Let $(T_1, \cdots, T_k)$ be a k-step interaction between a and b, and $(m_1, m_2, \cdots, m_{2k-1}, m_{2k})$ be the corresponding message-strings. Let m_{ab} and m_{ba} be the message-string sequences from a to b and vice-versa. Let QA_{ab} be the sequence $((q_1, a_1), \cdots, (q_j, a_j))$ s.t.: (1) for every (q_i, a_i) in QA_{ab}, $?_{\alpha_i}(q_i) \in m_{ab}$; and (2) $a_i = \cup\, !_{\alpha_i}(s)$, where $!_{\alpha_i}(s) \in m_{ba}$. We will call QA_{ab} the question-answer sequence for the interaction between a and b. Similarly for a question-answer sequence from b to a. It is sometimes helpful to identify the set of questions sent by a to b, or Q_{ab} as = $\{Q : (Q, \cdot) in \langle QA_{ab} \rangle\}$. A similar set of questions from b to a can also be identified.*

We define a special agent Δ called the *oracle*. The oracle's answers are taken to be always correct. The oracle is assumed to know everything.

Remark 1 (Interaction with the Oracle) *We note the following special features of the oracle: 1. Δ knows everything up to the present, including the content of conversations between any non-oracular agents; 2. Only a 1-step interaction is allowed between a non-oracular agent a and Δ. The interaction consists of a turn T where: $T = ((a, ?q, \Delta), (\Delta, !(s), a))$; or $T = ((a, ?q, \Delta), (\Delta, \square, a))$. 3. The answer(s) provided by Δ are always correct.*

The oracle allows us to categorize questions and answers as incomplete and ambiguous.

Def. 5 (Incomplete Question) *Without loss of generality, let $(T_1, T_2, \cdots, T_k)$ be a k-step interaction from a to b. Let $C_{b,i}$ denote the context for b on the i^{th} turn. Let $T_i = ((a, ?(q), b), \cdot)$, where agent a sends question q to b. Let $((b, ?(q), \Delta), (\Delta, !(s), b))$ be an interaction between b and Δ. If $s = \square$, we conclude that q is incomplete. In such a case, we will also say it is incomplete for b given $C_{b,i}$.*

That is, a question is incomplete, if the oracle does not give an answer. This is because if the oracle cannot provide a correct answer, neither can b. Similarly:

Def. 6 (Ambiguous Question) *Without loss of generality, let $(T_1, T_2, \cdots, T_k)$ be a k-step interaction from a to b. Let $C_{b,i}$ denote the context for b on the i^{th} turn. Let $T_i = ((a, ?(q), b), \cdot)$, where agent a sends question q to b. Let $((b, ?(q), \Delta), (\Delta, !(s), b))$ be an interaction between b and Δ. If $|s| > 1$ then we say q is ambiguous. In such a case, we will also say q is ambiguous for b given $C_{b,i}$.*

That is, a question is ambiguous if the oracle returns more than one answer. (we assume that questions that are either incomplete or ambiguous, but not both at once). We are still left with the impractical requirement of needing to consult the oracle to decide whether a question is one or the other. We propose the following tests to detect the possible presence of incompleteness and ambiguity using just the interaction sequence between non-oracular agents.

Def. 7 (Possibly Incomplete Question) *Let $\mathcal{I} = (T_1, T_2, \cdots, T_k)$ be a k-step interaction between a and b. Let $T_i = ((a, ?_\alpha(q), b), (b, ?_\beta(s1), a))$; where $?_\beta(s1)$ asks for missing data which adds information to the existing information in the question, and $T_{i+1} = ((a, !_\beta(s2), b), (b, s3, a))$, where $s3$ can be any statement. Then we will say q is a possibly incomplete question given $\mathcal{I}$; or (equivalently) interaction $\mathcal{I}$ between a and b has a possibly incomplete question on turn i.*

Def. 8 (Possibly Ambiguous Question) *Let $\mathcal{I} = (T_1, T_2, \cdots, T_k)$ be a k-step interaction between a and b. Let $T_i = ((a, ?_\alpha(q), b), (b, !_\alpha(s1), a))$; where $!_\alpha(s1)$ either answers incorrectly or clarifies the interpretation of the existing question and $T_{i+1} = ((a, \top(s2), b), (b, s3, a))$. Then we will say q is a possibly ambiguous question given $\mathcal{I}$; or (equivalently) interaction $\mathcal{I}$ between a and b has a possibly ambiguous question on turn i.*

These definitions for incompleteness and ambiguity can only be used with message-sequences with at least one additional turn after the question has been sent. In practice, if we want to intervene automatically to mitigate these properties, then we need to be able to identify the properties simply from the question and the prior context.

