GUIDEQ: Framework for Guided Questioning for progressive informational collection and classification

Priya Mishra [‡] Suraj Racha [‡] Kaustubh Ponkshe
Adit Akarsh Ganesh Ramakrishnan
Indian Institute of Technology Bombay
{23d0373,23d1627}@iitb.ac.in,
{ponkshekaustubh11, aditakarsh27}@gmail.com,
ganesh@cse.iitb.ac.in

Abstract

Question Answering (QA) is an important part of tasks like text classification through information gathering. These are finding increasing use in sectors like healthcare, customer support, legal services, *etc.*, to collect and classify responses into actionable categories. LLMs, although can support QA systems, they face a significant challenge of insufficient or missing information for classification. Although LLMs excel in reasoning, the models rely on their parametric knowledge to answer. However, questioning the user requires domain-specific information aiding to collect accurate information. Our work, GUIDEQ, presents a novel framework for asking guided questions to further progress a partial information. We leverage the explainability derived from the classifier model along with LLMs to ask guided questions, further enhancing the information. This further information helps in more accurate classification of a text. GUIDEQ derives the most significant key-words representative of a label using occlusions. We develop GUIDEQ's prompting strategy for guided questions based on the top-3 classifier label outputs and the significant words, to seek specific and relevant information, and classify in a targeted manner. Through our experimental results, we demonstrate that GUIDEQ outperforms other LLM-based baselines, yielding improved F1-Score through the accurate collection of relevant further information. We perform various analytical studies and also report better question quality compared to our method.

1 Introduction

Question Answering (QA) systems have been an integral part of the NLP landscape (Moise et al., 2010). In particular, the emergence of LLMs has enabled reasoning, proactive questioning, and better semantic understanding of the user response during questing answering or dialogue (Wang et al., 2023a). Proactive questioning is another important limb of such systems wherein the bot engages with the user and directs the conversation ahead(Keskar et al., 2019; Sun et al., 2023). It finds applications in many places like medical, customer support systems, and legal systems.

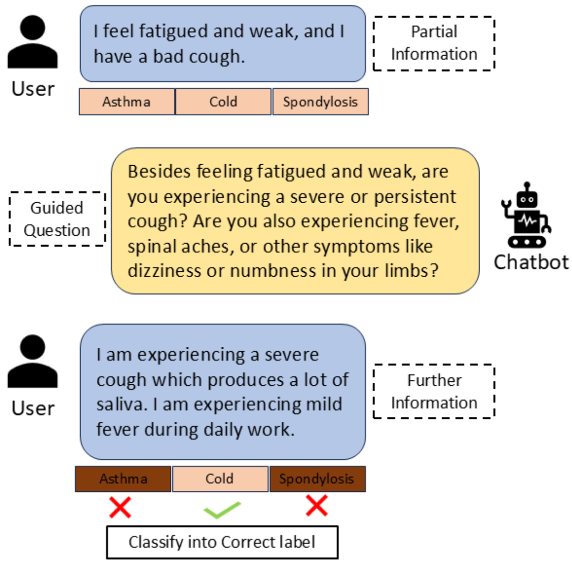

Figure 1: Illustration of partial information by user followed by a specific guided question

A more specific challenge such systems face is of the static classification of user textual data. Often a textual data needs to be classified into a particular category or label (Wang et al., 2023b; Chen et al., 2022). For example, consider a system to classify a patient symptoms descriptions into one of the disease conditions, or a customer complaint system wherein a user writes a complaint to be categorized into a particular category. A practical challenge faced by such static categorization is inadequate or missing information toward appropriate routing of the user input to an actionable category. It can be immensely benefited by introduction of questioning

[‡]Equal contributions

Findings of the Association for Computational Linguistics: NAACL 2025, pages 4645–4659

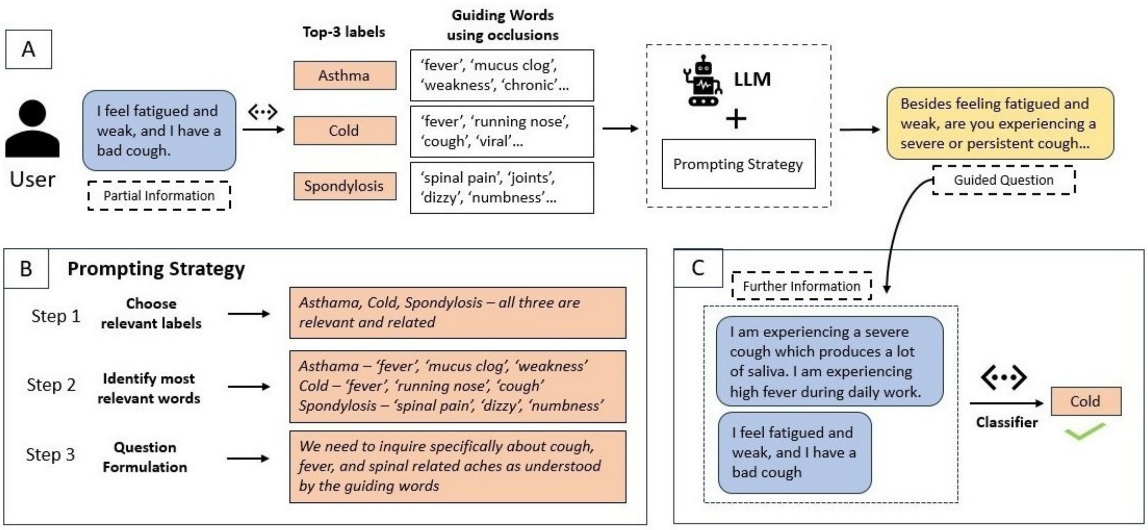

Figure 2: (A) Overall working framework of GUIDEQ to leverage LLM and label explainability for asking guided question. (B) Details of the prompting strategy used. (C) Final classification along with incremental information.

component to prompt for further information or knowledge, grounded in previous response and the domain itself. Figure 1 shows an example of the same.

