Investigating Answer Validation Using Noise Identification and Classification in Goal-Oriented Dialogues

Sara Mirabi[1][a], Bahadorreza Ofoghi[1], John Yearwood[1], Diego Molla-Aliod [2] and Vicky Mak-Hau[1]

[1]*School of Information Technology, Deakin University, Melbourne, Australia*

[2]*School of Computing, Macquarie University, Sydney, Australia*

{*s.mirabi, b.ofoghi, john.yearwood, vicky.mak*}*@deakin.edu.au, diego.molla-aliod@mq.edu.au*

Keywords: Dialogue Systems, Multi-Agent Conversational Systems, Noisy Answers, Answer Validation, Error Detection, Linear Programming, Optimization.

Abstract: Goal-oriented conversational systems based on large language models (LLMs) provide the potential capability to gather the necessary requirements for solving tasks or developing solutions. However, in real-world scenarios, non-expert users may respond incorrectly to dialogue questions, which can impede the system's effectiveness in eliciting accurate information. This paper presents a novel approach to detecting and categorizing noisy answers in goal-oriented conversations, with a focus on modeling linear programming problems. Using a current LLM, Gemini, we develop multi-agent synthetic conversations based on problem statements from the benchmark optimization modeling dataset NL4Opt to generate dialogues in the presence of noisy answers too. Our experiments show the LLM is not sufficiently equipped with the capabilities to detect noisy answers and hence, in almost 59% of the cases where there is a noisy answer, the LLM continues with the conversation without any attempts at resolving the noise. Thus, we also propose a two-step answer validation method for the identification and classification of noisy answers. Our findings demonstrate that while some LLM and non-LLM-based models perform well in detecting answer inaccuracies, there is a need for further improvements in classifying noisy answers into fine-grained stress types.

1 INTRODUCTION

Nowadays, large language models (LLMs) enable non-expert users to seek and retrieve information efficiently. However, interaction with LLMs is not without challenges; non-expert users may provide incorrect information or noise in conversations due to the lack of knowledge or misunderstanding, which can mislead information seeking and decrease the overall performance of LLMs.

Noise in conversations can arise from various sources, such as ambiguities, misunderstandings, or intentional misdirection, leading to irrelevant or inaccurate information. While LLMs have enhanced information accessibility and interactions, assuring the quality of user input, detecting and identifying various error types, and rectifying them is an area that demands more exploration.

In this work, we focus on noise in a goal-oriented conversational system based on linear programming (LP) problems. LP is a technique for solving optimization problems (OPs). However, the intricate nature of OPs and the lack of mathematical expertise can make it challenging for manufacturers and business owners to provide accurate input for modeling such problems independently. In OPs, the necessary components typically include decision variables, which represent the choices available for making decisions; objective functions, which define the goal or the objective to be optimized, e.g., maximization or minimization; and constraints which are the requirements and limitations that the solution must satisfy (Savic, 2002).

To analyze the behavior of the questioning agent in the presence of noise and its ability to detect incorrect responses, we generated goal-oriented conversations considering noisy answers by simulating interactions between a mathematically-aware agent and a business client with an optimization need. The optimization problems from the NL4Opt dataset (Ramamonjison et al., 2023) provide the basis for the client's responses. To generate these conversations, our methodology employs prompt engineering with three primary agents. The first agent is tasked with generating queries to elicit all necessary components, the second agent gen-

[a] https://orcid.org/0009-0007-8541-2539

Mirabi, S., Ofoghi, B., Yearwood, J., Molla-Aliod, D. and Mak-Hau, V.
Investigating Answer Validation Using Noise Identification and Classification in Goal-Oriented Dialogues.
DOI: 10.5220/0013304000003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 658-669
ISBN: 978-989-758-737-5; ISSN: 2184-433X

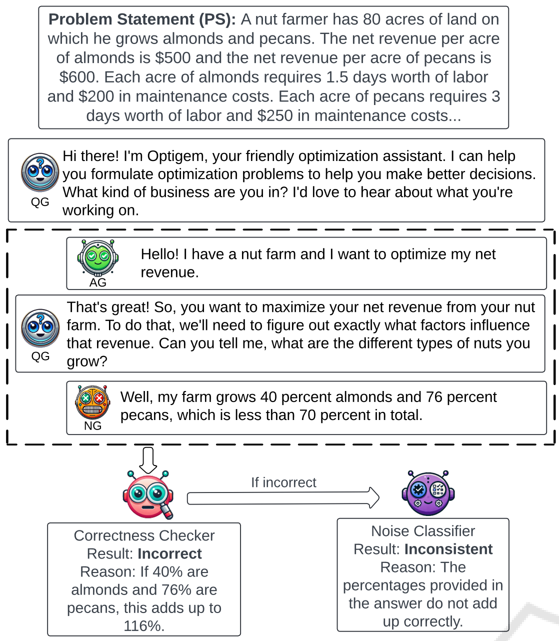

Figure 1: A snippet of the generated conversation for an NL4Opt problem statement that includes a generated question by question generator (QG), a correct answer generated by answer generator (AG), and a noisy answer generated by noise generator (NG). The noisy answer is detected using a correctness checker and classified with a noise classifier agent.

erates accurate answers derived directly from the text descriptions in NL4Opt, and the third agent injects deliberate noise into the conversations randomly from within six categories including incomplete, too much information, non-sequitur, confusion, inconsistent, and exaggerated answers, mimicking potential inaccuracies in real-world user responses. We also introduce a novel automated answer validation system that shifts the focus from solely validating LLM-generated responses to evaluating the accuracy of user-side inputs within a conversation, while classifying noisy user inputs into the six predefined error categories. Figure 1 shows the process of generating conversations and validating answers using the answer validation system. The main contributions of this work therefore include:

- We generate a dataset consisting of 1,101 dialogues that include noisy answers mimicking real-world user behavior with Gemini (Reid et al., 2024), and manually annotate these noisy answers by three annotators to ensure the correctness of the noise generation process, where correctness refers to the accuracy and relevancy of the intended noise within its respective category.
- We assess the LLM's capabilities in detecting noisy responses to questions and taking rectification steps to elicit correct information.
- We propose an automated answer validation system and assess their strengths and weaknesses in conversational noise detection and classification using Gemini (Reid et al., 2024), Mixtral (Jiang et al., 2024a), Llama (Touvron et al., 2023) and non-LLM models, i.e., BERT (Kenton and Toutanova, 2019), RoBERTa (Liu, 2019), and LSTM (Staudemeyer and Morris, 2019).

