Evaluating Robustness in LLM-based Medical Chatbots

Mukul Kumar
Wadhwani Institute for Artificial Intelligence
New Delhi, Delhi, India
mukul@wadhwaniai.org

Alugubelli Dinesh Reddy
Wadhwani Institute for Artificial Intelligence
New Delhi, Delhi, India
alugubelli@wadhwaniai.org

Sai Nikhilesh Reddy
Wadhwani Institute for Artificial Intelligence
New Delhi, Delhi, India
sai@wadhwaniai.org

Abstract

Large Language Models (LLMs) have demonstrated exceptional capabilities in understanding and generating human-like text, leading to their integration into various sectors, including agriculture and healthcare. There is growing interest in leveraging LLMs in Retrieval Augmented Generation (RAG) systems for medical chatbots to assist frontline healthcare workers, especially in resource-constrained settings like Indian primary healthcare. Building such systems hold great potential however, deploying LLM based solutions in such critical contexts demands rigorous evaluation of their robustness to noisy and diverse inputs inherent to real-world scenarios. This paper presents a way to stress-test and evaluate the robustness of LLMs in RAG-based medical chatbots for Indian primary healthcare. We assess their ability to handle noisy inputs in a question-answering task by introducing character-, word-, and sentence-level perturbations. Additionally, we evaluate the RAG system's ability to abstain from answering out-of-domain queries and analyze the impact of Automatic Speech Recognition (ASR)-generated transcriptions—commonly used to enhance accessibility for digitally low-skilled users—on chatbot performance.

Keywords

Large Language Models, Robustness, Evaluation, Health Chatbots, Indian Primary Healthcare, Retrieval Augmented Generation

1 Introduction

Large Language Models (LLMs) [12, 23] have demonstrated remarkable capabilities in understanding and generating human-like text, including sophisticated question-answering (QA) abilities [21, 25]. These strengths make them highly suitable for conversational applications like chatbots, leading to increasing interest in their potential to assist frontline workers in healthcare. However, deploying standard LLMs directly in such critical domains presents significant challenges. These models can be prone to hallucinations (generating plausible but factually incorrect outputs) [28], their knowledge is inherently limited by static training datasets with fixed cut-off dates [22, 30], they lack access to real-time or local contextual knowledge, and their grasp of specialized domains like healthcare may be incomplete due to the composition of their vast but general pre-training data [41]. Overcoming these limitations is crucial for building reliable and trustworthy healthcare applications.

Retrieval Augmented Generation (RAG) [26] is a technique used to address these limitations by augmenting an LLM with external knowledge retrieved from a dedicated knowledge database. By integrating real-time, domain-specific information into the generation process, RAG can significantly reduce hallucinations and enhance the LLM's expertise in specialized areas, such as healthcare. This approach has gained widespread adoption and is now powering several production-level LLM applications, including Perplexity [1] and ChatGPT Search [2], which leverage RAG to deliver more accurate, context-aware, and up-to-date responses.

Accredited Social Health Activists (ASHAs) and Anganwadi workers are frontline healthcare workers in India who play a crucial role in community health, especially for women and children. RAG-based chatbots can significantly support these workers [20, 36] by providing reliable, context-aware answers to queries related to maternal and child health, among other tasks. However, given the critical nature of healthcare, it is essential to rigorously test these chatbots for robustness before deployment. Real-world use presents challenges such as user typos in text inputs and errors from Automatic Speech Recognition (ASR) models in voice interactions, as voice technologies remain limited to a few resource-rich languages [18].

To evaluate the impact of real-world challenges on RAG-based chatbots in Indian primary healthcare settings, we designed experiments to simulate various input issues like typos, asking queries in different ways, etc. We also assessed the chatbot's robustness in abstaining from answering out-of-domain queries. Additionally, we incorporated Automatic Speech Recognition (ASR) transcriptions to analyze how imperfect voice-to-text conversions affect the accuracy and reliability of responses.

2 Related Work

Healthcare chatbots powered by Large Language Models (LLMs) have gained widespread adoption, particularly in India, where they assist patients and frontline health workers, such as Accredited Social Health Activists (ASHAs) and Anganwadi workers, in addressing diverse medical queries [34]. These chatbots leverage Retrieval-Augmented Generation (RAG) techniques [16, 26] to improve response accuracy by relying on trusted medical sources, mitigating the limitations of static training data. Despite evaluations of LLM-based medical chatbots in Indian contexts [16], there has been little research on their robustness to user input errors and Automatic Speech Recognition (ASR) inaccuracies in primary healthcare settings.

Robustness evaluations of LLMs have become a critical area of research, especially in high-stakes domains like healthcare [32]. Prior studies have explored various error types, including adversarial inputs, out-of-distribution data, and noisy environments [11]. However, for LLMs to be effective in healthcare, they must not only

2nd HEAL Workshop at CHI Conference on Human Factors in Computing Systems, Yokohama, Japan
2025.

process general language but also handle complex medical terminology [17] and domain-specific nuances [31]. Research has shown that LLMs struggle with medical jargon [32], text perturbations [14, 15], and context-specific retrieval challenges [39, 43, 44], raising concerns about their reliability. The integration of ASR further complicates this issue, as misrecognized medical terms can degrade chatbot performance and potentially lead to incorrect results[24]. Despite the growing adoption of LLM-based chatbots in healthcare, no prior work has examined the impact of ASR integration on RAG-based chatbots in the context of Indian primary healthcare settings.