Our work introduces a novel framework, GUIDEQ [1], aimed at framing guided questions based on prior partial information, such that specific relevant information can be asked for. This increment of information can aid for more accurate classification and completion of text. A classic techniques include classifier models for text classification(Devlin et al., 2019) (Liu et al., 2019) (He et al., 2020). LLMs, on the other hand, have shown impressive abilities for reasoning and context understanding (Nan et al., 2023). Further with techniques like in-context learning (Garg et al., 2022) and chain of thought (Wei et al., 2024) with few-shot exemplars (Brown et al., 2020), the task specific adaptation of LLMs greatly benefits. However, the LLM's parametric knowledge may still not capture domain specific requires. Finetuning options like FT (Full parametric training) and PEFT (Hu et al., 2021) for learning from data history posses a huge challenge of high computation cost (Sathish et al., 2024).

Our framework, GUIDEQ leverages the innate semantic understanding and reasoning ability of LLMs to combine with external classification explanability, for asking the most relevant guiding question. We specifically use Llama-3 8B-Instruct model (Touvron et al., 2023) for questioning and information seeking. We train DeBERTa and BERT models (Devlin et al., 2019) for classification tasks using complete information, which serves as the primary pivot for predicting the most probable labels.GUIDEQ aims to use the inherent explanations of classifications for question formation. For each class label we also learn the most significant words and phrases that contribute to the particular label classification. We use occlusions to find the keywords for a label using the training data. The LLM utilizes the top most probable classifier labels and their significant representative keywords to form the guiding questions. Intuitively, the keywords are representative of the most important concepts present in the label knowledge. The summarized overview with keywords example can be found in figure 2. The LLM finds similar concepts present in the partial information and the labels and frames a question based on the most distinguishing concepts between the labels. This helps direct further information that may not have been readily known to the user earlier.

We evaluate our framework, GUIDEQ, on 6 text classification datasets. We first report the F1-Score of partial information and how accuracy changes by questioning and appending the new answer. We also show how explanability is influenced by keywords, namely by testing with unigrams, bigrams, and trigrams. We also report a higher question generation quality compared to other baseline methods. Overall, our work, GUIDEQ, presents a novel

[1] Code available at: https://github.com/SDRMp/GuideQ

framework to ask guided questions when the initial text is incomplete or partial. The question is such that it effectively differentiates between the most likely labels.

We summarize the key contributions of our work as follows: (i) We introduce a novel framework, GUIDEQ, for providing guided questioning such that an initial partial information can be incremented. (ii) The guided questioning increments the information by leveraging explainability derived through top most confident classifier labels and their corresponding most significant key words. (iii) We show that further information collection through our framework significantly improves the classification accuracy as compared to other baselines. We also show that GUIDEQ generates more accurate and targeted questions in relation to the user query.

2 Related Works

2.1 Posthoc Explanations and LLMs

Post-hoc explanations enhance the interpretability of Large Language Models (LLMs) by providing insights into which input features influence model outputs, addressing the "black-box" nature of these models (Kroeger et al., 2024). The AMPLIFY framework (Krishna et al., 2024), for instance, uses attribution scores to generate natural language rationales, guiding LLMs to make more accurate predictions. Integrating LLMs with existing XAI algorithms like SHAP can produce more accessible and human-readable explanations. LLM-generated explanations are as effective as traditional gradient-based methods (Zytek et al., 2024).
Previous works use the technique of Integrated Gradients (Sundararajan et al., 2017), previously used for word-level attribution in language models(Enguehard, 2023) to generate keywords that contribute the most for a particular classification label.

2.2 Task oriented dialogues

Task-oriented dialogue systems aim to achieve specific goals through structured interactions, traditionally decomposing tasks into natural language understanding (NLU), dialogue management (DM), and natural language generation (NLG) . This modular approach often leads to error propagation and requires extensive domain-specific data (Budzianowski and Vulić, 2019). Recent advancements, like SimpleTOD (Hosseini-Asl et al., 2020), leverage LLMs to unify NLU, DM, and NLG into a single sequence prediction task. By treating all sub-tasks as a single sequence prediction problem, SimpleTOD exploits transfer learning from pre-trained models for improved performance. Pre-trained generative models can effectively support ToD by learning domain-specific tokens.

2.3 Information-seeking questions using LLMs

Large Language Models (LLMs) have significantly advanced the field of information-seeking question answering by generating contextually rich and coherent responses (Jin et al., 2023). Traditional search engines have evolved with models like GPT-3 and GPT-4, which can formulate search results in natural language, incorporating references from relevant sources. These models are particularly effective at summarizing and synthesizing information from various texts, making them valuable tools in medical and educational contexts. However, challenges such as "hallucination"—where the model generates plausible but incorrect information—highlight the need for mechanisms to verify and attribute sources accurately (Kamalloo et al., 2023).

2.4 In-context learning

In-context learning (ICL) has emerged as a novel paradigm where language models are capable of learning from a few examples within a context, making predictions without explicit parameter updates. The key mechanism of ICL is the use of demonstration contexts, which consist of input-output examples formatted as natural language templates (Li et al., 2023). This method allows large language models to perform various tasks by leveraging patterns learned from these demonstrations, which are provided as part of the input prompt. Recent studies highlight the adaptability of ICL to new tasks, significantly reducing computational costs compared to traditional supervised learning methods (Dong et al., 2024). Furthermore, ICL has demonstrated potential across different modalities, such as vision-language and speech tasks, by incorporating properly formatted data and architectural designs. Our work contributes to this growing field by utilizing ICL for the classification of incomplete sentences. By leveraging label explanations and guiding language models to ask targeted questions, we aim to refine user responses in medical contexts, ultimately enhancing the model's ability to generate accurate and relevant follow-up queries.

