

# Answering Unanswered Questions through Semantic Reformulations in Spoken QA

Pedro Faustini<sup>1\*</sup>, Zhiyu Chen<sup>2</sup>, Besnik Fetahu<sup>2</sup>, Oleg Rokhlenko<sup>2</sup>, Shervin Malmasi<sup>2</sup>

<sup>1</sup>Macquarie University, Sydney, NSW, Australia

<sup>2</sup>Amazon.com, Inc., Seattle, WA, USA

pedro.arrudafaustini@hdr.mq.edu.au {zhiyu, besnik, oleg, malmasi}@amazon.com

## Abstract

Spoken Question Answering (QA) is a key feature of voice assistants, usually backed by multiple QA systems. Users ask questions via spontaneous speech which can contain disfluencies, errors, and informal syntax or phrasing. This is a major challenge in QA, causing unanswered questions or irrelevant answers, and leading to bad user experiences. We analyze failed QA requests to identify core challenges: lexical gaps, proposition types, complex syntactic structure, and high specificity. We propose a Semantic Question Reformulation (SURF) model offering three linguistically-grounded operations (repair, syntactic reshaping, generalization) to rewrite questions to facilitate answering. Offline evaluation on 1M unanswered questions from a leading voice assistant shows that SURF significantly improves answer rates: up to 24% of previously unanswered questions obtain relevant answers (75%). Live deployment shows positive impact for millions of customers with unanswered questions; explicit relevance feedback shows high user satisfaction.

## 1 Introduction

Question Answering (QA) is a longstanding NLP task, and voice assistants like Alexa have made Spoken QA ubiquitous. Users often address such assistants with spontaneous speech, as they would a human. However, differences between spoken and written language (Chafe and Tannen, 1987), such as the presence of disfluencies, informal or incomplete speech, and different syntax have been shown to pose challenges for NLP tasks (Ward, 1989; Shriberg, 2005; Salesky et al., 2019). QA systems mostly use written data, and such phenomena impact question understanding and answer retrieval (Gupta et al., 2021), leading to irrelevant answers or unanswered questions, leaving users unsatisfied.

Recently, language generation has been used to improve QA through Question Rewriting (QR). For example, QR is used in conversational systems

\*Work done during an internship at Amazon.

Operation	Example
Repair Operation (disfluencies, syntax, formality, lexical gaps)	Q1: <b>does strawberries no i mean</b> blueberries grow on top of tree R1: do blueberries grow on trees
	Q2: how to store honey so <b>it doesn't get weird</b> R2: how to keep honey <b>fresh</b>
	Q3: <b>my</b> hamburger patty <b>is in the fridge for four days should i throw it</b> R3: <b>how long can</b> hamburger patty <b>be refrigerated</b>
Root Transformation Operation (reshape complex questions)	Q4: if you fall and got a bruised thigh should i put ice on it or heat R4: what is the <b>treatment</b> for a bruised thigh
	Q5: <b>our pet</b> dog <b>was playing in the park and ate</b> a rat <b>is that safe</b> R5: <b>what happens if a</b> dog <b>eats</b> a rat
	Q6: was winston churchill's mother an american nurse R6: <b>who</b> was winston churchill's mother
Generalize Operation (relax/remove constraints, modify entities)	Q7: did kamala harris move to canada <b>then back to america</b> R7: did kamala harris move to canada
	Q8: what is the maximum salary of a plumber in <b>san francisco</b> R8: what is the income of a plumber in <b>california</b>
	Q9: how do i get rid of flies <b>that keep come out of the bathroom</b> R9: how to get rid of flies

Figure 1: Examples of challenging questions (Q) and our proposed reformulation operations (R) on them.

to answer contextual questions in multi-turn dialogues (Ye et al., 2022). While QA models can be improved with fine-tuning, real-world systems have multiple QA backends and retraining is expensive, making input rewriting a practical solution (Chen et al., 2022). This has the added benefit that a single QR model may improve multiple QA systems.

We propose applying QR to reformulate difficult or unanswered questions. We analyzed millions of answered and unanswered real-world questions from a leading voice assistant to understand the factors impacting QA failure (§3). In addition to the well-known issue of disfluencies, we identify novel challenges from question structure and specificity. To address them, we propose three linguistically-informed reformulation operations that only require the question (§4). The operations, shown in Figure 1, are designed to *improve answerability*<sup>1</sup> based on common speech patterns, so that for a previously unanswered question, the same QA system is able to provide an answer for its reformulation.

<sup>1</sup>We consider a question answerable if a QA system decides with sufficient confidence that a suitable answer is retrieved.

While question repair has been studied, our *root wh-* and *question generalization* operators are novel contributions of this work. Our results demonstrate that our approach can achieve:

1. high reformulation accuracy of 83% for rewriting questions to a desired shape (§6.1);
2. improving the answer rate of previously unanswered questions by up to 24% (§6.2); and
3. 75% of answers on reformulated questions are relevant to the original question (§6.3).

Live deployment of our model (§6.4) achieves positive impact for millions of users with unanswered questions, and explicit relevance feedback from customers shows high satisfaction.