Def. 9 (Detector Function) *Let $\mathcal{I} = (T_1, T_2, \cdots, T_k)$ be a k-step interaction between a and b. Let $T_i = ((a, ?_\alpha(q), b), (b, ?_\beta(s1), a))$. Let $C_{a,i}$ denote the context for a on the i^{th} turn. A detector function is the classifier:*

$$h(q|C_{a,i}) = \begin{cases} incomplete & \text{if } q \text{ is possibly incomplete given } \mathcal{I} \\ ambiguous & \text{if } q \text{ is possibly ambiguous given } \mathcal{I} \\ normal & \text{otherwise} \end{cases}$$

That is, the detector is a function that classifies a question into one of 3 categories, just based on the question and context of the sender up to the question. In the experiments described next, we will construct detector functions using (training) data from some standard benchmarks.

3 Empirical Evaluation

Using benchmark datasets, our experimental goals are to answer the following concerning the "starter-question" posed by the human to an LLM:

(A) What is the relation between answer-correctness on a dataset and starter-question deficiency (measured by the proportion of incomplete or ambiguous starter-questions)?

(B) What is the relation between multi-turn interactions' length and starter-question deficiency?

(C) Can we build good detectors for possibly incomplete and possibly ambiguous starter-questions, using only the question and its prior context (if any)?

In (A) and (B), a starter-question is taken to be deficient if it is identified as being possibly incomplete or possibly ambiguous using Defns. 7 and 8. Recall that this requires information from the first 2 turns. In contrast, (C) attempts to do this using just the message sent from the human to the LLM.

3.1 Materials

Datasets: In our empirical study, we evaluate the QA systems on six datasets with different characteristics. 1) **SQuAD** (Stanford Question Answering Dataset), which is a widely used dataset for machine reading comprehension, consisting of over 100K questions based on Wikipedia articles[22, 23]. 2) **NQ-open** (Natural Questions open), which is a large-scale benchmark, featuring open domain real user queries and answers annotated from Wikipedia articles [24]. 3) **AmbigNQ** dataset is designed to handle ambiguous questions, focusing on event and entity references, covering 14,042 NQ-open questions[25, 24]. 4) **ShARC** (Shaping Answers with Rules through Conversation), a multi-turn dataset that focuses on 32K task-oriented conversations with reasoning covering multiple domains [26]. 5) **MultiWOZ** (Multi-domain Wizard-of-Oz), a dataset covering multiple domains such as hotels, restaurants, and taxis with 8438 task-oriented dialogues [27]. 6) **MedDialog** covers 0.26 million conversations between patients and doctors curated to understand real-world medical queries [28]. The first, fourth, and fifth datasets include questions along with the relevant context to provide answers, while the first, third, and sixth datasets consist of question-answer pairs.

Algorithms and Machines: We use the following models and software: (a) GPT-3.5-Turbo, GPT-4o, and Llama-4-Scout: LLMs used to test question labeling in single and multiple-turn interaction settings. (b) text-embedding-3-large: OpenAI embedding model to compute vector representations for the question and context. All implementations are in Python 3.10, with API calls to the model engine. Our experiments are conducted on a workstation based on Linux (Ubuntu 22.04) with 256GB of RAM, an Intel i9 processor, and an NVIDIA A5000 graphics processor with 24GB of memory.

3.2 Method

3.2.1 Preliminaries

IA Diagram and Opacity of Datasets: An *IA Diagram* depicts incompleteness and ambiguity in datasets. Let D be a set of interactions $\mathcal{I} = (T_1, \ldots, T_k)$ with $k \geq 1$ turns, and $\mathcal{I}_{i,j} = (T_i, \ldots, T_j)$, $j \leq k$. We define $D_{i,j} = \{\mathcal{I}_{i,j} : \mathcal{I} \in D\}$. Any $D_{i,j}$ maps to $(i, a) \in [0, 1]^2$, with i the fraction of (sub-)sequences with incomplete questions, a the fraction with ambiguous ones. The following quadrants are helpful: $Q1 = [0, 0.5]^2$ (low-i,low-a), $Q2 = [0, 0.5] \times (0.5, 1]$, $Q3 = (0.5, 1] \times [0, 0.5]$, and $Q4 = (0.5, 1]^2$ (high-i,high-a). For $\vec{d} = (i, a)$, *opacity* defined as $\frac{||\vec{d}||}{\sqrt{2}}$, is a normalized *(incompleteness, ambiguity)* measure. In the definitions we use, a question cannot be both (possibly) incomplete and (possibly) ambiguous, $i + a \leq 1 \Rightarrow ||\vec{d}|| \leq 1$ and the maximum opacity of a dataset is $1/\sqrt{2} \approx 0.71$. Here, only $D_{1,2}$ (starter question) is used to estimate incompleteness and ambiguities (Fig. 1 and Table.2).