Given the challenges of evaluating healthcare chatbots, expert human assessments remain the gold standard, ensuring accuracy, relevance, and safety. However, this process is resource-intensive. Recent studies have explored LLM-based evaluators as a complementary approach [13]. Li et al. [27] found that pairwise evaluations align closely with human judgment when assessing chatbot responses across different categories. While LLM-based evaluations offer scalability, they should complement rather than replace human assessments to ensure reliability [40, 42]. To leverage the strengths of both approaches, our evaluation pipeline integrates LLM-based assessments for rapid, large-scale testing while relying on human evaluation for deeper, high-quality validation.

3 Dataset Preparation Methodology

We created an evaluation dataset to assess our RAG-based chatbot, leveraging the expertise of our in-house public health specialists. The dataset is based on English-language training modules [5] developed by government authorities in India for training Accredited Social Health Activists (ASHAs). Our experts curated a Question-Answer dataset using these training materials, ensuring that the questions reflect real-world scenarios and challenges faced by frontline healthcare workers.

The dataset comprises 509 questions covering key topics such as pregnancy, maternal health, and child health. The Themes of these questions are shown in Figure 3. Each entry includes a question, its corresponding ground truth answer, and metadata such as the document name and page number from which the question was derived. This allows us to evaluate retrieval recall alongside comparing the chatbot's response with the ground truth answer.

For example:

- **Question:** When should a pregnant woman visit the clinic for the fourth time?
- **Ground Truth Answer:** The fourth visit should be after 36 weeks.
- **Document Name:** Induction Training Module for ASHAs in Urban Areas (English)
- **Page Number:** 78

To simulate real-world conditions, taking inspiration from Wang et al. [39], we introduced errors using various transformation functions at both the word and character levels to the medical questions along with creating paraphrased versions of the questions. Each mistake level represents a specified percentage of words transformed within the original query, categorized as follows: 1-10%, 11-20%, 21-30%, 31-40%, and so on till 91-100%. For example, a mistake level of 1-10% indicates that between 1 and 10 percentage of the words in a query are modified. We removed duplicate queries from the dataset. Our final dataset size is **48,828**. Data distribution across each mistake level is presented in Figure 2.

The Transformation functions are described further in the following sub-sections.

3.1 Character-Level Transformations

Typos and other character-level errors are common in real-world text inputs and have been known to degrade the performance of NLP systems [10]. Research has shown that evaluating these perturbations is crucial for assessing model robustness and ensuring real-world preparedness [35]. To test our chatbot's resilience against such errors, we implemented the following transformations:

Add: Insert a random lowercase letter $[a-z]$ at a random position in the word.

Remove: Delete a character from a random position in the word.

Substitute: Replace a character with its adjacent counterpart on a QWERTY keyboard to simulate human typing errors.

Swap: Swap the positions of two adjacent characters in the word.

3.2 Word-Level Transformations

Word-level perturbations introduce variations that test a chatbot's ability to handle real-world language inconsistencies. Users whose native language is not English may use a phonetically similar word, or mistakenly insert a space while typing. We implemented the following transformations:

Split Word: Randomly split a word into two separate words at a random index.

Phonetic Similarity: Generate a phonetically similar word to the original. We used GPT-4o [7] to create these examples. This type of transformation helps evaluate the chatbot's ability to handle words that sound alike but may have different spellings

Medical Terms: Here we only create mistakes in medical terms. First, we identify medical terms in a given query using GPT-4o [7]. Then, we introduce spelling errors in one medical term at a time, generating multiple variations while keeping the rest of the query unchanged.

3.3 Sentence-Level Transformations

Paraphrased queries are essential to evaluate the robustness of the RAG chatbot in handling variations in user input. In real-world scenarios, users may phrase the same query differently, and the chatbot should still retrieve relevant and accurate responses. To simulate this, we generated paraphrased queries using an LLM (GPT-4o) [7]. For example, the original query "What are the symptoms of anemia during pregnancy?" was paraphrased as "How can I identify if a pregnant woman has anemia?" This ensures that the chatbot is tested on semantically similar yet syntactically different inputs. Prompts used for various transformation functions to introduce noise in the input queries can be found in Appendix 8.1.

Examples of transformed queries for each transformation function are presented in Table 9.

3.4 Out of Domain queries

To evaluate the performance of the RAG chatbot in abstaining from answering out-of-domain queries, we generate a total of 533 queries

unrelated to healthcare across various domains, including Sports, Finance, Agriculture, Banking, Politics, and Education, using GPT-4o[7].