## 2 Related Work

**Question Quality:** QA models are typically trained on formal written language, and are known to be impacted by the quality of user questions. An analysis of the WikiAnswer dataset (Fader et al., 2014) by Liu et al. (2019) showed that 68% of the questions were ill-formed, usually due to wrong words, wrong order, or background noise, harming the answerability of those questions. Gupta et al. (2021) examined the impact of disfluencies in QA, showing that they had a large impact on answering performance. Many of these issues stem from natural properties of spontaneous speech, such as errors, self corrections, and informal syntax (Chafe and Tannen, 1987). Our work tackles these issues, and tries to go beyond corrections by considering question types and question specificity.

**Question Complexity:** Depending on the QA system, some questions may be more difficult to answer. It has been shown that questions requiring multi-hop reasoning are more challenging (Yang et al., 2018), often leading to no answers or wrong answers. Questions are affected by the broader types of syntactic complexity explored in the field (Nassar et al., 2019; Martin et al., 2020; Sheang and Saggion, 2021). Regardless of complexity, questions may also be unanswerable due to incorrect framing or false suppositions (Kim et al., 2021). Other work has analyzed questions in different datasets, showing that *wh*-\* words (e.g. *who*, *what*, *when*) are the dominant way to start a question (Ko et al., 2020), and that these words and related phrases (e.g., “*how much*”, “*how large*”) are associated with reduced answering complexity (Chali

and Hasan, 2012). In our work we consider how controlled syntactic restructuring can address the above challenges to reduce answering complexity.

**Question Rewriting:** Rewriting questions is a natural extension of query reformulation approaches used to improve Information Retrieval (He et al., 2016). Question rewriting has been applied to improve QA in different ways. Question paraphrasing has been used as a data augmentation approach to retrain QA systems to improve robustness (Gan and Ng, 2019). Buck et al. (2018) propose using a reinforcement learning agent between the original question and a black box QA system. The agent probes the QA system with several reformulations to learn how to elicit the best answer. Liu et al. (2019) propose a question refinement system to rewrite malformed questions.

**Rewriting Operations:** Text rewriting is based on specific linguistic changes. Nassar et al. (2019) note that text simplification changes can be lexical (rare words replaced by more common ones) and syntactic (complex structures are split, reordered, or deleted). Tomuro (2003) notes that paraphrasing questions is more difficult as the interrogative structure is separate from the declarative, and can have many variations. They quantified paraphrasing operations and showed that interrogative reformation accounted for 50% of changes, followed by lexical substitution (25%) and semantic changes (16%). Recent work on sentence rewriting has followed this direction, by breaking down reformulation into predefined editing operations (Choi et al., 2021).

Our work is inspired by all of the above, but differs in several ways. We expand on the known issues in QA by analyzing real voice assistant data to identify prevalent challenges to tackle; we consider malformed question correction as a prerequisite for dealing with challenges of complex questions. Additionally, prior rewriting approaches aim to improve QA via retraining, or by building a rewriter tailored to a single QA system. We take a different approach that does not rely on answer data or QA system feedback, and build a general model that can benefit multiple QA systems in a federated architecture. Instead of uncontrolled paraphrasing, we deal with question complexity via controllable reformulations that distinguish between lexical modification, interrogative clause restructuring, and semantic changes. We propose novel linguistic restructuring operations to deal with complex syntax, and generalize high-specificity elements.

### 3 Challenges in Real-world Spoken QA

First, to quantify and understand why spoken QA fails, we perform a failure analysis on 10 million real questions, by further distinguishing questions according to their *question type* (we define 5 types based on linguistic properties, see Appendix B for details) from a leading voice assistant.

**Scope:** we limit our work to questions that were not answered due to retrieval failure, but may potentially have relevant answers if reformulated. They must be valid questions (seek knowable knowledge) whose information need can be understood (by humans) and re-stated. QA may fail for other reasons; we do not consider such issues e.g., *inter alia*, ASR errors, invalid or difficult to understand questions, subjectivity, and other reasons for retrieval failure.

A quantitative and qualitative study was undertaken by domain experts (details in Appendix A), and identified the below challenges (C1-7) as contributing to a significant proportion<sup>2</sup> of failed requests, and potentially solvable by reformulation.

**C1. Malformed Utterances:** Questions with disfluencies and syntactic errors were more likely to fail e.g., Fig. 1 (Q1). Correction methods have previously been used to fix these (Gupta et al., 2021).

**C2. Lexical Gaps** Questions framed colloquially or lacking appropriate parlance for a topic e.g., Fig. 1 (Q2), were associated with failure. This is caused by lexical gaps (Riezler and Liu, 2010) arising from language mismatch between the user input and answer sources, as QA systems use formal knowledge sources for retrieval. Lexical substitution and rephrasing may address this challenge.

**C3. Complex Syntactic Structure:** Utterances with complex structure, such as multi-clause questions, can lead to QA failure. Such phrasing is more common in spoken language, and can be simplified via syntactic restructuring, e.g., Fig. 1 (Q3-5).

**C4. Polar Propositions:** Yes-No questions are asked to confirm a specific proposition, e.g., “*Do box turtles live in Japan?*”. Answering polar questions is more difficult than wh-questions for both humans (Moradlou et al., 2021) and QA systems (Clark et al., 2019), due to the entailment and inferences required to arrive at an answer. This can be simplified by reformulating to a factoid wh-question, e.g., “*Where do box turtles live?*”.

**C5. False Presuppositions:** Additionally, polar questions may contain false presuppositions that cause retrieval failure (Kim et al., 2021). We hypothesize rewriting such questions to wh-questions may retrieve relevant answers, e.g., Fig. 1 (Q6).