Dataset	I	A	Opacity
SQuAD	0.00	0.08	0.06
NQ-Open	0.02	0.17	0.12
AmbigNQ	0.01	0.36	0.25
ShARC	0.28	0.61	0.47
MultiWOZ	0.21	0.75	0.55
MedDialog	0.92	0.08	0.65

Table 2: Estimates of proportions of incompleteness I and ambiguity A

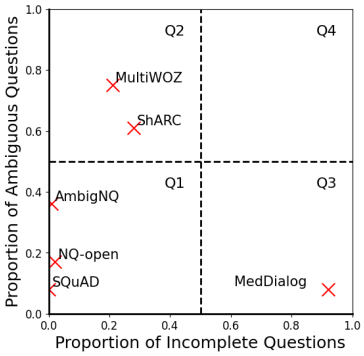

Figure 1: Corresponding IA diagram. Estimates are of "starter-questions", obtained using $D_{1,2}$ for each dataset D

Constructing Detectors: Before we construct detectors, we randomly partition each dataset into a set of sequences constituting a training set and interactions constituting a test set. We consider two classification configurations: (i) a three-class setting, where $y \in$ *incomplete*, *ambiguous*, *normal*; and

(ii) a two-class setting, where the *incomplete* and *ambiguous* categories are combined into a single *deficient* class, contrasted against the *normal* class.

Embedding-based detectors: Interaction data are converted into $(\mathbf{x}, y)$ pairs for training and testing. Here, $\mathbf{x}$ is the embedding of the starter question plus any prior context, computed using OpenAI's `text-embedding-3-large` model with embedding dimension 3072, which produces an embedding of shape $(N, 3072)$, where N is the total number of instances. Embedding-based detectors are trained on the resulting pairs. Here we construct detectors using: (a) MLPs; (b) Random Forests (RF); and (c) XGBoost (XGB). Given the relatively modest size of the instances, these methods provide a balanced trade-off between accuracy, generalization, and resource efficiency. The MLP consists of two fully connected hidden layers with ReLU activation functions, followed by a softmax output layer for classification. The network is optimized using the Adam optimizer. The RF model is an ensemble of 100 decision trees, with bootstrapped samples and randomized feature selection. XGB, a gradient boosting–based classifier, is configured for multi-class classification with an objective function set to binary:logistic or multi:softmax. Here, we assess whether simple embedding-based detectors can serve as viable alternatives to more resource-intensive text-based approaches.

Text-based detector: A detector can be constructed with a language model, directly using the text of a question and any prior context. We treat the class-label arising as a result of completing the sentence given a question and it's context ("the label of question $\langle q \rangle$ given context $\langle c \rangle$ is ..."). The context can be enriched by the addition of a small number of questions with prior labels (the "few-shot" setting) (See Appendix A.1 for the prompts). We distinguish two ways to construct the few-shots: (a) *Direct class sampling:* selecting a fixed number of examples directly from each class; (b) *k-NN retrieval:* using similarity-based retrieval to select the k nearest training examples for a given test instance (we use cosine similarity to find the k training instances most similar to the test instance). These instances are used as few-shot for classification (See Table. 7 for detailed comparison). We evaluate the best of these configurations against zero-shot.