3 GUIDEQ - Methodology

3.1 Overview

Given a partial input x, the objective is to classify x (or x with additional information) into the correct label. Here, partial input refers to text that may be incomplete or lacking essential details. We use a classifier, C, to map the input to one of n labels, where the label set $L = \{l_1, l_2, \ldots, l_n\}$ is dataset-dependent. To enhance this process, we identify representative keywords and phrases using occlusion techniques for each label. For a label l_i, the associated keywords are denoted as $r_i = \{w_1^i, w_2^i, \ldots, w_k^i\}$.

GUIDEQ leverages a large language model (Llama-3 8B-Instruct in our study) to refine the classification through an interactive process. The model takes as input the top-k predicted labels, their corresponding keywords, and a prompting strategy. It then generates a question q designed to elicit information from the user related to the most relevant concepts in the keywords, further guiding the classification towards the correct label.

We divide our methodology as: (1) Classifier Finetuning, (2) Keyword Learning, and (3) Explainability-Driven Question Generation.

3.2 Classifier Finetuning

The first step in our GUIDEQ framework involves finetuning a classifier using labeled training data. Our dataset consists of domain-specific text-label pairs, which are divided into training (80%), evaluation (15%), and test (5%) sets.

We train C on the complete input text and its corresponding label. Training is conducted on the training split, with performance monitored on the evaluation split. By training on full texts, the model learns to output the most likely labels for any input, even if the input is incomplete during inference, as described in later sections.

3.3 Keywords learning

Given n possible labels, $L = \{l_1, l_2, \ldots, l_n\}$, the classifier assigns a probability score to each label for a given input. Our goal is to identify the most significant words or phrases (unigrams, bigrams, and trigrams) that represent each label. These keywords capture the core semantic and conceptual elements of the label's category. For example, the label 'fever' might be characterized by keywords like "high temperature" or "body ache," while a telecommunication label might feature keywords like "mobile phone" or "network issues." These keywords enhance the explainability of the classification.

We employ the occlusion method to identify the top-i significant words or word pairs for each label l_i, represented as $r_i = \{w_1^i, w_2^i, \ldots, w_k^i\}$. Occlusions involve systematically removing or masking words from the input and observing the effect on the model's confidence score. A sharp drop in confidence indicates that the removed word is crucial for the label prediction. Each word or phrase is then assigned a weight based on its importance, and the most relevant keywords are aggregated with additive weights for each label. To capture diverse concepts, we consider unigrams, bigrams, and trigrams for each label. Figure 2(A) illustrates how these guiding keywords are used for question generation. In our experiments, we include the top 15 word pairs per label in the LLM prompt to guide the question generation process effectively.

3.4 Explainability-Driven Question Generation

The final question generation stage follows the following method: First, we pass the incomplete input x to the classifier, which returns the top-k labels with the highest confidence scores. We set $k = 3$, selecting the top-3 most likely labels for x. Using these labels, we prompt the Llama-3 8B-Instruct model, applying a tailored prompting strategy to generate a guided question.

The prompt provided to the LLM includes the following components: (i) the partial input x; (ii) the top-3 predicted labels; (iii) the corresponding guiding keywords for each of these labels; (iv) a structured instruction prompt; and (v) a few-shot examples to guide the LLM's output.

The generated question, q, can be formulated as:

$$q = LLM(P \,||\, x \,||\, \{(l_1^x, r_1^x), (l_2^x, r_2^x), (l_3^x, r_3^x)\})$$

where P represents the instruction prompt, which combines the logical structure of the components with three few-shot examples. These examples illustrate how to generate questions that efficiently target missing or unclear information, aiding in more accurate classification. The prompting strategy for generating questions follows a structured, three-step process:

Step 1: The LLM first filters out irrelevant labels from the top-3 predictions, retaining only those

Dataset	Train	Val	Test
cnews	25,062	4,423	1,552
dbp	240,942	36,003	60,794 (876)
s2d	969	171	60
salad	17,174	3,031	1,064
stress	2,291	405	142
20NG	9,064	1,600	7,019 (1000)

Table 1: Dataset statistics used in experiments along with training, validation, and testing splits

most relevant to the input query. This is achieved by comparing the input with the keywords associated with each label, allowing the model to focus on labels that contextually align with the partial information.

Step 2: Next, the LLM examines the guiding keywords for the remaining labels. These keywords represent key concepts associated with each label, serving as focal points for further inquiry. The LLM identifies the most contextually relevant keywords that can expand upon the incomplete information in the input x. This step ensures that the follow-up questions target the most meaningful aspects of the missing information.

Step 3: Finally, the LLM uses the selected keywords to generate a coherent, targeted question. The question is designed to elicit specific details necessary for distinguishing between the potential labels, facilitating more accurate classification.

As shown in Figure 2(B), this strategy ensures a focused and contextually relevant interaction, especially for specialized categories, reducing the risk of hallucinations and irrelevant questions by grounding the response in the prior training data.

4 Experimental Setup

This section outlines the experimental configurations and evaluation protocols employed to assess the performance of our framework, GUIDEQ. The goal of GUIDEQ is to enhance classification under incomplete information by generating guided questions that elicit relevant, missing inputs. This targeted information retrieval facilitates more accurate classification, grounded in prior learned patterns.

4.1 Datasets

We evaluated GUIDEQ on six diverse text classification datasets, each representing a unique domain and presenting distinct challenges: (i) Crypto News (cnews) (Oliviervha, 2023),(ii) DBpedia (dbp) (Lehmann et al., 2015), (iii) Symptom2Disease (s2d) (Hassan et al., 2024), (iv) saladBench (salad) (Li et al., 2024), (v) Human stress Prediction (stress) (Kreesh, 2024), and (vi) 20 Newsgroups (20NG) (Lang, 1995), representing healthcare, financial, and behavioral domains respectively. Table 1 summarizes the total data instances used for each of the six dataset along with the exact split for training, validation, and testing.