3.5 Audio Dataset

To evaluate our ASR system on local Indian dialects and medical vocabulary, we collected an audio dataset from two districts in Uttar Pradesh, India: Bahraich and Barabanki. With the assistance of our medical experts, we curated a list of medical queries and provided them to ASHA workers in these districts. The ASHA workers were then asked to speak these queries in Hindi to curate this dataset. Our dataset comprises 1,026 audio recordings.

4 Methodology

4.1 RAG Pipeline

The RAG (Retrieval-Augmented Generation) [26] pipeline consists of three key components:

- **Knowledge Base**: A collection of documents used to generate answers.
- **Retriever**: A model responsible for identifying the most relevant documents for a given query.
- **Answer Generator (LLM)**: A Large Language Model that generates answers based on the retrieved documents and the query.

Given a question Q, the pipeline operates in two steps:

(1) **Retrieval Step**: The retriever identifies the **top-k documents** that are semantically similar to the query.
(2) **Generation Step**: The LLM uses the retrieved documents and the query as context to generate an answer.

In our evaluations, we use `text-embedding-3-large` [3] (with an embedding dimension of 1024) as the retrieval model. We choose multiple LLM models gpt-4o-mini [8], qwen2.5:3b [33], llama3.2:3b [9], gemma2:2b [38] and 6 bit quantized mistral:7b [19] that are suitable for production-level chatbots, both open-source and close-sourced. To ensure reliability and avoid hallucinations, we prompt the LLM to provide a response **only if the answer is present in the top-k retrieved documents**. If the answer is not found, the LLM returns a predefined `No_answer` response:

> *"I don't have an answer to your question. My knowledge is based on the information provided by the government for your role. Please try rephrasing your question or ask something else related to your role."*

4.2 Experiment Details

To evaluate the robustness of the system, we examine the response generated by the RAG pipeline to the transformed queries generated in the Section 3 and compare it with response generated by the original queries.

(1) **Query Transformation:** For a given query q, we apply a transformation function to create a modified version of the query, denoted as q'.
(2) **Pipeline Execution:** Both the original query q and the transformed query q' are independently processed through the RAG pipeline to generate answer and the sources.
(3) **Comparison of Outputs:** The answers generated for q and q' along with the retrieval metadata (Document Name and Pages) are then compared to assess similarity in answer and correctness of the retrieved sources. Specifically, we analyze:
   - **Semantic Similarity:** Whether the answers convey the same meaning, despite the query transformation.
   - **Retrieval Differences:** Any variations in the top-k documents retrieved for q and q', which could impact the quality of the generated answers.

4.3 Answer Response Evaluation

We evaluate responses to transformed queries R' and original queries R using both human and LLM evaluators. The LLM evaluator offers scalability for testing variations, while human evaluation validates the findings.

LLM Evaluator: We use Gemini-1.5-Flash [37] to score responses (0, 0.5, or 1) based on similarity between R and R' , as outlined in Appendix 8.2. We compute the mean LLM score for transformed query responses per transformation function and a weighted aggregate score across all functions, based on the number of transformed queries per transformation function.

Human Evaluation: We sample 150 transformed queries (15 per transformation function) and ask a medical expert to compare the responses R' and R. For each R' , we answer 'Yes' or 'No' to these questions: (1) Does overall meaning and intent of R' same as R?, (2) Does R' include all key information from R? Scores are assigned: 1 if both are 'Yes', 0.5 if one is 'Yes', and 0 if both are 'No'. We then average these scores across all transformed query responses for each model.

Reference based Evaluation: We use below mentioned reference based metrics to compare R and R'.

(1) **ROUGE-L score (recall)** [29] **:** Suppose there are 2 sequences H and G, rougeL score is the ratio of length of longest common subsequence (LCS) between H and G to the total number of unigrams in G. We calculate rougeL score between between R' and R.
(2) **Cosine Similarity:** We measure cosine similarity between embeddings of R' and R created using 1024 dimensions of text-embedding-3-large embedding model [3].

4.4 Retrieval Recall

Our pipeline returns the pages it retrieved from the Knowledge Base, along with the answer. To observe the effect of input transformations on the retrieval abilities of the pipeline we compare the retrieval recall [6] between original queries and their transformed queries, which are generated using a transformation function.

(1) **Recall@k :** Whether the relevant page is present among top-k (we used k=5) retrieved pages.
(2) **Retrieval recall for original and transformed queries:**
   - For each original query q_i, we calculate recall@k, denoted as rr_{original_i}.
   - For each transformed query q_{ij} (a variation of the original query q_i, where j is a variation), we compute recall@k, denoted as $rr_{\text{transformed}_{ij}}$.
(3) **Change in retrieval recall**

- **Average recall of transformed queries:** For each original query, we compute the average recall@k across all its transformed variations.

$$rr_{\text{transformed}_i} = \frac{1}{n} \sum_{j=1}^{n} rr_{\text{transformed}_{ij}} \quad (1)$$

where n is the total number of transformed queries of the original query q_i.