3.2.2 Evaluation

Let h represent a human agent posing questions and λ denote the LLM used to answer the questions. We assume datasets $D_1, D_2, \ldots, D_n$, each consisting of a set of (q, a) pairs, where q denotes a question string, and a denotes a (correct) answer string (any initial context is assumed to be part of the question string). The method adopted to answer questions (A)–(C) in Sec. 3 follows:

1. Randomly split D_i into training and test subsets ($D_{i,tr}$ and $D_{i,te}$). Construct aforementioned detectors f_δ using training data $\bigcup_{i=1}^{n} D_{i,tr}$, where $\delta \in \{MLP, RF, XGB, LLM-ZS, LLM-FS\}$ ("ZS" and "FS" refers to zero-shot and few-shot respectively);
2. For each dataset $d \in \{D_1, \ldots, D_n\}$:
   (a) Let $d^{(k)} = \{(q, a, \mathcal{I}) : (q, a) \in d, \mathcal{I} \text{ is an interaction between } h \text{ and } \lambda\}$. We assume $\mathcal{I}$ to be restricted to k-turns if the correct answer is not obtained (here, k is 3)
   (b) Obtain accuracy for $j = 1 \ldots k$ turns using the interaction information in $d^{(k)}$:
      i. Let $Correct_{d,j} = \{(q, a, \mathcal{I}) : (q, a, \mathcal{I}) \in d^{(k)}, |\mathcal{I}| = j, \mathcal{I} \text{ ends with a correct answer}\}$
      ii. $Acc_{d,j} = |Correct_{d,j}|/|d|$
   (c) For each $(q, a, (T_1, T_2, \ldots, T_k)) \in d^{(k)}$:
      i. Obtain estimates of incompleteness (I), ambiguity(A), and hence opacity using (T_1, T_2) and Defns. 7, 8 ($I = A = 0$, if $k = 1$);
      ii. Obtain estimates of incompleteness, ambiguity and hence opacity using (T_1) and f_δ (the detectors).
      iii. If $(q, a) \in d_{te}$ then update estimates of cross-comparison between labels obtained using Defns. 7, 8 against the labels obtained using f_δ
   (d) Answer questions (A)–(C) using the estimates in Step 2.(c)ii above

Additional relevant experimental details are provided in Appendix A section of this paper.

4 Results

The principal experimental results are in Figs. 2, 3 and in Tables. 3, 4. Key findings include:

(F1) There is very strong evidence of a negative association between the opacity of a dataset and the accuracy of the answer from the LLM on the first turn (Spearman's rank correlation between the two variables $R_S = -1.0, p < 0.01$).

(F2) There is strong evidence of a positive association between opacity of a dataset and the error in the LLM's answer after 3 turns ($R_S == 0.94, p < 0.05$);[1]

(F3) Reasonably accurate detectors can be constructed using embeddings of question from the first turn of the interaction. There is no significant difference in accuracy among the embedding-based classifiers, and their accuracies are significantly higher than those of the text-based classifiers. ($p < 0.01$). There is significant evidence of a very strong positive association between the actual and the predicted Opacity of the dataset ($R_S = +1.0, p < 0.01$).

Using opacity of a dataset as an aggregate measure of the deficiency of a dataset (based on the proportion of incompleteness or ambiguous starter questions identified using Defns. 7 and 8), we now turn to questions (A)–(C) in Sec. 3. Finding F1 is directly relevant to question (A). It suggests that more deficient datasets are likely to have lower answer-accuracy. F2 is indirectly relevant to question (B). If it is reasonable to assume that the error in LLM-response at any point in a multi-turn interaction is directly related to the number of additional turns needed to correctly answer the starter-question, then F2 suggests that more deficient datasets are likely to have longer interactions. F3 directly relates to question (C). It suggests that the deficiency of a dataset can be predicted with reasonable accuracy using the starter-question and any prior context available. The strong positive association suggests, we would arrive at the same conclusions regarding (A) and (B) if the opacity based on the detector's prediction was used as a proxy for those obtained from Defns. 7 and 8.

Dataset	(I, A)	Opacity	Turn (j)	Accuracy
SQuAD	(0.00,0.08)	0.06	1	0.92
			2	0.95
			3	0.97
NQ-open	(0.02,0.17)	0.12	1	0.81
			2	0.87
			3	0.89
AmbigNQ	(0.01,0.36)	0.26	1	0.63
			2	0.69
			3	0.78
ShARC	(0.28,0.61)	0.48	1	0.11
			2	0.60
			3	0.83
MultiWOZ	(0.21,0.75)	0.55	1	0.04
			2	0.25
			3	0.48
Med Dialog	(0.92,0.08)	0.66	1	0.00
			2	0.18
			3	0.26

Table 3: Deficiency is measured by incompleteness (I) and ambiguity (A). Opacity combines (I, A) into a single score. Accuracy is the fraction of correct interactions after turn j.