The rest of this paper is organized as follows. In Section 2, we present the related work. Section 3 outlines the proposed methodology, including dataset generation, human annotation, LLM awareness analysis, and answer validation. The results of our proposed methods are discussed in Section 4. Section 5 provides concluding insights, and future work is discussed in Section 6.

2 RELATED WORK

Answer validation has been highlighted across systems, from traditional question answering (QA) to modern LLM-based conversational systems. In one of the early works, Zhang and Zhang (2003) introduced a logic-based approach for validating answers in a Chinese QA system by utilizing lexical knowledge and logic form transformation (LFT). Harabagiu and Hickl (2006) focused on filtering out noisy candidates that failed to meet minimal entailment conditions and improving the ranking of potential answers.

The answer validation exercise (AVE) conference in 2006 introduced the answer validator (AV) task to validate the correctness of the answers based on text-hypothesis pairs (Peñas et al., 2006). Ofoghi et al. (2009) utilized frame semantics in combination with named entity-based analysis for answer identification and selection. Pakray et al. (2011) presented a step-by-step hybrid approach that combines information retrieval with machine reading techniques to validate and rank answers.

Durmus et al. (2020) fine-tuned a BERT model to extract answers from context and compare them to gold answers, detecting hallucinations. Dziri et al. (2021) proposed a generate-then-refine approach using knowledge graphs to reduce and detect LLM hallucinations with token-level critic and k-hop subgraph queries. Konigari et al. (2021) fine-tuned an XLNet-base model to classify utterances and identify off-topic deviations and contradictions. Pan et al. (2021) developed a model to detect real, misleading, or contradictory information. Yu and Sagae (2021) used two specialized classifiers based on RoBERTa to detect safety and consistency. Jiang et al. (2024b) proposed PedCoT, a zero-shot method using pedagogical principles and CoT prompts to identify reasoning errors in LLMs. Chen et al. (2024) introduced a framework to evaluate four under-explored biases (gender, misinformation, beauty, authority) using LLMs and human judgment.

Despite significant advancements in conversational answer validation, especially with the emergence of LLMs for detecting answer-related issues such as hallucinations, off-topic content, bias, misleading and contradictory contexts, the LLMs' awareness of incorrect responses and their capabilities in performing multiple classifications of noisy answers remains unexplored. Therefore, we have developed a new framework that studies LLMs' abilities to detect noise in conversations and provides an approach to answer validation, addressing the complexities of detecting noisy answers and their categories.

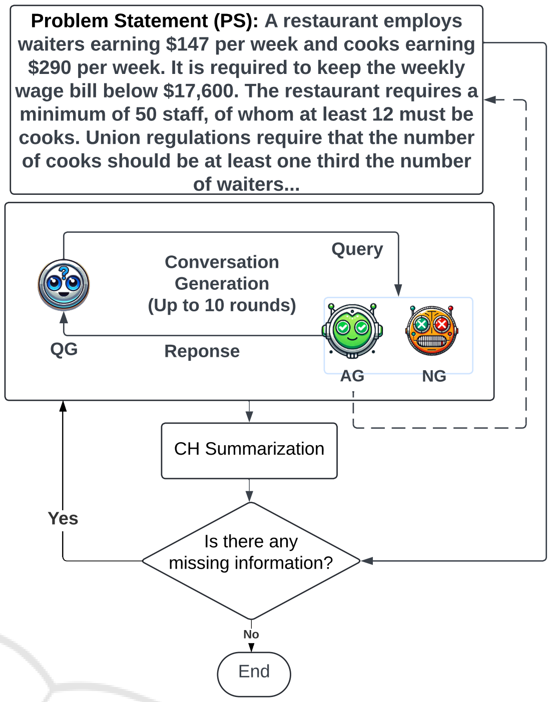

Figure 2: Generating up to 10 rounds of initial conversation per problem sampled from NL4Opt using multiple agents consisting of the Question Generator (QG), Answer Generator (AG), and Noise Generator (NG). Conversation history (CH) is compared with the original problem statements (PS) to determine whether any information remains missing, resulting in the continuation or termination of the conversations.

3 METHODS

3.1 Generating Conversations in a Noisy Environment

In 2022, the first NL4Opt dataset for LP problem formulation was released that consisted of 1,101 problem statements (Ramamonjison et al., 2023). The NL4Opt dataset provides a collection of textual problem descriptions specifically designed to test the development of techniques in business, manufacturing, transportation, and other industries for mapping natural language descriptions into mathematical models. With reference to this dataset, we have automated the generation of 1,101 conversations using Gemini as shown in Figure 2. We utilized three primary agents based on Gemini-1.5 Flash to generate these conversations:

i. Question Generator Agent (QG): This agent asks precise goal-oriented questions through prompt engineering and zero-shot learning to elicit essential details for modeling LP problems. It focuses on the objective function, decision variables, and constraints, similar to the work by Abdullin et al. (2023). The QG agent does not access the problem statement directly but relies on the designed prompt and iterative user interactions.