- **Difference in retrieval recall (Δ_i):** The difference between the retrieval recall of the original query and the average recall of its transformed queries is calculated as:

$$\Delta_i = rr_{\text{original}_i} - rr_{\text{transformed}_i} \quad (2)$$

(4) **Average Change in Retrieval Recall across all queries:**

- We calculate the average change in recall across all original queries.

$$\Delta = \frac{1}{N_{\text{original}}} \sum_{i=1}^{N_{\text{original}}} \Delta_i \quad (3)$$

where N_{original} is the total number of original queries.

5 Results

5.1 Retrieval Recall results

Table 1 presents the average recall@k difference between original queries and their transformed counterparts for each transformation function. On average, we observe a **4%** drop in retrieval recall for transformed queries compared to the original queries. This shows that the retrieval part is affected by the input perturbations. In the next few sections we will focus on the answer changes.

Table 1: Δ denotes the average difference in retrieval recall between the original query and its transformed queries, which are generated using a specific transformation function.

Transformation Function	Δ
Add	0.04
Remove	0.07
Substitute	0.06
Swap	0.04
Paraphrasing	0.05
Phonetic-sound	0.08
Split-word	0.05
Medical terms	0.06

5.2 Answer Response Evaluation Results

In this section, we analyze the impact of input perturbations on output responses. We begin by examining the occurrence of `No_Answer` responses in original queries and compare them with those in transformed queries. On average, transformed queries result in **12-17% more** `No_Answer` responses as compared to orginal queries across different models, suggesting that input perturbations can negatively impact the user experience. Despite having the necessary knowledge, the RAG pipeline fails to answer certain queries due to these perturbations. Table 2 contains the occurence of `No_answers` across different models we tested. From the table 3, we observe that phonetic similarity mistakes get more of `No_Answer` responses as compared to other transformations.

Table 2: percentage difference in count of `No_answer` responses to original queries and transformed queries

LLM model	Avg % diff in `No_answer` response counts
gpt-4o-mini	-16.74
qwen2.5:3b	-12.97
llama3.2:3b	-12.44
gemma2:2b	-12.93
mistral:7b[Q6_k]	-11.81

We further compare the responses for the queries which generated answer other than `No_answers` across the following robustness metrics:

- L_{go} denotes the LLM evaluator score comparing ground truth responses of original queries with LLM-generated responses to the same queries.
- L_{ot} represents the mean LLM evaluator score comparing responses generated from original queries and their transformed variants.
- L_{gt} is the mean LLM evaluator score comparing ground truth responses with responses to transformed queries.
- $Rouge_{ot}$ indicates the ROUGE-L score between responses to original and transformed queries.
- $Cosine_{ot}$ measures the cosine similarity between vector embeddings of responses to original and transformed queries.

A weighted aggregate score is computed for each metric across all transformation functions, where the weight corresponds to the number of transformed queries generated by each transformation function. Higher values (closer to 1) for metrics L_{go}, L_{ot}, L_{gt}, $Rouge_{ot}$, and $Cosine_{ot}$ indicate greater similarity between responses and, consequently, higher robustness to these transformations.

From Table 4 and Table 5, we observe variations in the responses generated by LLMs for transformed queries. However, there is no significant difference between the ground truth answers and responses to transformed queries, suggesting that LLMs either reject answering or provide responses largely comparable to the ground truth.

Furthermore, in Table 5, the scores for each transformation function confirm that responses generated by GPT-4o-mini for transformed queries are mostly similar to those for original queries. Notably, GPT-4o-mini also has the highest rate of `No_Answer` responses, indicating that it may be a more conservative model. This suggests that GPT-4o-mini may be more robust to perturbations, rejecting uncertain queries while still generating answers when confident in their correctness. Human evaluation results also exhibit a similar pattern across different models, aligning with the LLM evaluator's assessments. Table 6 presents the human evaluation results.

Furthermore, we analyze how the divergence between responses to original and transformed queries increases as the mistake level in the query rises. This trend is illustrated in Figure 1.

Table 3: percentage difference in count of `No_answer` responses for original and transformed queries across each transformation function.

Perturbation Type	**Character**				**Word**			**Sentence**
LLM model	Add	Remove	Substitute	Swap	Split word	Phonetic similarity	Medical terms	Paraphrasing
Support	7410	7281	7381	7300	7312	5096	6542	506
gpt-4o-mini	-18	-17	-21	-17	-11	-27	-9	-8
qwen2.5:3b	-14	-12	-16	-13	-6	-26	-8	-1
Llama3.2:3b	-14	-11	-16	-12	-6	-25	-7	-2
gemma2:2b	-14	-11	-15	-13	-8	-24	-9	-4
mistral:7b[Q6_k]	-13	-11	-14	-12	-6	-22	-8	-2

Table 4: Response Evaluation of various LLM models.

LLM model	L_{go}	L_{ot}	L_{gt}	$Rouge_{ot}$	$Cosine_{ot}$
gpt-4o-mini	**0.73**	**0.86**	**0.73**	**0.83**	**0.96**
qwen2.5:3b	0.59	0.68	0.56	0.63	0.9
Llama3.2:3b	0.51	0.58	0.48	0.59	0.83
gemma2:2b	0.41	0.55	0.38	0.47	0.84
mistral:7b[Q6_k]	0.37	0.5	0.35	0.46	0.84

Table 5: L_{ot} and L_{gt} scores of each transformation function for various LLM models. The highest score among various LLM models for each transformation function is shown in bold.