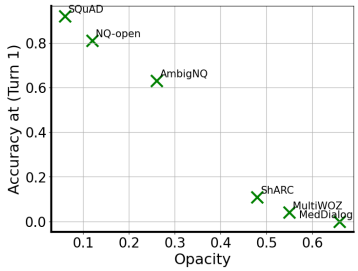

Figure 2: Opacity vs Accuracy on the first turn

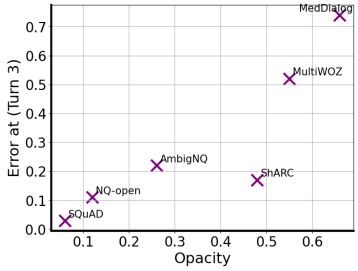

Figure 3: Opacity vs Error on the third turn.

The following additional aspects of the experiments are noteworthy:

Datasets. SQuAD, NQ-open, and AmbigNQ are largely fact-based. As expected, their opacities are low, and correct answers are usually obtained in one turn. AmbigNQ, however, was derived from NQ-open by deliberately introducing ambiguity, which is reflected in its much higher opacity (Table.3); thus, correct answers are less likely in a single turn. MedDialog is particularly challenging, with very high opacity, since questions often lack the necessary symptoms.

Context. Context (Def.3) reduces incompleteness and ambiguity in two ways. First, removing initial context raises the proportion of incomplete or ambiguous questions (Table 6 in the Appendix;

[1]The slight drop is due to the unusually high number of interactions in ShARC terminating after the 2nd turn.

Table 4: (a) Accuracy of detectors with the standard deviation; and (b) Comparison of predicted and actual deficiencies of the test datasets for 2- and 3-class detectors. Each entry compares (I, A) values obtained using proposed definitions on the first two turns and predictions obtained using the MLP 3-class detector. Differences from the Actual values in Table 3 arise because values (b) are computed from the test data.

(a)

Type	Detector	Accuracy	
		2-class	3-class
Embedding	MLP	0.80(0.02)	0.74(0.02)
	RF	0.78(0.02)	0.70(0.02)
	XGB	0.81(0.02)	0.73(0.02)
Text	LLM-ZS	0.68(0.02)	0.51(0.02)
	LLM-FS	0.71(0.02)	0.60(0.02)

(b)

Dataset	(I,A)		Opacity	
	Actual	Predicted	Actual	Predicted
SQuAD	(0.00,0.10)	(0.02,0.12)	0.07	0.09
NQ-open	(0.10,0.15)	(0.17,0.07)	0.13	0.13
AmbigNQ	(0.05,0.32)	(0.02,0.32)	0.23	0.23
ShARC	(0.20,0.42)	(0.15,0.35)	0.33	0.27
MultiWOZ	(0.32,0.62)	(0.40,0.52)	0.49	0.46
MedDialog	(0.90,0.10)	(0.97,0.02)	0.64	0.69

MedDialog, AmbigNQ, and NQ-Open are excluded as they have none). Second, as interaction length increases, more context becomes available. Adding turns as initial context lowers the proportion of incomplete and ambiguous questions (Table 5), enabling many to be resolved in a single turn.

LLMs. To test LLM-dependence, we repeated the runs (originally with GPT-3.5-Turbo) using Llama-4-Scout (Table 5). As the turns increase, the model receives more context. The column *Accuracy* reports the proportion of correct answers after the corresponding number of turns. A clear negative association is observed between opacity and accuracy over successive turns. We also compared various LLM models as text-based detectors, and the k-NN-based few-shot approach with Llama-4-Scout achieved the highest accuracy (Table 7 in Appendix). The trends remain consistent, suggesting our conclusions about Questions (A)–(C) generalize across models.

Table 5: Comparison of GPT-3.5-Turbo and Llama-4-Scout across datasets, showing how larger context (k) affects proportions of incomplete and ambiguous interactions (per Defns.7, 8).