**C6. High Specificity:** Highly specific questions (concerning very specific entities, or conditions) may not be answerable. We believe generalizing such questions by entity modification or constraint relaxation (Fig. 1 Q7-9) can broaden answer recall.

**C7. Irrelevant Info:** Related to C3 and C6, complex and high-specificity questions may contain contextual facts that are irrelevant to the answer. We believe removing such details can improve answer recall (Fig. 1 Q4/5/7).

### 4 SURF Question Reformulation Model

We now describe our proposed **Semantic Question Reformulation model (SURF)** and the reformulation operators that it supports.

#### 4.1 Reformulation Model

Inspired by controllable multi-task learning for text generation (Keskar et al., 2019; Raffel et al., 2020), we train a single model to perform different reformulations. Our reformulation model,  $\mathcal{F}(p, q)$ , represents a seq2seq Transformer model (Lewis et al., 2020), and is trained such that for an *input* question  $q$  and a *target* reformulation operator  $p \in \{\text{REP, ROO, GEN}\}$ , pre-pended as a prefix to  $q$ , it reformulates  $q$  into  $q'$  according to  $p$ .

**Model Training.**  $\mathcal{F}$  is trained in two stages: the first stage pretrains  $\mathcal{F}$  using a large *weakly-supervised* corpus (derived by a heuristic proposed in §5.3) of  $\langle q, q' \rangle$  for the REP and ROO operations. In the second stage, we finetune  $\mathcal{F}$  on manually annotated pairs of  $\langle q, q' \rangle$  for all operators in  $p$ .

#### 4.2 Reformulation Operators

Each prefix  $p$  instructs  $\mathcal{F}$  to perform a specific type of reformulation. We define the following prefix operators based on the challenges presented in §3.

**Question Repair (REP):** To address challenges C1-2, REP removes disfluencies, performs syntactic correction, and increases formality via lexical substitution with high-entropy words. For example, the input “*Where can I get a booze after 11 pm?*” is repaired to “*Which stores sell beer after 11 pm?*”.

<sup>2</sup>The exact numbers cannot be divulged for confidentiality.

**Root Wh- Transform (ROO):** Outlined in challenges C3-6, questions with complex structure are more difficult, but may be answered if simplified to factoid questions. ROO reformulates  $q$  such that the interrogative *wh*-\* phrase is clause initial (at the root of the sentence), making needed syntactic adjustments. For example, “*Do any universities in Germany offer degree programs taught in English?*” → “*Which universities in Germany offer degree programs in English?*”. This also handles contextualized multi-clause questions, e.g., “*I am chopping onions for a pizza dinner how fine should they be*” → “*how fine should onions be for pizza*”.

**Question Generalization (GEN):** To deal with highly specific questions, covered in challenge C4, we propose a novel question generalization operation. Inspired by similar approaches to improve recall in structured query languages (Motro, 1984) and IR (Boldi et al., 2011), we simplify questions through the removal or relaxation of semantic constraints. Creating a more general question allows the retrieval of a superset of results, which in many cases provides a highly related answer that may be better than no answer. GEN does this by dropping adjuncts, replacing nouns with hypernyms or holonyms, and removing adjectives. For example, the question “*Do poisonous pythons live in Miami?*” can be generalized to “*Do snakes live in Florida?*”. Note that “*python*” and “*Miami*” are turned into more generic entities “*snake*” and “*Florida*”, and at the same time the aspect “*poisonous*” is dropped.

In ROO and GEN, REP is always performed jointly with the respective operators. The output of all operators should not contain any syntactic or semantic errors present in the original question.

## 5 Experimental Setup

### 5.1 Intrinsic Evaluation Strategy

Using a human study, we intrinsically evaluate the *reformulation accuracy*<sup>3</sup> to assess if: (1) the reformulation *retains* the intent of the input question; and (2) the reformulation *satisfies* the properties of the reformulation operator  $p$ .

**Evaluation Data:** For each question type, we randomly sampled 50 questions from each reformulation operator. This data is then assessed by expert annotators, resulting in 1,000 annotations.

<sup>3</sup>Note that due to the unreliability of automated metrics such as BLEU and ROUGE, we opt for human evaluation and answering metrics in our intrinsic and extrinsic evaluations.

### 5.2 Extrinsic Evaluation Strategy

For the extrinsic evaluation, we assess the impact of the reformulated questions on two aspects:

**Answer Rate:** measured as the percentage of reformulations that obtain an answer.

**Answer Relevance:** a three-point scale measuring the answer relevance to the original question (obtained from the reformulated questions): **Irrelevant** (0): answer is not related to  $q$ ; **Related** (1): answer is partially relevant;<sup>4</sup> and **Exact** (2): answer exactly satisfies question’s information need.

**Evaluation Data:** For the two aspects we measure, we consider the following evaluation datasets:

- **Answer Rate:** We randomly sampled 1M *unanswered* questions by our QA system (see Appendix F for additional details).
- **Answer Relevance:** on the same questions used for intrinsic evaluation, the annotators also check the answer relevance w.r.t the original question.