Perturbation Type	**Character**								**Word**						**Sentence**	
Transformation Function	Add		Remove		Substitute		Swap		Split word		Phonetic similarity		Medical terms		Paraphrasing	
Support	7410		7281		7381		7300		7312		5096		6542		506	
Score	L_{ot}	L_{gt}	L_{ot}	L_{gt}	L_{ot}	L_{gt}	L_{ot}	L_{gt}	L_{ot}	L_{gt}	L_{ot}	L_{gt}	L_{ot}	L_{gt}	L_{ot}	L_{gt}
gpt-4o-mini	**0.86**	**0.74**	**0.85**	**0.72**	**0.85**	**0.72**	**0.86**	**0.73**	**0.86**	**0.73**	**0.85**	**0.73**	**0.88**	**0.73**	**0.82**	**0.71**
qwen2.5:3b	0.68	0.56	0.66	0.55	0.65	0.55	0.68	0.56	0.7	0.58	0.66	0.55	0.73	0.59	0.67	0.56
Llama3.2:3b	0.59	0.48	0.56	0.45	0.55	0.46	0.59	0.48	0.59	0.49	0.55	0.46	0.65	0.52	0.57	0.46
gemma2:2b	0.55	0.39	0.53	0.37	0.53	0.38	0.56	0.38	0.58	0.39	0.48	0.37	0.6	0.38	0.52	0.38
mistral:7b[Q6_k]	0.51	0.36	0.49	0.34	0.48	0.34	0.5	0.35	0.52	0.35	0.44	0.33	0.56	0.36	0.48	0.37

Table 6: L_{ot} and Human evaluation score on the 150 sample transformed queries.

LLM model	L_{ot}	Human Eval
gpt-4o-mini	0.86	0.92
qwen2.5:3b	0.71	0.78
Llama3.2:3b	0.58	0.71
gemma2:2b	0.57	0.6
mistral:7b[Q6_k]	0.51	0.54

There is not much difference however for the generated responses across different transformation functions, as can be seen in Table 5

5.3 ASR Transcriptions

The audio recordings in our dataset are transcribed using the AI4Bharat Indic ASR model [4]. We evaluate the robustness of our system to ASR-generated transcriptions by measuring the percentage difference in `No_answer` response counts between baseline queries and their transcriptions. Additionally, we report the LLM evaluator score L_{ot} between responses to baseline queries and ASR transcriptions. As shown in Table 7, all LLMs exhibit a slight increase in `No_answer` responses for ASR transcriptions compared to baseline queries. This suggests that errors introduced by the speech recognition model can prevent the RAG system from generating an answer, even when the correct information is available in the knowledge base. On observation, the transcription errors usually are more prevalent in domain-specific terminologies, for example, while asking about colostrum, it was picked as chlorostem; asphyxia was picked as X physia; completely changing the meaning of what the user asked.

Notably, GPT-4o-mini achieves a higher L_{ot} score than other LLMs, indicating greater robustness in its responses to ASR-transcribed queries. These findings highlight the importance of improving upstream components like ASR, as errors in transcription can have

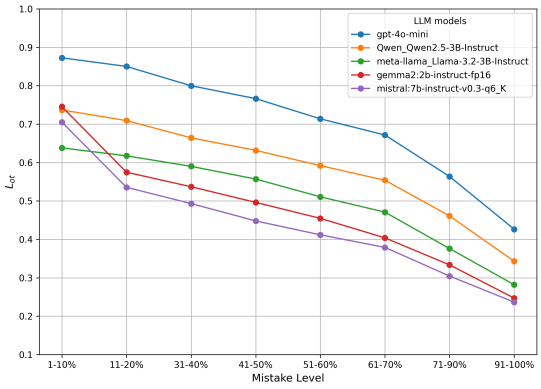

Figure 1: L_{ot} score of various LLM models across different Mistake levels.

a cascading impact on the overall utility of the system. Enhancing ASR accuracy can help ensure that the benefits of LLM-based technologies reach a wider user base.

Table 7: Above Table shows average difference in `No_answer` response counts between baseline queries and its ASR transcriptions for different LLM models.

LLM model	L_{ot}	Avg % diff in `No_answer` response counts
gpt-4o-mini	0.80	-5
qwen2.5:3b	0.57	-8
Llama3.2:3b	0.52	-5
gemma2:2b	0.73	-7
mistral:7b[Q6_k]	0.69	-4

5.4 Out of Domain queries

In our RAG pipeline, we prompt the LLM to respond only to queries if the answer is found in the context provided by retrieval model. Ideally our pipeline should generate `No_answer` responses for all the out of domain queries.

Table 8: Comparison of fraction of `No_Answer` responses for out-of-domain queries across different LLMs in a RAG setting.