We followed a standard 80%-15%-5% division into training, validation, and test sets for datasets without predefined splits (cnews, s2d, salad, and stress). For 20 Newsgroups (20NG) and DBPedia, which came with predefined splits, we extracted the validation set from the training data while preserving the original test set. Given the computational intensity of processing large test sets, particularly for 20NG (20 classes) and DBPedia (218 classes), we employed the Facility Location Selection method from the Apricot library to sample the test sets while maintaining class diversity. This sampling capped the maximum test samples at 1,000, yielding 1,000 samples for 20NG and 876 samples for DBPedia, enabling efficient evaluation while preserving result integrity. We provide more information about the dataset and it's relevance in appendix section A.

4.2 Baselines

We benchmark GUIDEQ against three baselines for question generation: (i) *Partial*: using only the initially provided partial information to assess classification performance without any additional inputs, serving as a direct comparison of classification under incomplete data. (ii) *LLM*: leveraging a standalone LLM (Llama-3 8B-Instruct), prompted to generate questions based on the partial input, representing a generic approach to eliciting missing information. As the input data mirrors realistic scenarios, the LLM's questions are grounded in semantic context and not in hypothetical constructs. (iii) *LLM-nk*: LLM with only labels and no keywords - combining the LLM with the top-3 predicted classification labels, where the LLM generates questions based solely on these labels without keywords. These baselines, denoted as Partial, LLM, and LLM-nk, provide a comprehensive evaluation of GUIDEQ's performance.

	BERT Classifier Model				DeBERTa Classifier Model			
Dataset	Partial	LLM	LLM-nk	GuideQ	Partial	LLM	LLM-nk	GuideQ
cnews	45.2	48.5 (3.3)	49.8 (4.9)	**50.9 (5.7)**	42.6	45.5 (2.9)	46.4 (3.8)	**49.6 (<u>7.0</u>)**
dbp	86.9	87.0 (0.1)	86.5 (-0.4)	**88.7 (1.8)**	85.0	84.9 (-0.1)	84.8 (-0.2)	**91.3 (<u>6.3</u>)**
s2d	61.1	72.3 (11.2)	66.9 (5.8)	**79.7 (18.6)**	64.7	71.5 (6.8)	68.4 (3.7)	**86.8 (<u>22.1</u>)**
salad	35.2	53.6 (18.4)	55.2 (20.0)	**57.1 (<u>22.2</u>)**	38.0	55.7 (17.7)	56.2 (18.2)	**58.7 (20.7)**
stress	32.3	**35.0 (2.7)**	33.3 (1.0)	32.9 (0.6)	43.1	41.4 (-1.7)	43.0 (-0.1)	**46.1 (<u>3.0</u>)**
20NG	67.5	68.2 (0.7)	68.0 (0.5)	**72.9 (<u>5.4</u>)**	63.2	64.0 (0.8)	63.9 (0.7)	**65.8 (2.6)**

Table 2: Comparison of % F1-Scores of GUIDEQ along with three baseline approaches - (i) partial: partial information; (ii) LLM: Only LLM is used for question framing; (iii) LLM-nk: LLM is provided with top 3 predictions. The results are reported for two classifier models: BERT and DeBERTa. Numbers in bracket constitute gain over partial information scores

4.3 Evaluation

We conduct multiple experiments to assess various aspects of the problem. For a robust classification analysis, we use two classifier models: BERT-uncased and DeBERTaV3 (a comparatively larger model). Throughout our experiments, we utilize Llama-3 8B-Instruct, an open-source LLM chosen for its strong reasoning capabilities and computational efficiency, offering a balance between performance and cost due to its smaller parametric size.

During testing, each data instance is split into two equal parts: the first half serves as the partial input, while the second half acts as the reference from which the generated question seeks to extract missing information. To evaluate question quality and answer relevance, we employ DeBERTaV3 finetuned on SQuadV2 (Rajpurkar et al., 2018), as the question-answering models, which extract the most relevant text from the reference based on the question. We set a 20% confidence threshold for answering.

We demonstrate four experimental settings to thoroughly analyze GUIDEQ:
(i) Classification Performance: We measure classification performance, which refers to correctly assigning the given text to its corresponding label by reporting F1-Scores. First, we report scores using only the partial information (the first half of each instance). For the other baselines, a question is generated, and relevant phrases are extracted from the reference text using the QA models with a 20% confidence threshold. This extracted information is appended to the partial input, and the combined text is classified. We use two-gram keywords for the important results while compare using one-gram and three-gram in a section 6.4.
(ii) Question Quality: We calculate the win rate between pairs of methods by determining which generated question is more aligned with the complete text (both partial input and reference). The win rate accuracy is reported for the following pairs: (GUIDEQ, LLM), and (GUIDEQ, $LLM - nk$).
(iii) Explainability via Keywords: We explore the explainability provided by different keyword types associated with each label—unigram, bigram, and trigram. This analysis compares the impact of keyword granularity on GUIDEQ's performance, helping us understand how keyword-based question generation improves the classification process.
(iv) Multi-Turn Interaction: We investigate the potential of multi-turn question generation, where multiple rounds of guided questions are used to iteratively refine the extracted information. This setting evaluates how effectively GUIDEQ handles scenarios where a single question is insufficient to gather all relevant details.

Setup: We fine-tune DeBERTa and BERT-uncased models as the classifier in our framework. Additionally, we use the Llama-3 8B-Instruct model for question generation. We select this smaller parametric model to reduce computational overhead while maintaining robust performance. For each dataset, we split the text instances at the sentence level, dividing them into two equal parts. The first half serves as the partial information provided to the model. The second half, referred to as the "grounded context", is used to extract answers to the generated questions. DeBERTaV3 finetuned on SQuadV2, is employed as the Question Answering model, tasked with extracting the most relevant answer snippets from the grounded contexts.