### 5.3 Training Data

**Pre-training Data.** We create a weakly-supervised dataset of 1.2M samples, derived from the MQR corpus (Chu et al., 2020), which provides tuples of ill-formed and well-formed questions (c.f. §D). To construct input tuples  $\langle p, q \rangle$  for pre-training  $\mathcal{F}$ , from a target question  $q'$  we derive  $p$  as follows. First, using the algorithm in Appendix B, we identify the question types of  $q$  and  $q'$ . If  $q$  and  $q'$  have the same type, then  $p = \text{REP}$ . If  $q'$  is a *root* question and  $q$  is not, then  $p = \text{ROO}$ . The GEN operator is novel to our work and cannot be automatically derived, and is part of the fine-tuning dataset.

**Fine-Tuning Data.** We sampled 3,851 questions and annotated reformulations, based on guidelines listed in §E, for all operators in §4.2. We use 10% of annotated data for validation; the rest is used during the second stage of training to fine-tune  $\mathcal{F}$ .

### 5.4 Reformulation Model Configurations

**SURF:** At inference time, our model<sup>5</sup> can do different reformulations based on  $p$ . We analyze the impact on question answerability from different reformulation operators REP, ROO and GEN. Additionally, we analyze the combination of ROO and GEN, i.e.,  $q$  is first reformulated by ROO, then the resulting  $q'$  is reformulated by GEN, denoted as

<sup>4</sup>e.g., it answers the question’s intent but the subjects may be different, as may be the case of entities in GEN.

<sup>5</sup>Appendix C shows training and hyperparameter details.

ROO+GEN. Note that the operators ROO, GEN, ROO+GEN, all perform a REP operation as well (see Section 4.2 for details).

**Baseline:** As a baseline model we consider an ablation of SURF without its pre-training stage, assessing its performance on the same four operators.

**OPTIMAL:** We consider the case where  $q$  is answered if *any* of its reformulations  $p \in \{\text{REP}, \text{ROO}, \text{GEN}\}$  obtains an answer. OPTIMAL represents the upper bound performance of the QA system.<sup>6</sup>

## 6 Results and Discussion

We now turn to a discussion of the results for the *intrinsic* (accuracy) and *extrinsic* (answer rate and relevance) evaluation strategies.

### 6.1 Intrinsic Evaluation

Task	Accuracy	Answer Relevance		
		Irrelevant	Related	Exact
ROO+GEN	77%	26% (4.6%)	25% ( <b>4.5%</b> )	48% ( <b>8.6%</b> )
GEN	<b>83%</b>	29% (4.2%)	22% (3.2%)	49% (7.0%)
ROO	73%	36% (4.7%)	18% (2.4%)	46% (6.1%)
REP	80%	26% (2.5%)	15% (1.4%)	59% (5.5%)

Table 1: Evaluation results from the human study on *reformulation accuracy* and *answer relevance*. For answer relevance, in brackets are shown the *extrapolated estimations* of the absolute percentages of answered questions from Table 3 and their respective answer relevance. ROO+GEN obtains the highest answer rate and relevance with 13.1% or 131k questions.

Table 1 shows the human evaluation results for reformulation accuracy. The best accuracy is achieved for GEN, with 83% of the reformulations being accurate. This is because GEN does not require changing the question type like ROO.

REP achieves second best accuracy. One reason for the slightly lower accuracy than GEN, is that it sometimes changes the question type (e.g. request to root), which goes beyond the REP’s reformulation scope. Although according to our intrinsic evaluation strategy such cases represent inaccurate reformulation, in practice this is benign as QA systems perform very well on root factoid questions.

Finally, we note that reformulations significantly shorten the input questions and result in higher type-token ratio (Appendix H). We list many examples of model input/output pairs in Appendix I.

<sup>6</sup>Live deployment latency requirements prohibit producing all possible reformulations and running through a QA system; therefore we try to determine the best single operator  $p$  offline.

Task	Baseline	SURF
No reformulation	0.00%	0.00%
REP	8.10%	9.41%
ROO	9.26%	13.18%
GEN	13.29%	14.34%
ROO+GEN	14.50%	17.92%
OPTIMAL	18.48%	24.15%

Table 2: Results for the baseline and SURF models using different reformulation types on our test set.

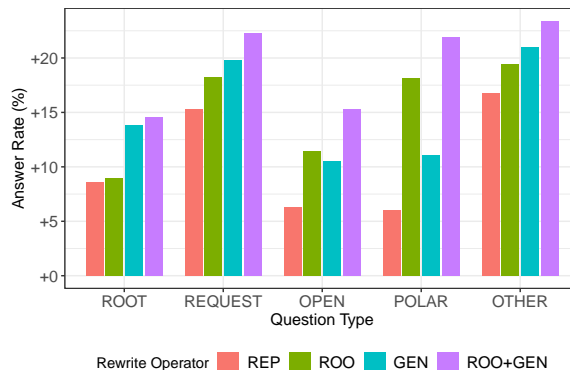


Figure 2: Answer rate of different reformulation tasks grouped by original question types.

### 6.2 Extrinsic Answer Rate Results

Table 2 shows results for all reformulation tasks and models. OPTIMAL represents the case where for an input question at least one reformulation operator gets answered by the QA system. Pre-training yields consistent improvement in all tasks. Our large weakly supervised  $\langle q, q' \rangle$  data enables learning the REP and ROO operations, leading to an answer rate improvement for SURF-ROO with 13.18% over the Baseline of 9.26% (a 3.92% absolute improvement). Figure 2 shows a breakdown of the impact of the operators by question type.