LLM model	Fraction of `No_answer` responses
gpt-4o-mini	1
qwen2.5:3b	0.93
Llama3.2:3b	0.95
gemma2:2b	0.72
mistral:7b[Q6_k]	0.71

Table 8 presents the fraction of out-of-domain queries for which the pipeline generated a `No_answer` response. Notably, GPT-4o-mini produced a `No_answer` response for 532 out of 533 queries, demonstrating strong robustness in handling out-of-domain inputs. In contrast, models like Gemma and Mistral performed worse, indicating a higher tendency to generate responses even when the query falls outside the system's knowledge scope which is not ideal for a production-settings.

6 Conclusion

In this work, we propose the need and a methodology to evaluate the robustness of Retrieval-Augmented Generation (RAG) systems against various types of real-world noise, such as typos and imperfect Automatic Speech Recognition (ASR) transcriptions. Our evaluation results demonstrate the impact of input errors on the RAG pipeline, emphasizing the importance of stress-testing for robustness before deploying such systems in real-world settings. These findings underscore the need for rigorous testing to ensure reliability and effectiveness, particularly in high-stakes applications like healthcare. Potential solutions to mitigate these effects like spell-checkers or using language models for input correction can be explored.

7 Limitations

We evaluated the robustness of LLMs in a RAG setting on English queries except the ASR transcriptions. Our future work will extend this to multilingual queries. The current work evaluated 5 LLMs chosen based on their practicality in the production environment due to their relatively smaller size. The pipeline proposed is however model-agnostic given the rapid evolvement of the LLMs.

Acknowledgments

We acknowledge the contributions of our in-house public health experts in developing the original evaluation dataset and supporting the evaluation process. We also extend our thanks to the Product, Engineering, Quality Assurance, and Monitoring, Evaluation, and Learning (MEL) teams behind this chatbot, whose collective efforts laid the groundwork for this project. We appreciate the valuable feedback provided by the senior members of our Machine Learning team on the manuscript. Finally, we thank the Gates Foundation for the support through their grant, which made this work possible.

References

[1] [n. d.]. https://www.perplexity.ai/. [Accessed 14-02-2025].
[2] [n. d.]. https://openai.com/index/introducing-chatgpt-search/. [Accessed 14-02-2025].
[3] [n. d.]. https://openai.com/index/new-embedding-models-and-api-updates. [Accessed 25-02-2025].
[4] [n. d.]. https://github.com/AI4Bharat/indic-asr-api-backend. [Accessed 14-02-2025].
[5] [n. d.]. ASHA Training Modules. https://nhm.gov.in/index1.php?lang=1&level=3&sublinkid=184&lid=257. [Accessed 25-02-2025].
[6] [n. d.]. Evaluation in Information Retrieval. https://nlp.stanford.edu/IR-book/pdf/08eval.pdf. [Accessed 25-02-2025].
[7] [n. d.]. GPT-4o. https://openai.com/index/hello-gpt-4o/. [Accessed 25-02-2025].
[8] [n. d.]. GPT-4o-mini. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. [Accessed 25-02-2025].
[9] [n. d.]. Lllama3.2. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/. [Accessed 25-02-2025].
[10] Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173* (2017).

[11] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 610–623.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[13] Cheng-Han Chiang and Hung-yi Lee. 2023. Can Large Language Models Be an Alternative to Human Evaluations?. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 15607–15631. doi:10.18653/v1/2023.acl-long.870

[14] Sukmin Cho, Soyeong Jeong, Jeongyeon Seo, Taeho Hwang, and Jong C Park. 2024. Typos that Broke the RAG's Back: Genetic Attack on RAG Pipeline by Simulating Documents in the Wild via Low-level Perturbations. *arXiv preprint arXiv:2404.13948* (2024).

[15] Brian Formento, Chuan Sheng Foo, Luu Anh Tuan, and See Kiong Ng. 2023. Using punctuation as an adversarial attack on deep learning-based NLP systems: An empirical study. In *Findings of the Association for Computational Linguistics: EACL 2023*. 1–34.

[16] Varun Gumma, Anandhita Raghunath, Mohit Jain, and Sunayana Sitaram. 2024. HEALTH-PARIKSHA: Assessing RAG Models for Health Chatbots in Real-World Multilingual Settings. *arXiv preprint arXiv:2410.13671* (2024).

[17] Bright Huo, Amy Boyle, Nana Marfo, Wimonchat Tangamornsuksan, Jeremy P Steen, Tyler McKechnie, Yung Lee, Julio Mayol, Stavros A Antoniou, Arun James Thirunavukarasu, et al. 2025. Large Language Models for Chatbot Health Advice Studies: A Systematic Review. *JAMA Network Open* 8, 2 (2025), e2457879–e2457879.

[18] Tahir Javed, Sumanth Doddapaneni, Abhigyan Raman, Kaushal Santosh Bhogale, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Towards Building ASR Systems for the Next Billion Users. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 10 (Jun. 2022), 10813–10821. doi:10.1609/aaai.v36i10.21327

[19] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[20] Kalpana Joshi and K Verma. 2018. Knowledge of Anganwadi workers and their problems in Rural ICDS block. *IP journal of paediatrics and nursing science* 1, 1 (2018), 8–14.