	Context from	**GPT-3.5-Turbo**				**Llama-4-Scout**			
Dataset	Turn (k)	Incomplete	Ambiguous	Accuracy	Opacity	Incomplete	Ambiguous	Accuracy	Opacity
SQuAD	1	0.00	0.08	0.92	0.06	0.01	0.06	0.93	0.04
	2	0.00	0.05	0.95	0.04	0.01	0.05	0.94	0.04
	3	0.00	0.03	0.97	0.02	0.01	0.03	0.96	0.02
NQ-open	1	0.02	0.17	0.81	0.12	0.13	0.19	0.68	0.16
	2	0.00	0.13	0.87	0.09	0.07	0.07	0.86	0.07
	3	0.00	0.11	0.89	0.08	0.06	0.08	0.86	0.07
AmbigNQ	1	0.01	0.36	0.63	0.26	0.18	0.25	0.57	0.22
	2	0.00	0.31	0.69	0.22	0.10	0.13	0.77	0.12
	3	0.00	0.22	0.78	0.16	0.05	0.13	0.82	0.10
ShARC	1	0.28	0.61	0.11	0.48	0.57	0.29	0.14	0.45
	2	0.02	0.38	0.60	0.27	0.09	0.22	0.69	0.17
	3	0.01	0.16	0.83	0.11	0.02	0.13	0.85	0.09
MultiWOZ	1	0.21	0.75	0.04	0.55	0.55	0.38	0.07	0.47
	2	0.19	0.56	0.25	0.42	0.29	0.49	0.22	0.40
	3	0.18	0.34	0.48	0.27	0.17	0.34	0.49	0.27
MedDialog	1	0.92	0.08	0.00	0.65	0.92	0.08	0.00	0.65
	2	0.21	0.61	0.18	0.46	0.50	0.40	0.10	0.45
	3	0.18	0.56	0.26	0.42	0.19	0.52	0.29	0.39

5 Conclusion

Our focus is on natural language QA systems. Natural language interfaces have long been sought, and substantial recent progress has come with LLMs. Even so, challenges remain to identify when an interaction requires clarificatory feedback using conversational turns, particularly to detect *incompleteness* and *ambiguity* in questions. We treat these as properties of exchanged messages and propose an aggregate measure, *opacity*, capturing their prevalence. High-opacity datasets are those where interactions often begin with incomplete or ambiguous questions. Empirically, such datasets yield weaker initial answers and longer interactions. We show that an MLP-based detector can identify

problematic starter questions with $\approx 75\%$ accuracy. The detector yields opacity estimates close to the definitions, as sentence-level errors smooth out. Having such detector enables us to develop mitigation strategies, such as agents that reformulate questions. There are several ways in which the work here could be extended. More datasets should be tested, and agent-based pipelines could combine a detector of the kind we have constructed with a reasoning agent to repair defects, possibly with retrieval-augmented methods [29]. Our current experiments address only starter questions, but the definitions extend to later turns, where issues may further degrade response quality. Conceptually, we assumed incompleteness and ambiguity are mutually exclusive, but they may co-occur. Generalizing the definitions would allow a finer-grained analysis of multi-turn LLM interactions.

References

[1] Pradnya Kulkarni, Ameya Mahabaleshwarkar, Mrunalini Kulkarni, Nachiket Sirsikar, and Kunal Gadgil. Conversational ai: An overview of methodologies, applications & future scope. In *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, pages 1–7. IEEE, 2019.

[2] Seon-Ok Na, Young-Min Kim, and Seung-Hwan Cho. Insurance question answering via single-turn dialogue modeling. In Xianchao Wu, Peiying Ruan, Sheng Li, and Yi Dong, editors, *Proceedings of the Second Workshop on When Creative AI Meets Conversational AI*, pages 35–41, Gyeongju, Republic of Korea, October 2022. Association for Computational Linguistics.

[3] Zhou Zhao, Xinghua Jiang, Deng Cai, Jun Xiao, Xiaofei He, and Shiliang Pu. Multi-turn video question answering via multi-stream hierarchical attention context network. In *IJCAI*, volume 2018, page 27th, 2018.

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[5] Siqing Huo, Negar Arabzadeh, and Charles Clarke. Retrieving supporting evidence for generative question answering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 11–20, 2023.

[6] Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family. In *International Semantic Web Conference*, pages 348–367. Springer, 2023.

[7] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[8] Pranjal Chitale, Jay Gala, and Raj Dabre. An empirical study of in-context learning in llms for machine translation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7384–7406, 2024.

[9] Xinzhe Li. A review of prominent paradigms for LLM-based agents: Tool use, planning (including RAG), and feedback learning. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9760–9779, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[10] Taufiq Daryanto, Xiaohan Ding, Lance T Wilhelm, Sophia Stil, Kirk McInnis Knutsen, and Eugenia H Rho. Conversate: Supporting reflective learning in interview practice through interactive simulation and dialogic feedback. *Proceedings of the ACM on Human-Computer Interaction*, 9(GROUP):1–32, 2025.

[11] Peter Burggräf, Moritz Beyer, Jan-Philip Ganser, Tobias Adlon, Katharina Müller, Constantin Riess, Kaspar Zollner, Till Saßmannshausen, and Vincent Kammerer. Preferences for single-turn vs. multiturn voice dialogs in automotive use cases—results of an interactive user survey in germany. *IEEE Access*, 10:55020–55033, 2022.