Skyline Classification					
Datasets	F1-score/ BERT	F1-score/ DeBERTa	Datasets	F1-score/ BERT	F1-score/ DeBERTa
cnews	63.8	65.6	salad	64.5	66.4
dbp	95.5	94.5	stress	43.5	47.7
s2d	99.8	100.0	20NG	75.8	71.6

Table 3: Skyline F1-Scores for complete original text (both partial and reference combined) on BERT and DeBERTa classifier models

Win Rate Scores					
Datasets	LLM	LLM-nk	Datasets	LLM	LLM-nk
cnews	66.0%	67.0%	salad	72.0%	62.0%
dbp	93.0%	92.0%	stress	65.0%	70.0%
s2d	90.0%	85.0%	20NG	89.0%	93.0%

Table 4: Win Rate (WR) % scores of questions generated for (i) GUIDEQ with LLM baseline; (ii) GUIDEQ with LLM-nk baseline

5 Results and Observations

Our work, GUIDEQ, was evaluated on six classification datasets, enabling a rigorous assessment of the framework's ability to generate guided questions from partial information and improve classification accuracy.

The strongest results for GuideQ framework can be seen in table 2, wherein we report the F1 score for classification task post-answering the generated question. The question answering model is such that it extracts the most relevant text from the reference text in respect to the asked question. We observe that for all datasets at all instances (expect for stress with BERT classifier), our method shows the highest overall classification scores (F1 score) across both classifier models. Second, considering results of both classifier models together, table 1 also shows that GUIDEQ always has the highest margin of improvement compared to F1 scores with partial information.

We observe an improvement of 18.6% (BERT) and 22.1% (DeBERTa) for s2d and 22.1% (BERT) and 20.7% (DeBERTa) for salad dataset with our method over partial information. This large shift shows our method is effectively able to frame questions based on previous training data and keywords. This makes the question asked more grounded in context of the partial information. A question maybe raised as to why some datasets show more improvement in F1 score than the others. We observe the skyline F1 score results for each dataset using the complete text on both classifier models (table 3). The skyline results reveal the inherent classification ability of the datasets themselves.

We also observe that in some situations with other baselines, the addition of new answer based on generated question negatively impacts the classification, *i.e.*, reduces F1 score. Example of the same is a drop of 0.4% for dbp dataset with BERT classifier and a drop of 0.2% with DeBERTa classifier using LLM-nk approach. Many instances also include when the baselines show only a slight improvement. However, our method never shows a dip in score compared to partial information classification. The range of percentage gain is also comparatively larger than other methods. This is crucial in realistic situations where we want to ensure that originally correct categorized text is not misclassified.

6 Analysis

6.1 Effect of Classifier Model

We perform our experiments by finetuning two different classifier models - BERT and DeBERTa for a robust analysis. Firstly, GUIDEQ outperforms other methods on both the classifier models in most settings. Only for stress dataset with BERT classifier, a classic LLM based approached worked better and showed 2.7% increase in F1 score. When absolute percentage gain over partial information is taken into consideration for our method, classification using DeBERTa leads for four out of six datasets. BERT shows highest absolute gain for salad and 20NG datasets. We represent the percentage gains in brackets in table 2 and underline the highest gain across a dataset. Although we observe comparable gains for partial information classification along with LLM, and LLM-nk baseline approaches.

A possible reason for the same maybe as follows: DeBERTa being a larger parameter model compared to BERT is able to capture more relevant explainable keywords for a given label using occlusions. This results in a better formed and focused question. The conclusion we can derive is that even though a better classifier model doesn't necessarily show higher F1 scores with partial information, it aids to improve results when combined with GUIDEQ framework leveraging keywords.

6.2 Analysis of baseline approaches

We also observe that the performance of the other two baselines, *i.e.*, the use of only an LLM to ask

	BERT			DeBERTa		
Dataset	uni	bi	tri	uni	bi	tri
cnews	**6.8**	5.6	6.1	7.2	7.1	**8.9**
dbp	**2.5**	1.8	1.6	**6.6**	6.3	6.2
s2d	**19.7**	18.6	17.0	14.6	**22.1**	12.8
salad	20.7	**21.9**	21.3	20.1	**20.7**	20.6
stress	-0.3	0.6	**1.6**	-0.2	**3.0**	-1.5
20NG	**5.8**	5.4	5.0	**3.9**	2.7	3.0

Table 5: Comparison of % absolute gain over partial information F1-Scores for GUIDEQ framework with unigram (uni), bigram (bi), and trigram (tri) keywords

relevant question and providing an LLM with only the top-3 classifier labels perform almost at par with each other. In other words, though we conclude that GUIDEQ shows improved performance over other baselines, the two baselines themselves are comparable to each other in performance. For example, for salad dataset LLM-nk baseline performs better while for s2d dataset only LLM baseline shows higher results. This holds true for both classifier models. While GUIDEQ leverages explainability through keywords along with labels, the LLM-nk baseline uses only labels. Apparently, the labels themselves do not add explicit information that would help guide for completion of the text.

6.3 Quality of Generated Questions

Next, we evaluate the question quality of different baselines. We do this by calculating the win rates using LLM model of question generated with our method taking LLM and LLM-nk as base on a subset of 100 random instances for each dataset. Table 4 summarizes the results for the same. We observe that our method always has a win rate above 50%. The minimum win rate reached is 62.0% for salad datasets with LLM-nk. For three datasets: db, s2d, and 20NG, GUIDEQ performs exceptionally higher. Overall, table 4 shows that our method frames questions which are more relevant and specific to the partial information and unseen reference answer.

6.4 Effect of n-grams in GUIDEQ

In this section we compare the results three different n-gram approaches for keywords generation, namely: unigram, bigram, and trigram, which means the keywords are restricted exactly to be single words, two words, and three words respectively. The results comprising of percentage gain of F1 score over that of partial information are summarized in table 5. We observe that while all the three perform almost similar, still results with questions formed using unigram and bigram are superior in 30/36 total setups as compared to results using trigram. In other words, GUIDEQ slightly performs better when the explainable keywords are restricted to two or three words. Although there is no particular trend between unigram and bigram themselves.