**Impact of Speech Errors:** the REP operation, which performs correction and makes question more formal, shows a consistent answer rate improvement across all tasks and models, improving it by 9.4%. This demonstrated that for many questions speech errors and framing cause retrieval failure. In Figure 2 we note that REP provides a consistent improvement across all question types. This improvement is intuitive given that a core component of QA systems is their ability to understand questions before answering, hence any speech or syntactic errors negatively impact answering.

**Impact of Root Transformation:** the ROO operation repairs and reformulates the question to its root form. It shows better performance than REP, although it may change the original question type. For SURF, the improvement of ROO over REP are with 3.77%, contrary for baseline where the improvement is only 1.16%. This further highlights the importance of the pre-training stage for SURF. Figure 2 shows that for all question types, reframing them as root questions significantly improves the answer rate. ROO is the most effective operator for polar questions, as they are particularly hard to answer (§3). For example, “*Is Sherlock Holmes a real person?*” can also be answered via the alternative question “*Who is Sherlock Holmes?*”.

**Impact of Generalization:** the GEN operation repairs and generalizes the original question to be less specific. For SURF, GEN obtains 4.93% absolute improvement over REP in terms of answer rate, similar is improvement for the baseline with 5.19% (cf. Table 2). As we show in §6.3, most of the provided answers to the generalized questions are in fact relevant to the original question’s intent.

**Impact of Joint Reshaping and Generalization:** ROO+GEN achieves the best performance across all tasks. This is intuitive as questions are first corrected for possible errors, then converted into a root wh- structure, after which high specificity elements are dropped to construct a more generic question (cf. Figure 1, Q7, Q8, Q9). SURF-ROO+GEN only has an 8% gap to the OPTIMAL performance. Figure 2 shows that for all question types, ROO+GEN obtains the highest improvement in answer rate.

Comparing the answer rates of ROO+GEN and OPTIMAL we make an interesting observation: although ROO+GEN combines all operators in  $p$ , its answer rate is still lower than OPTIMAL. This shows that applying all operators is not desirable for all questions. However, in practical settings, processing questions separately with all operators is not feasible due to the induced generation and QA latency. Hence, our proposed solution represents a trade-off between deployment feasibility and improvement in answer rate.

### 6.3 Answer Relevance Results

It is important to consider if the provided answers to previously unanswered questions are relevant to the user’s information need. Since SURF performs numerous syntactic and semantic changes, there is a risk that the reformulated questions will result in

answers that are not related to the user’s intent.

Table 1 shows the answer relevance results for the different operators based on a human study where answers are assessed for their relevance to  $q$ .

REP has the highest exact relevance with 59% (cf. Table 1), but in absolute terms as shown in Table 2 it obtains the lowest answer rate increase of 9.41%. The other operators are more complex and more likely to change the intent, the answer relevance is shifted towards *related* and *irrelevant* answers. For instance, ROO and GEN have the highest irrelevant answers, with 36% and 29%, respectively. This is intuitive given that the scope of the original question is reduced in  $q'$ , which can lead to unrelated answers. On the other hand, we observe that ROO+GEN has the most answers that are related to  $q$ , with 73% based on the human annotations, or 13.1% on the 1M test set (extrapolated results). It also obtains the least irrelevant answers as well as the highest answer rate, which we speculate is because the root wh- transformation and generalization reduce answering complexity and broaden recall, leading to a better pool of candidate answers for the QA system. Furthermore, the different operators are complementary (cf. Appendix G), hence, their combination achieves the best result.

### 6.4 Live QA Deployment

The SURF-ROO model<sup>7</sup> was deployed for real-time reformulation of unanswered questions in a leading voice assistant. This live deployment enables answering for millions of previously unanswered requests. Each day we solicit explicit binary relevance feedback from a portion of customers receiving answers of SURF reformulations, with metrics exceeding or matching those reported in Table 1.

## 7 Conclusion

We tackled the problem of improving spoken QA, and analyzed questions from live data to identify key challenges that could be addressed with reformulation. Based on this we proposed SURF with novel linguistically-motivated reformulation operators to solve the identified challenges. Offline experiments show the effectiveness of our novel root transformation and generalization operations, with up to 24% of unanswered questions being answered via reformulations with high answer relevance. Live deployment in a leading voice assistant has positively impacted millions of requests.

<sup>7</sup>We chose the best single operation model due to the doubled latency of ROO+GEN.

We showed reformulation helps QA systems adapt to spoken user questions. We presented key insights from a deployed solution showing that performance can be significantly increased, without changing the underlying QA backends, by simply improving questions in their syntax and semantics.

## Limitations and Future Work

In this work we did not consider the following aspects, which we discuss below and lay out directions for how to address them in future work.

**Combining Reformulation Operations:** The reformulation operators, except REP, which is applied jointly with other operators, are applied sequentially, in their given order, e.g. ROO+GEN. This has two potential limitations that we aim to address in future work. First, applying multiple operators sequentially has the negative impact of increased inference latency as the SURF model needs to be applied multiple times, which can become a bottleneck for systems that process large traffic volumes. Second, by applying sequentially the reformulation operators, the likelihood of cascading errors or the model making mistakes in terms of the target reformulation shape increases. We aim to address this limitation in the future by fine-tuning the model to jointly perform multiple reformulation operators in a single pass.