[12] Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. Conversational question answering: a survey. *Knowledge and Information Systems*, 64(12):3151–3195, 2022.

[13] Nazib Sorathiya, Chuan-An Lin, Daniel Chen Daniel Xiong, Scott Zin, Yi Zhang, He Sarina Yang, and Sharon Xiaolei Huang. Multi-turn dialog system on single-turn data in medical domain. *arXiv e-prints*, pages arXiv–2105, 2021.

[14] Baokui Li, Sen Zhang, Wangshu Zhang, Yicheng Chen, Changlin Yang, Sen Hu, Teng Xu, Siye Liu, and Jiwei Li. S2m: Converting single-turn to multi-turn datasets for conversational question answering. In *ECAI 2023*, pages 1365–1372. IOS Press, 2023.

[15] Md Mehrab Tanjim, Yeonjun In, Xiang Chen, Victor S Bursztyn, Ryan A Rossi, Sungchul Kim, Guang-Jie Ren, Vaishnavi Muppala, Shun Jiang, Yongsung Kim, et al. Disambiguation in conversational question answering in the era of llm: A survey. *arXiv preprint arXiv:2505.12543*, 2025.

[16] Angus Addlesee and Marco Damonte. Understanding and answering incomplete questions. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, pages 1–9, 2023.

[17] Vineet Kumar and Sachindra Joshi. Incomplete follow-up question resolution using retrieval based sequence to sequence learning. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*, pages 705–714, 2017.

[18] Gaurav Kamath, Sebastian Schuster, Sowmya Vajjala, and Siva Reddy. Scope ambiguities in large language models. *Transactions of the Association for Computational Linguistics*, 12:738–754, 2024.

[19] Nathan Ellis Rasmussen. *Broad-Domain Quantifier Scoping With RoBERTa*. The Ohio State University, 2022.

[20] Simone Papicchio, Paolo Papotti, and Luca Cagliero. Evaluating ambiguous questions in semantic parsing. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 338–342. IEEE, 2024.

[21] Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore, December 2023. Association for Computational Linguistics.

[22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[23] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[24] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[25] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online, November 2020. Association for Computational Linguistics.

[26] Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. Interpretation of natural language rules in conversational machine reading. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[27] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[28] Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. MedDialog: Large-scale medical dialogue datasets. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250, Online, November 2020. Association for Computational Linguistics.

[29] Aigerim Mansurova, Aiganym Mansurova, and Aliya Nugumanova. Qa-rag: Exploring llm reliance on external knowledge. *Big Data and Cognitive Computing*, 8(9):115, 2024.

[30] Max Peeperkorn, Tom Kouwenhoven, Daniel Brown, and Anna Jordanous. Is temperature the creativity parameter of large language models? *CoRR*, 2024.

A Appendix

A.1 Experimental Details

Initial Context

We examine the role of initial context or background knowledge, which can resolve incompleteness and ambiguity in the question. We perform this experiment on SQuAD, ShARC, and MultiWOZ, which were accompanied by the supporting context using GPT-3.5-turbo, and we observe that the initial context helps in reducing deficiencies at a certain level (See Table 6). But to improve it further, we have to incorporate turn-based context.

Table 6: Role of initial context on the proportions of incomplete and ambiguous questions.

Dataset (d)	**Initial Context**	**Incomplete** (PI_d)	**Ambiguous** (PA_d)	**Correct (after 1 turn)**
SQuAD	No	0.09	0.45	0.46
	Yes	0.00	0.08	0.92
ShARC	No	0.63	0.27	0.10
	Yes	0.28	0.61	0.11
MultiWOZ	No	0.83	0.15	0.02
	Yes	0.21	0.75	0.04

Human classification of Incompleteness and Ambiguity

We make API calls to GPT-3.5-Turbo and Llama-4-scout with the temperature set to 0.7, as it provides a balanced trade-off between creativity and reliability while generating text [30]. Using the question and context, we assemble a prompt with instructions. The LLM uses this prompt to generate responses. We, as human agents, respond with clarification to improve the quality and accuracy of the final answer. For the quantitative comparison, the proportions PI_d and PA_d are compared to assess the alignment of the model's responses with predefined patterns (Table. 5). This comparison provides insights into the distribution of interactions across different categories that seek multiple interactions. Since incomplete and ambiguous questions are only properties defined on interactions with at least 2 turns, $PA_d = PI_d$ will be 0 if all questions can be answered correctly in 1 turn. For questions requiring longer interactions, we estimate the proportion of multi-turn interactions in which the question initiating the interaction is either incomplete or ambiguous (to the extent defined by Defns. 7 and 8).