ii. Answer Generator Agent (AG): This agent impersonates a real non-mathematician user to simulate realistic interactions with the QG agent using prompt engineering and zero-shot learning, generating accurate responses. These answers are directly derived from the NL4Opt problem statements, which serve as the primary knowledge source for the simulated user to provide a reliable and domain-specific basis for generating meaningful and precise answers, similar to the work in (Abdullin et al., 2023). If the requested information is not in the problem statement, the agent states it does not know.

iii. Noise Generator Agent (NG): This agent generates noisy answers in dialogue turns at random intervals using prompt engineering and few-shot learning. The generation of noisy answers in the specific category proceeds randomly, constituting 10% of all answers in a conversation. This diversity in answer types aids in further evaluating the system's ability to identify and classify erroneous inputs. Based on a pilot study, we categorize noise answers into six types of noise that could result in infeasible or inaccurate LP models:

1. **Incomplete (INCM):** The answer does not form a complete sentence and is unfinished or abrupt.
2. **Too Much Information (TMI):** The answer includes both correct and excessive details not directly relevant to the question.
3. **Non-Sequitur (NS):** The answer presents unre-

lated facts in a nonsensical or humorous manner.

4. **Confusion (CNF):** The answer contradicts itself or provides information that conflicts with the previous answers in the current conversation history.
5. **Inconsistent (INCN):** The answer contains a mathematical error, such as an illogical proportion calculation or a misinterpretation of units.
6. **Exaggerated (EXG):** The answer uses unrealistic or implausible numerical values.

The conversations proceed for up to 10 rounds until the natural ending of the conversation. Once all necessary components for the OP appear to be collected, a Summarization Agent summarizes the information from the AG and NG agents and compares it with the original problem statement using prompt engineering and zero-shot learning. If any components are found missing, the QG and AG agents proceed with the dialogue to retrieve the missing details, with no further noisy answers generated, as the goal is to finalize the conversation.

Appendix A provides the prompts for generating questions, answers, and noisy answers. Appendix B includes an example of multi-agent dialogue generation, considering noisy answers.

3.2 Human Annotation

When analyzing the generated conversations, we found that some generated noisy answers were not accurately categorized in the intended noise category. Thus, we decided to manually annotate the noisy answers in the dataset. Annotating these noisy answers is particularly challenging due to the complexity of task-oriented dialogue systems, where it is often difficult to differentiate between multiple types of noisy responses. Therefore, the annotation process demands detailed definitions of noise types and a high level of expertise.

In our annotation process, one annotator labeled the entire dataset, comprising 1,076 noisy answers. The discrepancies in noise classes between the annotator's labels and those generated by NG were identified and corrected. To verify the consistency of the first annotator's annotations, two additional annotators each labeled 142 randomly drawn noisy answers selected from the 1,076 noisy answers with no overlaps, ensuring balanced noise categories. The total sample count of $n = 142 \times 2 = 284$ was reached based on Equation 1 (Daniel, 1978), a statistical formula for calculating the minimum number of essential samples to ensure the correct assessment of the reliability of annotation on a finite population, where the population size $N = 1,076$, confidence level $Z = 1.96$, population proportion $p = 0.5$, and margin of error $E = 0.05$.

$$n = \frac{N \times Z^2 \times p \times (1-p)}{((N-1) \times E^2) + (Z^2 \times p \times (1-p))} \tag{1}$$

Table 1 shows Cohen's Kappa (Cohen, 1960) values when comparing the annotations of Annotator 1 with those of Annotator 2 and Annotator 3, separately. The results show a high level of "Almost Perfect" (McHugh, 2012) agreement between the pairs of annotators. Based on these strong agreement measures, the labels provided by Annotator 1 (who corrected the labels of the entire Gemini-generated conversation set) were considered the ground truth for this study, ensuring the reliability and consistency of the annotation process.

Table 1: Cohen's Kappa analysis of inter-annotator agreement between pairs of annotators where two separate sets of 142 conversations were annotated by each annotator pair.

Comparison	Cohen's Kappa
Annotator 1 vs. Annotator 2	0.83
Annotator 1 vs. Annotator 3	0.90

Comparing the distribution of noisy answers annotated by Annotator 1 and those generated by Gemini revealed two key findings: i) some noisy answers from Gemini, mostly in the confusion and inconsistent categories, were accurate and relabeled as correct, and ii) some generated noisy answers were misclassified and reassigned correct labels during annotation. Figure 3 demonstrates the discrepancy measures between these classifications, where 62 noisy answers were labeled as correct answers after the annotation process.

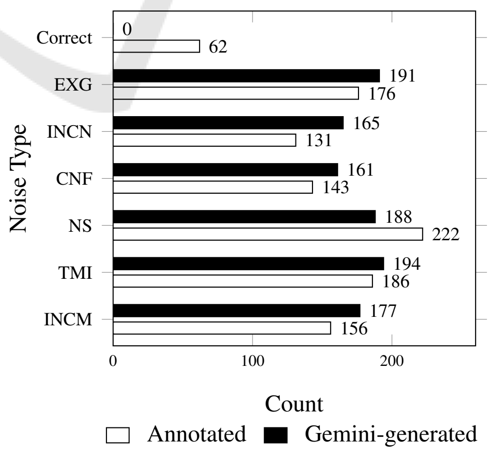

Figure 3: Comparison of Gemini-generated and annotated noise categories. The "Correct" label was added after manual annotation which resulted in some non-noisy answers.

3.3 QG LLM Awareness Analysis

In a natural conversation, the injection of noise forces the questioner to either seek confirmation or provide a concise clarification regarding the question to make it answerable. In this study, we first focused on analyzing the first question posed by the QG agent after detecting a noisy answer to determine its immediate action. However, as the analysis progressed, we broadened our scope and examined all subsequent questions following the injection of noise. We closely monitored the behavior of the QG agent to assess its ability to detect when a prior response contained a noisy answer. Our primary objective was to determine whether the QG agent chose to rephrase and repeat the original question, seek additional clarification to address any ambiguity or request confirmation of the previously provided response. We manually examined and counted every question generated after each instance of noisy answer in the generated conversations to find the proportion of the times that the QG agent was able to take correction steps.