[21] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1601–1611. doi:10.18653/v1/P17-1147

[22] Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2024. REALTIME QA: what's the answer right now? *Advances in Neural Information Processing Systems* 36 (2024).

[23] Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2023. RealTime QA: What's the Answer Right Now?. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 49025–49043. https://proceedings.neurips.cc/paper_files/paper/2023/file/9941624ef7f867a502732b5154d30cb7-Paper-Datasets_and_Benchmarks.pdf

[24] Korbinian Kuhn, Verena Kersken, Benedikt Reuter, Niklas Egger, and Gottfried Zimmermann. 2024. Measuring the accuracy of automatic speech recognition solutions. *ACM Transactions on Accessible Computing* 16, 4 (2024), 1–23.

[25] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[27] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579* (2024).

[28] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6449–6464. doi:10.18653/v1/2023.emnlp-main.397

[29] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[30] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 9802–9822. doi:10.18653/v1/2023.acl-long.546

[31] Zabir Al Nazi and Wei Peng. 2024. Large language models in healthcare and medical domain: A review. In *Informatics*, Vol. 11. MDPI, 57.

[32] Robert Osazuwa Ness, Katie Matton, Hayden Helm, Sheng Zhang, Junaid Bajwa, Carey E Priebe, and Eric Horvitz. 2024. MedFuzz: Exploring the Robustness of Large Language Models in Medical Question Answering. *arXiv preprint arXiv:2406.06573* (2024).

[33] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. arXiv:2412.15115

[34] Pragnya Ramjee, Mehak Chhokar, Bhuvan Sachdeva, Mahendra Meena, Hamid Abdullah, Aditya Vashistha, Ruchit Nagar, and Mohit Jain. 2024. ASHABot: An LLM-Powered Chatbot to Support the Informational Needs of Community Health Workers. *arXiv preprint arXiv:2409.10913* (2024).

[35] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. *arXiv preprint arXiv:2005.04118* (2020).

[36] Seema Sharma, Amit Kumar, Sneh Kumari, Divyae Kansal, and Suryamani Pandey. 2023. A comparative study of knowledge of accredited social health activist (ASHA) workers regarding child health services working in rural and urban areas of a block of Haryana. *Indian Journal of Forensic and Community Medicine* 9, 4 (2023), 169–172.

[37] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530* (2024).

[38] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause,

Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving Open Language Models at a Practical Size. arXiv:2408.00118 [cs.CL] https://arxiv.org/abs/2408.00118

[39] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840* (2021).

[40] Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. Pariksha: A large-scale investigation of human-llm evaluator agreement on multilingual and multi-cultural data. *arXiv preprint arXiv:2406.15053* (2024).

[41] Rui Yang, Ting Fang Tan, Wei Lu, Arun James Thirunavukarasu, Daniel Shu Wei Ting, and Nan Liu. 2023. Large language models in health care: Development, applications, and challenges. *Health Care Science* 2, 4 (2023), 255–263.

[42] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623.

[43] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv e-prints* (2023), arXiv–2306.

[44] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models. *arXiv preprint arXiv:2402.07867* (2024).

8 Appendix

8.1 Prompts used to create transformed queries using various Transformation functions.

Transformation function: Paraphrasing

> **System Prompt:**
> f"'You will be given a query in triple quotes. Your task is to paraphrase the words within the query, without changing the overall meaning and context. Output should contain only the modified query. \n {query}"'.

Transformation function: Phonetic similarity

> **System Prompt:**
> f"You will be given a list of words in triple quotes, each of which you need to modify to create a phonetically similar word with different spelling. Return the output as a JSON object, where each original word is a key, and its modified word is the value.\n "'{list of words}"'"

Transformation function: Medical terms

> **Prompt used to Detect medical terms:**
> f"Given a medical query enclosed in triple quotes, identify and extract terms related to healthcare. Return the output as a JSON object with key 'Medical_terms' and value contains list of relevant healthcare terms. Ensure that the extracted terms retain their original spelling and letter casing exactly as they appear in the query.\n "'query"'"
> **Prompt used to introduce spelling errors in medical terms**
> f"Given a medical term enclosed in triple quotes, generate a list of 10 unique incorrect variations of the medical term by introducing spelling and typographical mistakes. Output should be a json object with key as medical term and value is the list of erroneous terms. Ensure that all variations are unique and realistic and also variations should not be same as original medical term.\n "'medical_term"'"

Prompt used to generate Out of Domain queries

> **System Prompt:**
> f"We have an ASHA Health Assistant that responds to medical queries posed by ASHA workers. We aim to evaluate the robustness of the Health Assistant to out-of-domain queries that are unrelated to healthcare. Given a domain enclosed in triple quotes, generate at least 50 to 100 queries related to that domain in a JSON object, with keys labeled as 'domain' and 'queries' \n "'{domain}"'"

8.2 LLM Evaluator

Prompt used to compare between the ground truth and generated answer.