Language Model Selection as Detectors

We compare three LLMs here: GPT-3.5-turbo, GPT-4o, and Llama-4-Scout. Model performance is evaluated by tuning hyperparameters such as temperature, class-wise few-shot, and k-NN examples. In our experiments, we find the optimal configuration at a temperature of 0.0 with four shots per class and top-5 similar examples. We compare these configurations with zero-shot in Table 7. We select the model with the highest accuracy and the least delay, which is LLaMA-4-Scout with K-NN based few-shot.

Table 7: LLM performance across different few-shot approaches. Accuracy (%) and time per sample (min) are reported for zero-shot (ZS) and few-shot (FS): class-wise samples and k-NN retrieval, highlighting the trade-off between accuracy and computation.

		2-class		3-class	
Approach	Model	Accuracy	Time per Sample	Accuracy	Time per Sample
ZS	GPT-3.5-turbo	61.38	0.71	49.70	0.80
	GPT-4o	69.00	0.83	49.57	1.17
	Llama-4-Scout	68.00	0.17	51.00	0.17
FS	GPT-3.5-turboo	52.58	0.71	54.08	0.74
	GPT-4o	61.59	0.76	57.08	1.13
	Llama-4-Scout	64.59	0.18	51.86	0.41
FS k-NN	GPT-3.5-turbo	68.24	0.69	60.30	0.78
	GPT-4o	69.30	0.80	57.51	1.01
	Llama-4-Scout	**71.24**	**0.15**	**60.18**	**0.16**

We employ the following prompts for 2 and 3-class problems:

Zero-shot Prompt

```
prompt_2_class =
f"""You are a text classifier.
Classify the given question as either
'normal' or 'deficient'.

Question: {question}
Context: {context}

Answer with only one word:
'normal' or 'deficient'

Answer:"""
```

```
prompt_3_class =
f"""You are a text classifier.
Classify the given question as
'normal' or 'incomplete' or 'ambiguous'.

Question: {question}
Context: {context}

Answer with only one word:
'normal' or 'incomplete' or 'ambiguous'

Answer:"""
```

Few-shot Prompt

```
prompt_2_class =
f"""You are a text classifier.
Based on the examples below,
classify the given question
as either 'normal' or 'deficient'.

Examples: {example_text}
Now classify this text.

Question: {question}
Context: {context}

Must answer with only one word:
'normal' or 'deficient'

Answer:"""
```

```
prompt_3_class =
f"""You are a text classifier.
Based on the examples below,
classify the given question
as 'normal' or 'incomplete' or 'ambiguous'

Examples: {example_text}
Now classify this text.

Question: {question}
Context: {context}

Must answer with only one word:
'normal' or 'incomplete' or 'ambiguous'

Answer:"""
```

A.2 Examples: Incompleteness and Ambiguity in LLM-Human interaction

Tables 8 and 9 illustrate interaction examples from the ShARC dataset between a human agent and an LLM. The Table 8 highlights an example of an incomplete question. It shows how the model requests for clarification due to lack of specificity in the question to provide a precise response. The Table 9 illustrates an ambiguous question. It shows how the model initially provides an incorrect response, and corrects it based on the human agent's input.

Table 8: Snippet from ShARC dataset: Incomplete Question. The highlighted span in question represents the source of Incompleteness.

Data sample	**Question**	Does this country have social security agreements with the UK?
	Context	Other countries with UK benefits arrangements. The following countries have social security agreements with the UK: Kosovo, Mauritius, Montenegro, and New Zealand.
Agent response	**LLM**	Which country are you referring to?
	Human Agent	Montenegro.
	LLM (Revised)	Yes.

Table 9: Snippet from ShARC dataset: Ambiguous Question. The highlighted span in context represents the source of ambiguity.

Data sample	**Question**	Can I get Housing Benefit?
	Context	If you're single and under 35, you can only get Housing Benefit for bed-sit accommodation or a single room in shared accommodation.
Agent response	**LLM**	Yes, if you're single and under 35, you can get Housing Benefit for bed-sit accommodation or a single room in shared accommodation.
	Human Agent	No, I am not single and under 35.
	LLM (Revised)	No.