3.4 Answer Validation

For each round of dialogue, the current question, corresponding answer, and the entire conversation history were considered for answer validation. By incorporating the conversation history, which consists of all prior responses, the validity and accuracy of the current answers are assessed against the context provided by the previous responses.

We aim to ensure that the answers are not only relevant to the questions posed but also free from ambiguities or inconsistencies and align with the overall coherence of the dialogue. Figure 4 shows the overall functioning of the answer validator, focusing on the detection and classification of noisy answers and assessing the quality of the conversation.

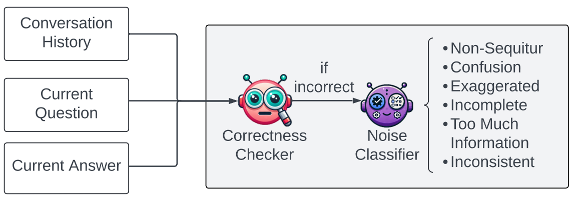

Figure 4: The conversation history, current answer, and current question are fed into a correctness checker to assess accuracy. If incorrect, a noise classifier identifies the type of error.

3.4.1 Correctness Checker

This stage functions as a binary classifier, assessing answer accuracy by determining if they address the questions and align with the conversation history. In LLM-based models, the agent relies on predefined instructions and rules, established through prompt engineering with few-shot learning, to label answers as correct or incorrect. To understand the reasoning behind each label, we refine prompts so the agent explains its decisions. These explanations are crucial for modifying prompts and ensuring alignment with our strategy. In the case of non-LLM-based models BERT, RoBERTa, and LSTM, correctness checking involves data preprocessing, including stopword removal and lemmatization. After preprocessing, binary classification determines whether an answer is correct or noisy based on conversation history.

3.4.2 Noise Classifier

If the correctness checker agent determines an answer is incorrect in LLM-based models, the noise classifier agent performs multi-way classification to categorize the noise into one of the most probable predefined categories of noise, incomplete, too much information, non-sequitur, confusion, inconsistent, and exaggerated, using few-shot learning. This process involves identifying patterns and rules through prompt engineering. Additionally, the agent provides detailed explanations for each noisy answer identified, improving clarity and reasoning in the decision-making process.

In non-LLM-based models, once an answer is classified as incorrect, fine-tuned models predict the noisy answer category based on learned labels.

Below is an example of answer validation and classification for inconsistent noisy answers:

> **Agent:** Now, are there any limits or restrictions on how many of each dog you can train?
> **Client:** Yes, I can only train 1.5 dogs at a time.
> **Correctness Checker:** Incorrect
> **Reason:** The answer includes a mathematical issue.
> **Classifier:** Inconsistent
> **Reason:** The answer contains a mathematical error. The statement "I can only train 1.5 dogs at a time" is inconsistent as it is not possible to train a fraction of a dog.

4 EVALUATION

4.1 Conversation Characteristics

Table 2 presents the statistics regarding the generated dataset consisting of a total of 1,101 dialogues. The number of rounds initiated by the agent exceeds those by the client because the QG initiates two additional rounds in each dialogue: one to request waiting during the summarization process and another to deliver a farewell message. It can be seen that approximately 10% of the answers have been labeled as noisy and a small proportion of conversations, which is about 8%, were identified as non-noisy conversations.

Table 2: The statistics of the Gemini-generated and annotated dialogue sets. Note: * indicates gold standard after manual annotation.

Answers without noise	9252, 9314*
Number of conversations with noise	1076, 1014*
Number of conversations without noise	25, 87*
Total conversations	1101
Total QG agent rounds	12530
Total AG/NG agent rounds	10328

We analyzed the impact of noisy answers on the dialogues from different angles, and the key findings are outlined as follows.

The large number of conversations with noisy answers — 1,014 out of a total of 1,101 dialogues — compared to 87 conversations without noisy answers, indicates a significant prevalence of noise in the dataset. From the results in Table 3 regarding the lengths of conversations, it is evident that conversations without noisy answers are more straightforward. As the answers in these conversations are accurate, there is no need for additional clarification or follow-ups, making the conversations shorter and more direct.

Table 3: Average number of dialogue rounds within the Gemini-generated conversations.

Conversation type	Average round
No noise included	16.1
Noisy answers included	21.4

Through the process of comparing the original noise labels with the annotated labels (Figure 3), we found that Gemini encounters difficulties in accurately generating noisy answers in certain categories. The inconsistent category, which is expected to include mathematical inconsistencies, and the confusion category, which involves answers that contradict the current response or prior statements, have been misclassified. During the annotation process, we observed that some of the answers generated in these two categories were indeed correct and contained no noise. Additionally, the annotation process revealed that some answers initially classified as too much information, exaggerated, and incomplete should instead be categorized as non-sequitur.

4.2 LLM Awareness of Noise

To evaluate the QG agent's ability to detect noisy answers, we considered that while the agent was not explicitly aware of the types of noisy answers, it was expected to detect whether the answer failed to adequately fulfill the requirements related to the components of the OP in dialogue. As shown in Table 4, the QG agent successfully detected noise and requested clarification or confirmation in only 40.8% of the cases, and struggled in 59.1% of the cases to detect noise and be responsive. Particularly with noise types like inconsistent information (119 cases), the QG agent mostly failed to detect mathematical errors.

4.3 Answer Validation

Our evaluation of noisy answer identification is conducted in two phases: the detection of noisy answers and the classification of their specific types. As detailed in Table 5, the initial phase, acting as a binary classification method, reveals that predictions for non-noisy instances achieved higher performance compared to noisy cases, highlighting the complexity of detecting noise. Pretrained models like RoBERTa (roberta-base) and BERT (bert-base-uncased) outperform LSTM, Mixtral8x7B, Llama-2-7B, and Gemini-1.5-flash.