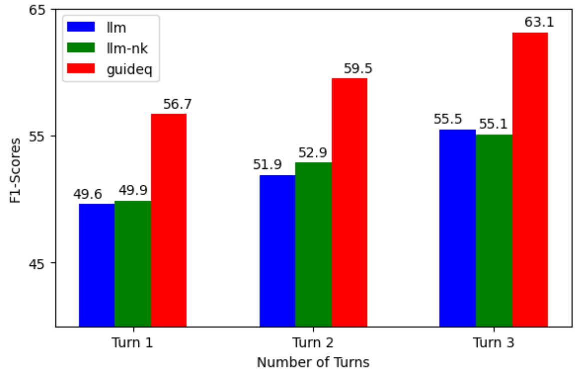

Figure 3: Multiturn Results: F1-Scores for three turn question answering on cnews datatset.

7 Ablation Study: Multi-turn

Our framework explores a multi-turn setting, where successive guided questions are posed following a prior response. We specifically report multi-turn results on the cnews dataset due to its larger context per instance and significant differences in partial information scores versus skyline F1-Scores. The text is divided into three segments, with the first serving as partial information to initiate the guided questioning (GuideQ) process. This method involves dynamically updating the pool of guiding words, removing those already used in previous turns. After generating questions, the answer extraction model derives responses from the residual text. This answer, combined with the initial partial information and the refreshed guiding words, informs the next GuideQ round. This cycle repeats over three turns, continually refining the guiding words to enhance the relevance and depth of information retrieval. Figure 3 illustrates the summarized results, showing our method's superior performance in multi-turn scenarios.

8 Conclusion and Future Work

Our work GUIDEQ, introduces a novel framework designed to generate guided questions that enhance classification accuracy and improve information gathering in scenarios with partial data. By using explainable AI, GUIDEQ combines keywords identified from classifier models with LLM-based question generation to create guided questions. This approach shows better performance on multiple datasets compared to other baseline methods. Our results show consistent improvements in F1 scores, with gains of up to 22% on certain datasets. GUIDEQ generate high-quality, context-relevant questions is evident from the win rates against baseline methods. GUIDEQ's effectiveness in multi-turn interactions and its flexibility in accommodating different n-gram approaches for keyword generation further underscore its potential for real-world applications in information retrieval and classification tasks.

We focus exclusively on open-source LLMs like Llama-3 8B-Instruct, which strike an excellent balance between performance and computational efficiency. Our approach highlights the potential of open and accessible models to drive innovation without the heavy resource demands of proprietary alternatives.

9 Limitations

Despite GUIDEQ's promising results, several limitations should be noted. First, the framework's performance is dependent on the quality of the initial classifier model and the relevance of extracted keywords. Suboptimal classifier training or keyword selection could lead to less effective question generation. Secondly, the framework's reliance on LLMs for question generation also introduces potential biases and inconsistencies inherent to these models. Finally, the computational resources required for running large language models may pose scalability challenges in certain applications.

For evaluating question quality, we recognize the inherent challenges of obtaining domain expert human evaluations across diverse fields such as healthcare, finance, and news. However, the win rate metric serves as a systematic and practical tool for comparison, offering meaningful insights even in the absence of gold-standard labels. This metric enables us to effectively evaluate generated questions across a variety of applications.

10 Acknowledgments

We thank the reviewers for their constructive feedback. Suraj Racha is funded and supported by the Prime Minister Research Fellowship (PMRF) for his Ph.D. work. This research was supported by and is part of BharatGen, an Indian Government-Funded Multimodal Large Language Model initiative for Indian Languages.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz 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. *ArXiv*, abs/2005.14165.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Haihua Chen, Lei Wu, Jiangping Chen, Wei Lu, and Junhua Ding. 2022. A comparative study of automated legal text classification using random forests and deep learning. *Inf. Process. Manag.*, 59:102798.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning.

Joseph Enguehard. 2023. Sequential integrated gradients: a simple but effective method for explaining language models.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *ArXiv*, abs/2208.01066.

E. Hassan, T. Abd El-Hafeez, and M.Y. Shams. 2024. Optimizing classification of diseases through language model analysis of symptoms. *Scientific Reports*, 14:1507.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Qiao Jin, Robert Leaman, and Zhiyong Lu. 2023. Retrieve, summarize, and verify: How will chatgpt affect information seeking from the medical literature? *Journal of the American Society of Nephrology : JASN*, Publish Ahead of Print.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution.

Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Unifying question answering, text classification, and regression via span extraction. *arXiv: Computation and Language*.

R. Kreesh. 2024. Human stress prediction dataset. Accessed: 2024-02-06.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. 2024. Post hoc explanations of language models can improve language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Nicholas Kroeger, Dan Ley, Satyapriya Krishna, Chirag Agarwal, and Himabindu Lakkaraju. 2024. In-context explainers: Harnessing llms for explaining black box models.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S. Auer, and Christian Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models.

Yingcong Li, M. Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: generalization and stability in in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Maria Moise, Ciprian Gheorghe, and Marilena Zingale. 2010. Developing question answering (qa) systems using the patterns. *WSEAS Transactions on Computers archive*, 9:726–737.

Linyong Nan, Ellen Zhang, Weijin Zou, Yilun Zhao, Wenfei Zhou, and Arman Cohan. 2023. On evaluating the integration of reasoning and action in llm agents with database question answering. *ArXiv*, abs/2311.09721.

Oliviervha. 2023. Crypto news.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad.

Vishwas Sathish, Hannah Lin, Aditya K Kamath, and Anish Nyayachavadi. 2024. Llempower: Understanding disparities in the control and access of large language models.