**Large Language Models (LLM):** In this work we relied on BART (Lewis et al., 2020) as our seq2seq model, and did not experiment with newer multi-billion parameter LLMs. Recently we have seen rapid progress in the space of LLMs, both in terms of model size and their capabilities to perform various tasks (Chung et al., 2022). However, we note that deploying LLMs is limited by their high inference latency, particularly in high-traffic, low-latency systems such as ours. Furthermore, for experimenting with API-based approaches such as ChatGPT and GPT-4, using these systems was not possible due to data confidentiality. While we will explore leveraging LLMs for this task in the future, current experimental results show that even smaller language models such as BART, with a sufficient amount of training data, can be fine-tuned to perform the task accurately.

**Evaluation on Public Datasets:** Our evaluation focused on real-world unanswered user utterances from voice assistants. We did not use public

datasets as currently available resources do not accurately represent customer behavior at scale. However, the community is aware of this divergence, and there are initial efforts in different NLP tasks to create public datasets that represent real-world user behavior. For example, in the task of Named Entity Recognition there has been recent work on bridging the gap between academic datasets and real-world problems by creating new resources that represent contemporary challenges that are encountered in practice (Fetahu et al., 2023; Malmasi et al., 2022). In future work we will consider evaluating SURF on such datasets as they become available. Furthermore, the findings from our work may be used to create data that includes the challenges we identified as part of our analysis (either by organically collecting such data, or simulating it to generate synthetic data).

**Multilingual Experiments:** We only considered English-language questions in this work, and it will be of interest to consider how our approach can be extended to other languages using multilingual models. The evaluation of cross-lingual transfer for this task is another open research area.

## Acknowledgments

We would like to thank the following people for their valuable assistance and feedback: Danielle Class, Eugene Agichtein, Jim Fakonas, Leela Prabhu, Mohamed Nasreldin, and Zhiji Liu.

## References

- Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. 2011. Query reformulation mining: models, patterns, and applications. *Information retrieval*, 14:257–289.
- Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. *Ask the right questions: Active question reformulation with reinforcement learning*. In *International Conference on Learning Representations*.
- Matthew Byrd and Shashank Srivastava. 2022. *Predicting difficulty and discrimination of natural language questions*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–130, Dublin, Ireland. Association for Computational Linguistics.
- Wallace Chafe and Deborah Tannen. 1987. The relation between written and spoken language. *Annual review of anthropology*, 16(1):383–407.

- Yllias Chali and Sadid A. Hasan. 2012. [Simple or complex? classifying questions by answering complexity](#). In *Proceedings of the Workshop on Question Answering for Complex Domains*, pages 1–10, Mumbai, India. The COLING 2012 Organizing Committee.
- Zhiyu Chen, Jie Zhao, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. [Reinforced question rewriting for conversational question answering](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 357–370, Abu Dhabi, UAE. Association for Computational Linguistics.
- Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. [Decontextualization: Making sentences stand-alone](#). *Transactions of the Association for Computational Linguistics*, 9:447–461.
- Zwei Chu, Mingda Chen, Jing Chen, Miaosen Wang, Kevin Gimpel, Manaal Faruqui, and Xiance Si. 2020. How to ask better questions? a large-scale multi-domain dataset for rewriting ill-formed questions. In *Proc. of AAAI*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [Boolq: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.
- Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. [Low cost portability for statistical machine translation based on n-gram frequency and TF-IDF](#). In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.
- Majda Ennaciri. 2022. Searching for explanation of difficult scientific terms.
- Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open Question Answering Over Curated and Extracted Knowledge Bases. In *KDD*.
- Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023. [Semeval-2023 task 2: Fine-grained multilingual named entity recognition \(multiconer 2\)](#). *CoRR*, abs/2305.06586.
- Wee Chung Gan and Hwee Tou Ng. 2019. [Improving the robustness of question answering systems to question paraphrasing](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.
- Aditya Gupta, Jiacheng Xu, Shyam Upadhyay, Diyi Yang, and Manaal Faruqui. 2021. [Disfl-QA: A Benchmark Dataset for Understanding Disfluencies in Question Answering](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3309–3319.
- Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to rewrite queries. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1443–1452.
- Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. [Ctrl: A conditional transformer language model for controllable generation](#). *arXiv preprint arXiv:1909.05858*.
- Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. [Which linguist invented the lightbulb? presupposition verification for question-answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, Online. Association for Computational Linguistics.
- Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. [Inquisitive question generation for high level text comprehension](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6544–6555, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Ye Liu, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2019. [Generative question refinement with deep reinforcement learning in retrieval-based qa system](#). In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19*, page 1643–1652, New York, NY, USA. Association for Computing Machinery.
- Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. [Multiconer: A large-scale multilingual dataset for complex named entity recognition](#). In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 3798–3809. International Committee on Computational Linguistics.



- Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. [Controllable sentence simplification](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.
- Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. [What kind of language is hard to language-model?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.
- Sara Moradlou, Xiaobei Zheng, TIAN Ye, and Jonathan Ginzburg. 2021. Wh-questions are understood before polar-questions: Evidence from english, german, and chinese. *Journal of Child Language*, 48(1):157–183.
- Amihai Motro. 1984. Query generalization: A method for interpreting null answers. In *Expert Database Workshop*, pages 597–616.
- Islam Nassar, Michelle Ananda-Rajah, and Gholamreza Haffari. 2019. [Neural versus non-neural text simplification: A case study](#). In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 172–177, Sydney, Australia. Australasian Language Technology Association.
- Jeffrey Pomerantz. 2005. A linguistic analysis of question taxonomies. *Journal of the American Society for Information Science and Technology*, 56(7):715–728.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Stefan Riezler and Yi Liu. 2010. [Query rewriting using monolingual statistical machine translation](#). *Computational Linguistics*, 36(3):569–582.
- Elizabeth Salesky, Matthias Sperber, and Alex Waibel. 2019. Fluent Translations from Disfluent Speech in End-to-End Speech Translation. In *Proceedings of NAACL*, pages 2786–2792.
- Kim Cheng Sheang and Horacio Saggion. 2021. [Controllable sentence simplification with a unified text-to-text transfer transformer](#). In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Elizabeth Shriberg. 2005. Spontaneous speech: How people really talk and why engineers should care. In *Ninth European Conference on Speech Communication and Technology*.
- Noriko Tomuro. 2003. [Interrogative reformulation patterns and acquisition of question paraphrases](#). In *Proceedings of the Second International Workshop on Paraphrasing*, pages 33–40, Sapporo, Japan. Association for Computational Linguistics.
- Wayne Ward. 1989. [Understanding spontaneous speech](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Hai Ye, Hwee Tou Ng, and Wenjuan Han. 2022. On the Robustness of Question Rewriting Systems to Questions of Varying Hardness. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2100–2113.

## Appendix

### A Spoken QA Failure Analysis

We analyzed data to understand prevalent challenges in spoken QA failure. Unlike prior work, which uses Machine Reading Comprehension (MRC) datasets like SQuAD, we leverage real questions from a leading voice assistant.<sup>8</sup>

We performed a quantitative analysis, taking two large random samples of answered and unanswered user queries, totalling 10 million unique questions. For all questions, we compute several sentence-level variables (length, type-token ratio, TF-IDF) which are predictive of language complexity (Mielke et al., 2019; Byrd and Srivastava, 2022; Ennaciri, 2022), and measure their correlation with whether the question was answered. We also hypothesize that the question’s linguistic shape is important (see §2). Following prior work (Pomerantz, 2005), we define a syntactic question typology (Table 3) and develop an accurate type classification heuristic (c.f. Appendix B for details).

Table 4 shows the results. For confidentiality, we only report correlations and relative differences between the answered and unanswered groups, whose sizes cannot be disclosed. We note that longer questions and those with higher specificity (i.e., IDF) are more likely to be unanswered. Higher TTR (i.e., fewer repeated tokens) results in higher answer rates, likely because repetition is associated

<sup>8</sup>We do not consider ASR challenges in this work, and only deal with text transcripts.

Type	Description
Root wh-question	The wh-phrase is clause-initial. (“ <i>Who is the US president?</i> ”, “ <i>How large is an elephant?</i> ”)
Polar (Yes-No)	Asks if a statement is true. (e.g. “ <i>Is it going to rain tomorrow?</i> ”, “ <i>Can cats eat onions?</i> ”)
Open	Open-ended <i>how</i> questions. (e.g. “ <i>How does depression affect the body?</i> ”)
Request	Direct request beginning with a verb. (e.g. “ <i>Tell me the capital of Utah.</i> ”)
Other	Any other utterance. (e.g. “ <i>watermelon health benefits</i> ”, “ <i>sports softball in Denver</i> ”)

Table 3: A list of the types in our question typology.

Variable	Pearson Correlation ( $r$ )
Token Length	-0.25
Char Length	-0.24
Type-Token Ratio (TTR)	+0.12
Mean of IDF scores	-0.13
Sum of IDF scores	-0.30
Sum of TF-IDF scores	-0.31
Mean of TF-IDF scores	-0.12

Question Type	Difference w/ Answered
Root	-10.4%
Polar	+8.5%
Open	+2.4%
Request	-2.1%
Other	+1.6%

Table 4: Top: Pearson correlation between question characteristics and if it was answered (all  $p < 0.001$ ). Bottom: Distributional differences in question types between the unanswered and answered questions.

with disfluencies. Question type also has a big impact on answerability. Simple root wh-questions are less prevalent in the answered subset, while polar questions are much more frequent in the unanswered subset.

## B Question Type Classification

We develop a rule-based algorithm to classify a question into a predefined type (cf. Table 3).

Algorithm 1 shows our heuristic to determine the question type. The algorithm is a rule-based and applied in cascade, until there is a match between question and type. The evaluation order is the same as listed in Table 3, from top to bottom.

- **Root:** a question which starts with a *wh*-\* or some specific *how*- bigrams.
- **Polar:** A yes or no question starting with predefined keywords.
- **Open:** Start with *how*, but not a root question.

- **Request:** They are a command to a QA system and start with a verb.<sup>9</sup>
- **Other:** If anything else, sentences are labeled as other.

Below are listed some of the input variables necessary for Algorithm 1.