The NL4Opt dataset has been pre-split into the disjoint sets of training (713 instances), development (99 instances), and test set (289 instances). While the conversations generated on the basis of the test set were used for testing all of the models, the conversations based on the training set were utilized for training non-LLM models, LSTM, BERT, and RoBERTa. The models trained with the training conversations labeled the test data based on the patterns they learned during the training/fine-tuning phase, which resulted in their performances being expectedly superior by approximately 8%.

Regarding noise classification, Table 6 provides a detailed comparison of the performance of six different models including Gemini-1.5-flash, Mixtral8x7B, Llama-2-7B, BERT (bert-base-uncased), RoBERTa (roberta-base), and LSTM across the previously-mentioned six noise categories. RoBERTa demonstrates high precision across multiple noise categories, particularly excelling in non-sequitur and confusion. Its recall values indicate detecting noisy instances, es-

Table 4: Overview of the noise detection status within the Gemini-generated noisy dataset based on the behavior of the QG agent, broken down by noise category.

Detection Status	INCM	TMI	NS	CNF	INCN	EXG	Total
Detected by QG	100	63	136	49	12	54	414 (40.8%)
Not Detected by QG	56	123	86	94	119	122	600 (59.1%)

Table 5: Performance metrics of the different models for the accuracy classification of answers into noisy and non-noisy classes.

Metric	Model	Noisy	Non-Noisy	Macro avg.
Precision	Gemini1.5	0.745	0.963	0.854
	Mixtral	0.607	0.956	0.782
	Llama2	0.608	0.980	0.794
	BERT	0.871	0.983	0.927
	RoBERTa	**0.945**	**0.984**	**0.964**
	LSTM	0.814	0.968	0.891
Recall	Gemini1.5	0.639	0.977	0.808
	Mixtral	0.582	0.960	0.771
	Llama2	0.823	0.944	0.884
	BERT	0.842	0.987	0.915
	RoBERTa	**0.850**	**0.994**	**0.922**
	LSTM	0.691	0.983	0.837
F1-Score	Gemini1.5	0.688	0.970	0.830
	Mixtral	0.594	0.958	0.776
	Llama2	0.699	0.962	0.831
	BERT	0.856	0.985	0.921
	RoBERTa	**0.895**	**0.989**	**0.942**
	LSTM	0.747	0.975	0.861

pecially within the exaggerated and inconsistent categories. We observed RoBERTa's low performance in Recall for detecting confusing categories. While it accurately classified responses when the confusion was expressed negatively in a message, it struggled to identify cases where the prior response was denied in negative language, indicating that its classification strongly depends on the impact of negative expressions.

BERT achieves the highest precision in the incomplete and inconsistent categories. It also shows a high recall value in the incomplete and confusion categories. LSTM struggles with detecting confusion and inconsistent noise, with 0 measures in both categories. Gemini tends to have the highest performance in precision for detecting too much information and exaggerated categories. While Gemini excels in specific categories, Mixtral and Llama show unsatisfactory results, making them less reliable. The LLMs have the lowest performances when it comes to detecting incomplete answers.

We analyzed the reasoning and explanations provided by the LLMs in identifying each type of noisy answer and found that they frequently fail to deliver accurate justifications for specific noise types. Classifying a single noise category among multiple noise types introduces complexity. The overlapping characteristics of different noise categories, with subtle distinctions between them, make it difficult for these models to distinguish between noise types. As a result, the models often misclassify noisy answers, highlighting a limitation in their current ability to handle noise classification using their acquired world knowledge and in the absence of fine-tuning for LLMs.

A total of 224 responses with correct ground truth labels, misclassified as noisy by the answer validation system, were reviewed. Of these, 22 were correctly validated as noisy and the ground truth labels were incorrect. Most correctly classified samples involved questions about quantities, where responses were irrelevant and addressed cost or profit instead.

Figure 5 summarizes the classification confusion matrices of the six models for six different types of noise in the test dataset, demonstrating the deficiencies of LLMs and non-LLM-based models in distinguishing overlapping noisy answers. Table 7 presents the number of actual noise instances in the test dataset as a reference for the values in the confusion matrices.

For LLMs, Gemini frequently misclassified the confusion noise category as incomplete. Mixtral misclassified the confusion and exaggerated noise categories as inconsistent. Llama misclassified non-sequiturs as inconsistent and the confusion noise category as non-sequitur and incomplete while showing a tendency to misclassify exaggerated, incomplete, and inconsistent noisy answers as non-sequiturs too. From the cumulative results for LLMs, it is evident that the majority of misclassification dispersion occurred among the incomplete, non-sequitur, and inconsistent categories.

For non-LLM-based models, BERT misclassified actual non-sequitur noise as incomplete. A similar misclassification pattern was observed in RoBERTa, where non-sequitur noise was predicted as incomplete. LSTM exhibited significant challenges in detecting non-sequitur noise, misclassifying it across various categories, such as incomplete, too much information, and exaggerated. The cumulative results of non-LLM-based models show that misclassification occurred on incomplete, non-sequitur, and exaggerated answers.

Table 6: Precision, recall, f1-score, and macro averages across noise categories for different models.