Yuxuan Sun, Kai Zhang, and Yu Su. 2023. Multimodal question answering for unified information extraction. *ArXiv*, abs/2310.03017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023a. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *ArXiv*, abs/2308.13259.

Yu Wang, Yuan Wang, Zhenwan Peng, Feifan Zhang, Luyao Zhou, and Fei Yang. 2023b. Medical text classification based on the discriminative pre-training model and prompt-tuning. *Digital Health*, 9.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Alexandra Zytek, Sara Pido, and Kalyan Veeramachaneni. 2024. Llms for xai: Future directions for explaining explanations. *ArXiv*, abs/2405.06064.

A More dataset details

We use six different NLP classification datasets to demonstrate the effectiveness of our framework. Each of the six focuses on a different domain, allowing for robust testing.

s2d: The Symptom2Disease (s2d) dataset challenges the model's ability to classify diseases based on incomplete symptom descriptions, which is critical in medical decision-making.

cnews: The Crypto News dataset assesses the framework's adaptability to rapidly evolving financial data.

stress: The Human Stress Prediction dataset evaluates the model in a psychological context, where input data is often sparse or incomplete.

20NG: The 20 Newsgroups dataset, with its wide range of discussion topics, tests the model's generalizability across various themes.

dbp: The DBpedia dataset challenges the model with a broad spectrum of structured factual information, which is essential for handling real-world knowledge-based queries.

salad: The Salad-Bench dataset is a novel benchmark designed for evaluating LLMs in safety, defense, and attack scenarios.

B Prompts used

Primarily, we use two different prompting baselines along with the GuideQ framework. 'LLM' is the baseline where the model alone generates the clarifying question without external information. The only input is the partial information. Secondly, LLM-nk is the baseline where LLM sees the partial information and the top-3 categories to choose from without exposure to the guiding keywords. Finally, GuideQ leverages both top labels and guiding keywords corresponding to each of them for the formation of clarification questions.

We present the prompts corresponding to each of them in figure 4 , figure 5, figure 6, for LLM, LLM-nk, and GuideQ respectively.

The few shot examples used are as follows. Each time we take increasing inputs of partial information, categories, and guiding keywords:

Example 1:
Partial information: I constantly sneeze and have a dry cough.

Category: Allergy
keywords: headache, coughing, wet, sneeze, pain
Category: Diabetes
keywords: severe, feet, skin, rashes, infection
Category: Common Cold
keywords: swollen, cough, body, shivery, ache, dry

QUESTION: "Besides fever, are you experiencing symptoms such as cough, severe headaches, localized pain, or inflammation? Also, can you describe the pattern of your fever—is it continuous or does it occur in intervals?"

Explanation (GuideQ): Sneeze and dry cough are the main subjects of the partial information. Coughing is present in Allergy and common cold, but cough or sneeze is not present in Diabetes. Therefore, Diabetes can't be a possible label. Only two labels—Allergy and Common Cold—are considered. The keywords suggest that knowing about symptoms like headache, body pain, shivery, etc., will help refine the classification into one of the labels.

Example 2: Partial information: The software keeps crashing.

Category: Software Bug
Keywords: crash, error, bug, glitch
Category: User Error
Keywords: instructions, setup, incorrect, usage
Category: Hardware Issue
Keywords: overheating, components, failure, malfunction

Explanation: The main subject of the partial information is the software crash. The keyword 'crash' is directly related to Software Bug but could also be indirectly related to User Error and Hardware Issue. However, to differentiate, asking about the conditions under which the crash happens or if any error messages appear could help narrow down the correct category.

QUESTION: "When the software crashes, do you receive any specific error messages, or does it happen during particular tasks? Have you noticed any hardware malfunctions or overheating before the crashes?"

Example 3:
Partial information: The car is making a strange noise.

Category: Engine Problem
Keywords: noise, misfire, engine, smoke
Category: Tire Issue
Keywords: flat, noise, pressure, alignment
Category: Transmission Issue
Keywords: shifting, noise, gears, slipping

Explanation: The main subject of the partial information is the strange noise. The keyword 'noise' is present in all three categories—Engine Problem, Tire Issue, Transmission Issue. Knowing more about the type of noise and when it occurs can help identify the correct category.

QUESTION: "Can you describe the noise in more detail? Is it a grinding, squealing, or clicking sound? Does it happen while driving, when shifting gears, or when the car is stationary?"

C Examples

C.1 Generated Questions

Table 8 shows examples of generated questions using various baselines as well as our GuideQ framework for a comparative analysis. In the example shown, we observe that guiding keywords help reach the second half better. is highly effective in addressing the user's symptoms. It specifically targets the brownish, stringy phlegm they described, key to diagnosing pneumonia, and explores critical symptoms like chest tightness and breathing difficulty. Additionally, it inquires about the duration of symptoms, providing important context for assessing illness progression. In contrast, question-LLM and questionnk are more general and do not focus on the phlegm's characteristics, which are essential for diagnosing pneumonia. While they ask about respiratory infections and symptoms like wheezing and chest tightness, they are less detailed and may result in vague responses, missing crucial diagnostic information.

C.2 Generated Keywords

Table 9 shows examples of generated keywords based on one-gram, two-gram, and three-gram occlusion methods. From a qualitative observation, we note that two-gram is can be an optimum approach to capture relevant information. While one-gram may miss important information, three-gram can have the tendency to in-cooperate irrelevant information.

D Further Results of classification

We further report the recall and precision scores for the main classification task following table 1 results. The recall and precision values show a similar trend as that of F1-score in table 1. We report precision scores in table 6 and recall in table 7. GuideQ shows significant improvement over other baselines in most of the situations.

You are an AI expert. You are provided with a partial information. Your task is to ask an information-seeking question based on the partial information such that when answered, one of the categories to which it belongs to can be selected with confidence.