> **System Prompt:**
> You will be provided with two answers related to primary healthcare: a ground truth answer and a generated answer. Your task is to compare the two answers and assign an LLM score based on the following criteria:
> **Semantic Similarity:** The overall meaning and context and intent of the generated answer are the same as the ground truth answer. Ignore minor differences such as synonyms, grammar, punctuation, or formatting, as long as the meaning remains unchanged.
> **Key Information:** The generated answer should not omit critical details from the ground truth. Do not penalize if the generated answer includes additional information that is correct and relevant.
> **Scoring:**
> - **1**: Both criteria are fully satisfied. The generated answer accurately captures both the meaning and all key information of the ground truth answer.
> - **0.5**: The first criterion is satisfied but second criterion (Key Information) is not satisfied.
> - **0**: Neither of the criteria is satisfied.
>
> Additionally, provide a reason for the assigned LLM score. Output should be a json object with keys as `'llm_score'` and `'reason'`.

8.3 Dataset Preparation

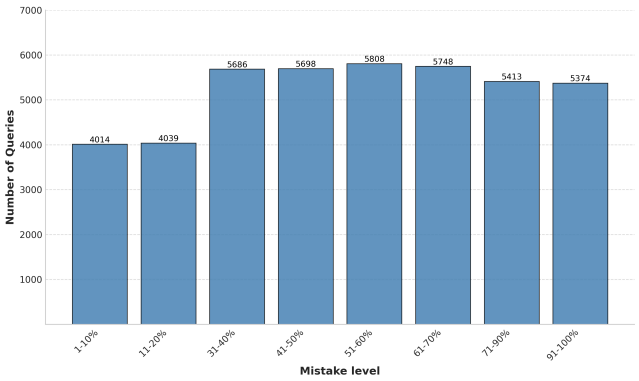

Figure 2: Sample size of each mistake level in robustness dataset. Numbers on each bar represents number of queries in each mistake level.

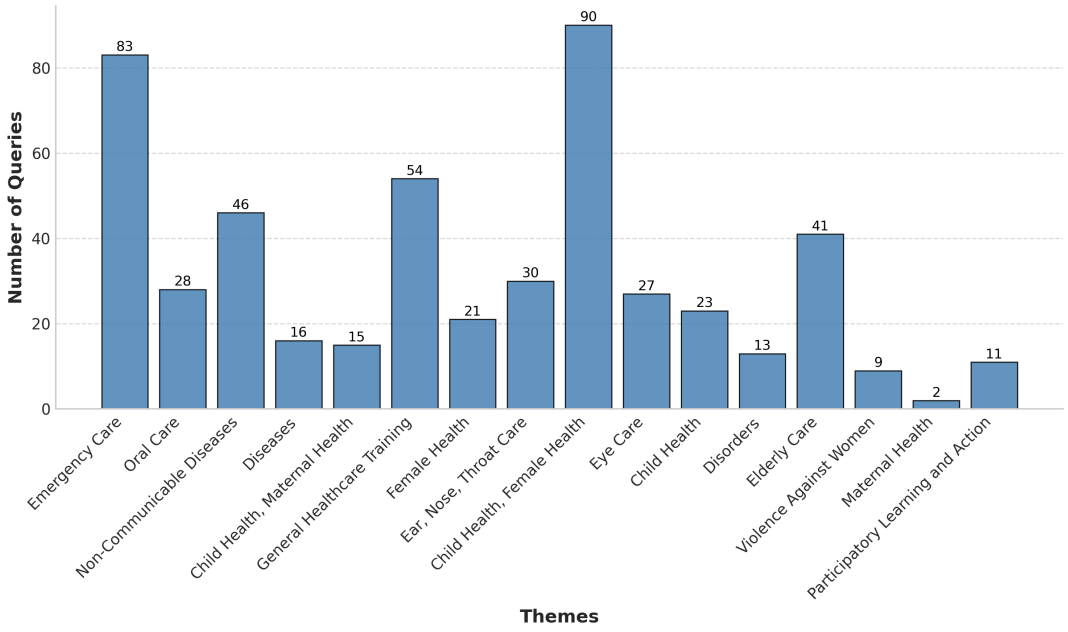

Figure 3: Distribution of query themes across the evaluation dataset. Number on each bar represents number of queries in each theme.

Table 9: Examples of transformed queries with various perturbation along with original query. All modified words in the transformed queries are highlighted in red.

Transformation Function	Support	Perturbation level	Transformed query
Original Query	509	None	What are risk factors for diseases of the ear
Add	7410	character	What arze grisk fakctors jfor diseasaes of thqe ear
Remove	7281	character	Wat ae ris factors or diseses of he ear
Substitute	7381	character	Whst ars risk factord fir diseqses pf fhe eqr
Swap	7300	character	What are irsk afctors ofr diseases fo hte era
Split word	7312	word	Wh at ar e risk f actors for dise ases of th e e ar
Phonetic similarity	5096	word	What are risc fakturz for dizeez of the ear
Medical terms	6542	word	What are risk factors for diseasess of the ear
Paraphrasing	506	sentence	What are the contributing factors for ear diseases?