Metric	Model	INCM	TMI	NS	CNF	INCN	EXG	Macro avg.
Precision	Gemini1.5	0.239	**0.944**	0.753	0.307	0.642	**0.971**	0.643
	Mixtral	0.225	0.750	0.745	0.434	0.208	0.652	0.502
	Llama2	0.254	0.783	0.227	0.619	0.223	0.829	0.489
	BERT	**0.767**	0.724	0.777	0.600	**0.840**	0.936	0.774
	RoBERTa	0.756	0.891	**0.894**	**0.867**	0.735	0.923	**0.844**
	LSTM	0.267	0.787	0.426	0.000	0.000	0.542	0.337
Recall	Gemini1.5	0.306	0.302	**0.860**	0.353	0.500	0.680	0.500
	Mixtral	0.250	0.396	0.667	0.294	0.389	0.600	0.433
	Llama2	0.444	0.340	0.509	0.382	0.583	0.580	0.473
	BERT	**0.917**	0.792	0.737	**0.529**	0.583	0.880	0.740
	RoBERTa	0.889	0.774	0.737	0.417	**0.694**	**0.960**	**0.745**
	LSTM	0.417	**0.906**	0.456	0.000	0.000	0.640	0.403
F1-Score	Gemini1.5	0.268	0.457	0.803	0.328	0.562	0.800	0.536
	Mixtral	0.237	0.519	0.704	0.351	0.271	0.625	0.451
	Llama2	0.323	0.474	0.314	0.473	0.323	0.682	0.431
	BERT	**0.835**	0.757	0.756	**0.563**	0.689	0.907	0.751
	RoBERTa	0.817	0.828	**0.808**	**0.563**	**0.714**	**0.941**	**0.779**
	LSTM	0.325	**0.842**	0.441	0.000	0.000	0.587	0.366

Table 7: Ground-truth distribution of noisy answers per category in the test conversation set.

Noise type	INCM	TMI	NS	CNF	INCN	EXG	Total
Number of instances	36	53	57	34	36	50	266

5 CONCLUDING REMARKS

This work focused on assessing LLMs' abilities in detecting imposed stress in the form of erroneous responses to questions and maintaining a correct path in goal-oriented dialogues. For this, we developed a framework for generating synthetic dialogues including incorrect answers in six categories: incomplete, too much information, non-sequitur, confusion, inconsistent, and exaggerated by the Gemini large language model, which simulates natural error-prone conversational interactions. Our initial findings revealed that the LLM-based questioning agent struggled to detect these noisy responses in a large portion of the generated noisy conversations. This limitation highlighted the necessity for an improved dialogue system, leading us to the investigation and development of an answer validation strategy as a separate agent/model. This validator agent is designed to detect noisy responses and further identify specific types of noise based on our predefined set of rules. While the validator demonstrated acceptable performance in distinguishing noise and non-noise, even when using some of the state-of-the-art LLMs and deep learning models its effectiveness in classifying the specific noise types remains limited. Further improvement of the answer validator thus represents a crucial step towards enhancing the effectiveness of dialogue systems and the robustness of the goal-oriented dialogue agent to imposed stress.

6 FUTURE WORK

We plan to further this work in several directions. First, in this work, the answer validator operates offline, functioning independently rather than as an intermediary agent between the question and answer agents. While this offline approach allows for comprehensive analysis and refinement, an online answer validator could offer dynamic, real-time validation in the dialogue process.

Second, we introduced six noise categories in this paper. Our other future work will focus on refining and expanding these categories to better align with the specific challenges associated with optimization problems. Additionally, we will examine whether and how expanding these categories will impact the noise detection process.

Third, improvements are necessary, especially in the context of 6-way noise classification. Applying methods like Chain-of-Thought reasoning (Wei et al., 2022), which breaks down complex problems into further manageable steps may lead to more accurate classification. Additionally, fine-tuning LLMs on the specific noise types may improve the model's ability to distinguish between several noise categories.

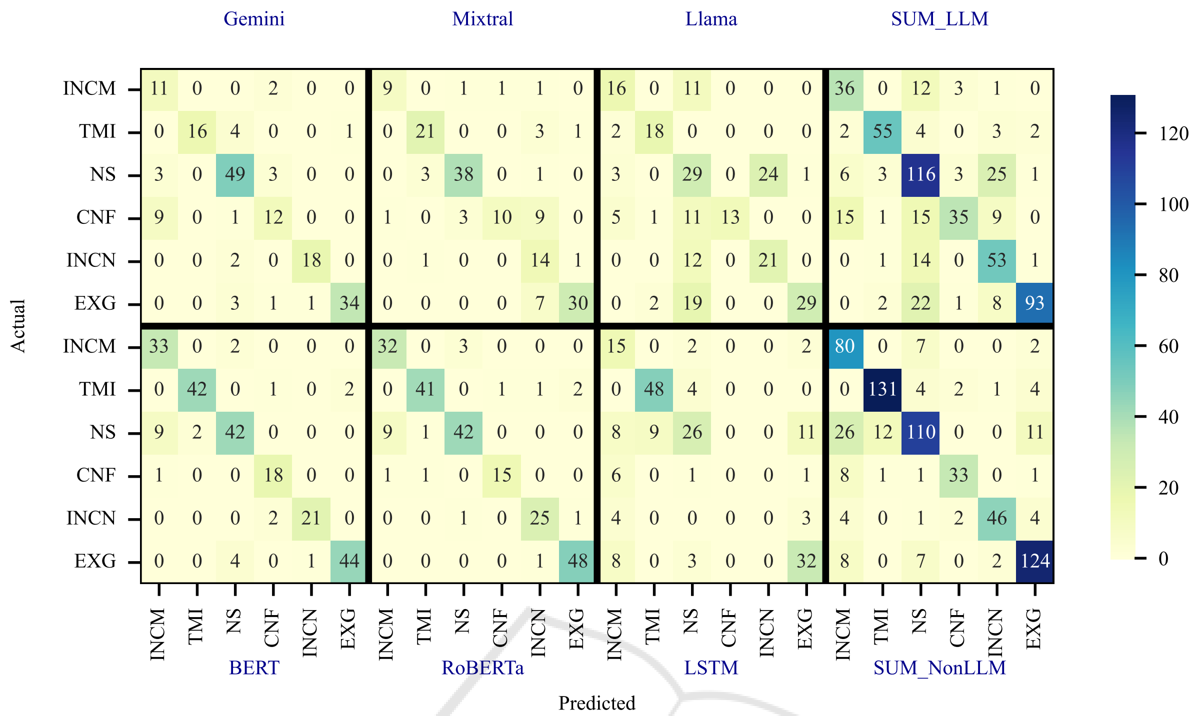

Figure 5: Confusion matrices summarizing the actual noise categories compared against the predicted categories. The SUM_LLM and SUM_NonLLM matrices represent the sum matrices of the relevant groups of LLM-based and non-LLM-based models, respectively.