Follow the following thinking strategy:
- Identify the main context of the partial information.
- Now generate a question. This question should further probe for information that will help refine the understanding of the most likely situation.
- Generate both the thought process and the question. Strictly follow the format shown in examples for output generation. Double quote the final question.

Here are a few examples to understand better:
<example 1>
<example 2>
<example 3>

Now generate note and QUESTION for:
{partial information}

{categories}

Output Question:

Figure 4: LLM prompt baseline

You are an AI expert. You are provided with a partial information statement along with the top-3 categories where this information could belong. Your task is to ask an information-seeking question based on the partial information such that when answered, one of the categories to which statement belongs to can be selected with confidence.
Follow the following thinking strategy:
- First, eliminate the labels that are not probable for the given information.
- Identify the main context of the partial information and see if a similar content matches in any of the label. If it doesn't, then the label can be taken out of consideration.
- Now generate a question for collecting further information. This question should further probe for information that will help refine the identification of the most likely diagnostic category. Generate both the thought process and the question.
- Generate both the thought process and the question. Strictly follow the format shown in examples for output generation. Double quote the final question.

Here are a few examples to understand better:
<example 1>
<example 2>
<example 3>

Now generate note and QUESTION for:
{partial information}

Output Question:

Figure 5: LLM-nk prompt baseline

	BERT Classifier Model				DeBERTa Classifier Model			
Dataset	partial	LLM	LLM-nk	GuideQ	partial	LLM	LLM-nk	GuideQ
cnews	0.489	0.514	0.523	0.545	0.568	0.566	0.574	0.603
dbp	0.943	0.934	0.926	0.913	0.926	0.919	0.919	0.932
s2d	0.625	0.757	0.722	0.805	0.697	0.763	0.757	0.793
salad	0.414	0.584	0.607	0.622	0.447	0.613	0.611	0.627
stress	0.323	0.352	0.338	0.330	0.471	0.426	0.468	0.421
20NG	0.697	0.699	0.700	0.748	0.659	0.674	0.688	0.704

Table 6: Precision results for GuideQ

You are an AI Expert. You are provided with partial information along with the top-3 categories where this information could belong. Each category also has a list of keywords that represent the characteristic content covered by the category. Your task is to ask an information-seeking question based on the partial information and the category keywords such that when answered, one of the categories can be selected with confidence.

Follow the following thinking strategy:
- First, eliminate the categories that are not probable based on the given information.
- Identify the main context of the partial information and see if similar content matches any of the keywords in a category. If it doesn't, then the category can be taken out of consideration.
- Now generate a question. This question should further probe for information that will help refine the identification of the most likely category.
- Your question should strategically use the keywords tied to each potential category, aiming to effectively differentiate between them.
- Generate both the thought process and the question. Strictly follow the format shown in examples for output generation. Double quote the final question.

Here are a few examples to understand better:
<example 1>
<example 2>
<example 3>

Now generate note and QUESTION for:
{partial information}
{categories}: {keywords}

Output Question:

Figure 6: GuideQ prompt

	BERT Classifier Model				DeBERTa Classifier Model			
Dataset	partial	LLM	LLM-nk	GuideQ	partial	LLM	LLM-nk	GuideQ
cnews	0.436	0.469	0.484	0.496	0.418	0.440	0.451	0.495
dbp	0.838	0.846	0.845	0.885	0.820	0.825	0.825	0.912
s2d	0.653	0.729	0.681	0.799	0.667	0.722	0.688	0.806
salad	0.358	0.547	0.562	0.571	0.374	0.556	0.561	0.591
stress	0.339	0.364	0.347	0.368	0.427	0.417	0.424	0.433
20NG	0.678	0.688	0.685	0.730	0.641	0.649	0.648	0.679

Table 7: Recall results for GuideQ

Context
First Half: I can’t seem to catch my breath and I’m sweating a lot. I feel really sick and have a lot of phlegm in my throat. **Second Half as Reference:** My chest hurts and my heart is racing. The mucus I’m coughing up is brownish and stringy.
LLM-nk: Have you experienced any recent exposure to airborne pathogens, such as a viral outbreak at work or school, or have you been around someone with a similar illness? Are there any other symptoms you’re experiencing, such as chest tightness, wheezing, or a productive cough?
LLM only: Have you recently been exposed to anyone with a respiratory infection, or have you been in a crowded or poorly ventilated area? Are you experiencing chest tightness, wheezing, or any other symptoms besides difficulty breathing, sweating, and excessive phlegm?
GuideQ: Can you tell me more about the phlegm you’re coughing up? Is it yellow, green, brown, or red? Are you experiencing any chest tightness or pain? And do you have any difficulty breathing when you’re resting or exerting yourself?

Table 8: Example of generated question using various baseline approaches along with GuideQ.

Context
First Half: I can’t seem to catch my breath and I’m sweating a lot. I feel really sick and have a lot of phlegm in my throat. **Second Half as Reference:** My chest hurts and my heart is racing. The mucus I’m coughing up is brownish and stringy.
1-gram: <’pressure.’, ’sinuses’, ’cold’, ’eyes’, ...> <’phlegm’, ’cough,’, ’coughing’, ’throat’, ...> <’nauseous,’, ’affects’, ’eyes’, ’chest’, ...>
2-grams: <’weak, and’, ’My eyes’, ’My sinuses’, ’keep sneezing’, ...> <’red phlegm.’, ’time breathing.’, ’some brownish’, ’brownish mucous’, ...> <’occasionally get’, ’have nighttime’, ’can occasionally’, ’occasionally flake’, ...>
3-grams: <’days. My sinuses’, ’cough. I ve got’, ’chills and a’, ’sinuses are congested’, ...> <’and red phlegm’, ’brownish phlegm coming’, ’difficulty breathing I m’, ’phlegm I m coughing’, ...> <’night I get’, ’occasionally flake. My’, ’breathing Sometimes at’, ’difficulty breathing Sometimes’, ...>

Table 9: Example of various guiding keywords generated using different n-gram methods.