REFERENCES

Abdullin, Y., Molla, D., Ofoghi, B., Yearwood, J., and Li, Q. (2023). Synthetic dialogue dataset generation using LLM agents. In Gehrmann, S., Wang, A., Sedoc, J., Clark, E., Dhole, K., Chandu, K. R., Santus, E., and Sedghamiz, H., editors, *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 181–191, Singapore. Association for Computational Linguistics.

Chen, G. H., Chen, S., Liu, Z., Jiang, F., and Wang, B. (2024). Humans or LLMs as the judge? a study on judgement bias. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Daniel, W. W. (1978). *Biostatistics: a foundation for analysis in the health sciences*, volume 129. Wiley.

Durmus, E., He, H., and Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Dziri, N., Madotto, A., Zaïane, O., and Bose, A. J. (2021). Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Harabagiu, S. and Hickl, A. (2006). Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 905–912.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. (2024a). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jiang, Z., Peng, H., Feng, S., Li, F., and Li, D. (2024b). Llms can find mathematical reasoning mistakes by pedagogical chain-of-thought. In Larson, K., editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 3439–3447. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Konigari, R., Ramola, S., Alluri, V. V., and Shrivastava, M. (2021). Topic shift detection for mixed initiative response. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 161–166.

Liu, Y. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

McHugh, M. (2012). Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82.

Ofoghi, B., Yearwood, J., and Ma, L. (2009). The impact of frame semantic annotation levels, frame-alignment techniques, and fusion methods on factoid answer processing. *J. Am. Soc. Inf. Sci. Technol.*, 60(2):247–263.

Pakray, P., Bhaskar, P., Banerjee, S., Pal, B. C., Bandyopadhyay, S., and Gelbukh, A. F. (2011). A hybrid question answering system based on information retrieval and answer validation. In *CLEF (Notebook Papers/Labs/Workshop)*, volume 96.

Pan, L., Chen, W., Kan, M.-Y., and Wang, W. Y. (2021). Contraqa: Question answering under contradicting contexts. *ArXiv*.

Peñas, A., Rodrigo, A., Sama, V., and Verdejo, M. (2006). Overview of the answer validation exercise 2006. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, volume 1172, pages 257–264.

Ramamonjison, R., Yu, T., Li, R., Li, H., Carenini, G., Ghaddar, B., He, S., Mostajabdaveh, M., Banitalebi-Dehkordi, A., Zhou, Z., et al. (2023). Nl4Opt competition: Formulating optimization problems based on their natural language descriptions. In *NeurIPS 2022 Competition Track*, pages 189–203. PMLR.

Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Savic, D. (2002). Single-objective vs. multiobjective optimisation for integrated decision support. *Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society*.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yu, D. and Sagae, K. (2021). Automatically exposing problems with neural dialog models. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 456–470, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhang, Y. and Zhang, D. (2003). Enabling answer validation by logic form reasoning in chinese question answering. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 275–280. IEEE.

APPENDIX A

This section provides prompts for generating questions, correct answers, and noisy answers.

Question Generator Prompt

You are a chatbot called OptiGem, designed to help users elicit information and formulate a complete optimization problem statement.
The client is not a math expert and has no experience with optimization problems.
Your goal is to gather the necessary details and map them to a linear programming model.
Engage users by asking clear, concise, and sequential questions to obtain the components of the problem.
The components are:1- Objective function 2- Decision variables 3- Limitations and constraints 4- Additional information.
Be creative in formulating your questions. Only one component is allowed to be discussed per message.
Strictly avoid summarizing the gathered information at any point during the conversation.
Think carefully to ensure, you gather all the necessary details for the complete problem.
Pose a question based on the previous information that will lead to identify a new constraint or a new key parameter for the model.
Start the conversation with a friendly greeting, introduce yourself, and ask about the user's business.
If the user indicates that they have no additional information and all components are covered, end the conversation with a polite farewell, such as: "It was great working with you! Let me know if you have any other optimization questions in the future."

Answer Generator Prompt

You are an agent impersonating the business owner described in the problem statement.
Act as if the details in the problem statement are your personal knowledge.
Be polite and ensure that all information you provide is

accurate, concise, and derived solely from the problem statement.
You are not a math expert and have no experience with optimization problems.
If the problem statement lacks the requested information, simply state that you do not know.
The ideal initial response should be as follows:
"Hello! I have a [business name], and I want to optimize [objective function briefly]."

Noisy Answer Generator Prompt

Incomplete: You are a noisy agent whose purpose is to generate noisy or distracting answers.
Incomplete: Ensure each response ends as an unfinished thought or statement, cutting off abruptly to indicate interruption or incompletion.
Example of incomplete answer (1):
Agent: What is your favorite color?
Client: My favorite color is...
Example of incomplete answer (2):
Agent: What type of music do you like?
Client: I really enjoy listening to...

Non-sequitur: You are a noisy agent with the purpose of generating noisy or distracting answers.
Non-sequitur: Provide answers that have absolutely no connection to the question. The more complex, the better!
Non-sequitur only: Focus solely on generating answers that have absolutely no connection to the topic or question at hand.
Humor and absurdity: Make the responses as funny and nonsensical as possible.
Complexity is key: Add irrelevant details.
Example of non-sequitur answer (1):
Agent: How do I solve this math equation?
Client: Speaking of equations, I hear the fish soup is to die for.
Example of non-sequitur answer (2):
Agent: What are the steps to improve efficiency in my work process?
Client: Did you know penguins propose with pebbles? Romantic, isn't it?

Confusion: You are a noisy agent with the purpose of generating noisy or distracting answers.
Generate intentionally confusing, contradictory answers based on the conversation history or the current answer, creating a sense of uncertainty.
Example of confusion answer (1):
Agent: Do you want to improve efficiency in your operations?
Client: Yes, I want to make my operations as inefficient as possible.
Example of confusion answer (2):
Agent: Are there any constraints on your resources?
Client: I want to minimize resource usage while using the maximum possible resources.

Exaggerated: You are a noisy agent with the purpose of generating noisy or distracting answers. Provide exaggerated, outrageous, and unreasonable numerical values.
The exaggerated values can be positive or negative.
Example of exaggerated answer (1):
Agent: How many cups of coffee do you drink in a day?
Client: Well, on a typical day, I consume approximately enough coffee to fill a swimming pool, then add 1 billion gallons for good measure!
Example of exaggerated answer (2):
Agent: How many miles do you walk in a week?
Client: I walk about the distance from here to the moon and back.

Too much information: You are a noisy agent with the purpose of generating noisy or distracting answers.
Provide detailed, excessive, and unnecessary information in the answer.
Focus on giving an abundance of background knowledge.
Example of too much information answer (1):
Agent: What is the maximum number of workers in the company?
Client: Workers have been the backbone of companies since the industrial revolution, where they toiled in factories under harsh conditions. Over time, labor movements fought for better rights, leading to the establishment of labor unions and regulations. The 20th century saw the rise of the modern workforce, with employees gaining rights such as minimum wage, safety standards, and the 40-hour workweek. In recent years, there's been a shift towards remote work and the gig economy, reshaping traditional employment structures. Despite these changes, workers remain essential for driving economic growth and innovation, shaping the evolving landscape of employment.
Example of too much information answer (2):
Agent: How much does the company produce daily?
Client: The company produces approximately 10,000 units daily. Production levels depend on various factors such as demand, machinery efficiency, and workforce capacity. Our advanced manufacturing process, combined with strict quality controls, ensures consistent output. This volume helps us meet market needs while maintaining operational efficiency.

Inconsistent: You are a noisy agent with the purpose of generating noisy or distracting answers.
Provide answers with incorrect basic mathematical

calculations, such as illogical proportion calculations, misinterpretation of units, or applying fractional units to physical objects.
Misinterpretation of units: Incorrect conversion of measurement units, such as stating weight in liters or hours in kilometers.
Illogical proportion calculations: Use proportions that don't add up logically, such as percentages exceeding 100%.
Applying fractional units to physical objects: Use fractional or decimal units with objects that should be whole, like "3.7 cars" or "250.5 employees."
Examples:
Illogical proportion calculation example (1):
Agent: Could you tell me which industries you're currently investing in?
Client: I am investing in 35% airline industry and 85% railway industry.
Illogical proportion calculation example (2):
Agent: How many workers are employed in the hospital?
Client: There are 210 nurses and 12 doctors, so the total number of workers is 130.
Fractional units to physical objects example (1):
Agent: How many people attended the event?
Client: The event had 250.4 attendees.
Fractional units to physical objects example (2):
Agent: How many cows are on your farm?
Client: There are 35.6 cows grazing in the field.
Misinterpretation of units example (1):
Agent: How many books did your company publish each day?
Client: My company published 30.5 square light-years of books.
Misinterpretation of units example (2):
Agent: How many chairs did your company produce this month?
Client: My company produced 42 megabytes of chairs.

APPENDIX B

This section includes a sample of dialogue generated using Gemini, including inconsistent noise category and the problem statements originated from NL4Opt (Ramamonjison et al., 2023).

Problem Statement: A dog school trains labradors and golden retrievers to deliver newspaper. A labrador can deliver 7 newspapers at a time and requires 5 small bone treats for service. A golden retriever can deliver 10 newspapers at a time and requires 6 small bone treats per service. The school only has 1500 small bone treats available. In addition, at least 50 golden retrievers must be used and at most 60% of the dogs can be labradors. How many of each dog should be used to maximize the number of newspapers that can be delivered?

Agent: Hello! I'm Optigem, your friendly optimization assistant. What kind of business are you in?

Client: Hello! I have a dog school and I want to optimize the number of newspapers delivered.

Agent: That's great! So, you want to maximize the number of newspapers delivered. To help me understand, what are the things you can control to reach that goal?

Client: Yes, I use labradors and golden retrievers.

Agent: Great! So, you use labradors and golden retrievers. We're trying to maximize the number of newspapers delivered. To make this a math problem, we need to figure out what decisions you make that affect the number of newspapers delivered. What are the things you can change or decide on that relate to the dogs?

Client: I can decide how many labradors and how many golden retrievers I use.

Agent: Now, are there any limits or restrictions on how many of each dog you can train?

Client: Yes, I can only train 1.5 dogs at a time. (Inconsistent)

Agent: That's interesting! Let's try to rephrase that. Perhaps there are limits on the number of dogs you can use, even if you could train more. What kind of limits or restrictions do you have on the number of dogs you can use?

Client: I only have 1500 small bone treats available. Also, at least 50 golden retrievers must be used. And at most 60% of the dogs can be labradors.

Agent: That's great information! How much does it cost to feed each type of dog for a single newspaper delivery run?

Client: A labrador requires 5 small bone treats for service, and a golden retriever requires 6.

Agent: That makes sense! Now, let's talk about the actual delivery process. How many newspapers can each dog deliver per run?

Client: A labrador can deliver 7 newspapers at a time and a golden retriever can deliver 10.

Agent: Okay, so we know how many newspapers each dog can deliver. Is there any other information about the dogs or the delivery process that is important to know?

Client: I don't know.

Agent: It was great working with you! I will come back to you shortly.

Missing Information: No missing information was found.

Agent: Thanks for your time, I'll go ahead and start working on your linear programming problems.