- **wh-\* or how-\* bigrams:** “*what*”, “*where*”, “*when*”, “*which*”, “*who*”, “*why*”, “*how much*”, “*how many*”, “*how long*”, “*how old*”, “*how early*”, “*how soon*”, “*how wealthy*”, “*how rich*”, “*how big*”, “*how small*”, “*how tall*”, “*how short*”, “*how heavy*”, “*how often*”, “*how late*”, “*how far*”, “*how high*”, “*how fast*”, “*how quickly*”, “*how close*”;
- **polar keywords:** “*do*”, “*does*”, “*did*”, “*can*”, “*was*”, “*were*”, “*should*”, “*is*”, “*isn*”, “*has*”, “*have*”, “*are*”, “*aren*”, “*will*”;

---

### Algorithm 1 Heuristic for question types

---

**Require:** sentence  $s$

```

if  $s$  starts with wh-* or how-* bigrams then
  type  $\leftarrow$  root
else if  $s$  starts with a keyword from polar keywords list then
  type  $\leftarrow$  polar
else if  $s$  starts with “how” then
  type  $\leftarrow$  open
else if  $s$  starts with verb then
  type  $\leftarrow$  request
else
  type  $\leftarrow$  other
end if
return type

```

---

### B.1 Heuristic Accuracy Evaluation

To evaluate the accuracy of our heuristic algorithm, we randomly sampled 100 questions from each question type from the testing set and annotated whether the classified question type is correct. In total, 500 questions were annotated and the overall accuracy is 95%. The accuracy of each question type is summarized in Table 5.

<sup>9</sup>We use spaCy for POS-Tagging.

Question Type	Accuracy (%)
Root	0.98
Polar	0.98
Open	0.95
Request	0.89
Other	0.95
Average	0.95

Table 5: Question classification heuristic accuracy (based on human assessment), for each question type.

Original Question Type	ROO	REP
Root	0	544,440
Polar	23,201	200,942
Open	36,322	257,704
Request	15,405	450
Other	113,568	9,348
Total	188,496	1,012,884

Table 6: Distributions of question types in the weakly supervised pre-training data for the ROO and REP operators.

## C Model Implementation Details

For both our approach and the baseline, we adopt BART (Lewis et al., 2020)<sup>10</sup> as our reformulation model  $\mathcal{F}$ . As annotating the GEN task is not possible for all questions (as not all of them are generalizable, e.g., “Who is Joe Biden?”), this results in a smaller amount of training data for the GEN task. To address this, we upsampled the generalized reformulations by 5x during training so that the number of generalization samples matches other types of reformulations. We train it for up to 20 epochs with a learning rate of  $lr = 1e - 6$  and use Adam as our optimizer, and batch size of 16. The training is halted using early stopping, if the validation loss is non-decreasing after 3 epochs.

## D Pre-training Data

To prepare the weakly-supervised data for pre-training, we first apply our question type heuristic from Appendix B to classify the original questions and reformulations in the MQR dataset.<sup>11</sup> We then automatically derive operator task labels from those question types using the method described in §5.3. This process yields 1.2M samples. Table 6 shows

<sup>10</sup><https://huggingface.co/facebook/bart-base>

<sup>11</sup><https://github.com/ZeweiChu/MQR>

the distribution of task labels and question types in the data. As noted earlier, data for the GEN operator cannot be reliably derived with weak supervision on this dataset. The large majority of the data contains repairs, as that is the intended purpose of the MQR dataset. Table 7 and Table 8 list some example questions from the MQR dataset with their assigned question types for the REP and ROO operators, respectively.

## E Annotation Guidelines

Here we describe in detail the question reformulation annotation guidelines. First, the steps for each reformulation operator are described, then a general overview of annotation guidelines for the entire annotation process is shown.

### E.1 Instructions for REP

REP reformulations must:

- not contain repetitions, false starts, and self corrections.
- be grammatically correct. For example, “*Is Bill Pullman have a son?*” → “*Does Bill Pullman have a son?*”.
- be impersonal and formal. For example, “*Where can I get a booze after 11 pm?*” → “*Which stores sell beer after 11 pm?*”.
- keep the original question type (e.g., root → root).

### E.2 Instructions for ROO

ROO question reformulations must satisfy the following constraints:

- The reformulation must be a root question as defined in Appendix B. For example, “*Is there any easy way to make money online?*” → “*What is the easiest way to earn money online?*”.
- Reformulations must retain the intent of the original question. In the above example, the question type is changed from polar to root. However, the answer to the reformulated root question can still provide an answer to the original polar question. Reformulations where the intent is changed are invalid: “*Can you freeze chicken that’s already been thawed?*” → “*How long can chicken be frozen for before going bad?*”.
- The reformulation additionally should satisfy the REP constraints, with the exception of altering the question type.

Question	Source Type	Reformulation	Target Type
"where does spider live in?"	root	"where does a spider live?"	root
"what is the oridgin of the word mosque?"	root	"where does the word mosque come from?"	root
"how remember pronunciation of danish words?"	open	"how can i remember the pronunciation of danish words?"	root
"how can we make money from youtube?"	open	"how do people earn money from youtube?"	root
"does the grammar generates the words?"	polar	"does the grammar generate the words?"	polar
"can charity claim patent on medicine?"	polar	"can charities be granted patents on medicine?"	polar
"winners in olympic in 2000?"	other	"names of olympic winners of 2008?"	other
"at what tempature does alcohol freeze?"	other	"at what temperture does alcohol freeze?"	other
"find out some advantages for setting up a partnership?"	request	"give 2 advantages of a business partnership?"	request
"name three groups of polymers and name one type of a composite?"	request	"name three common polymers?"	request

Table 7: REP examples of weakly-labeled pre-training data from the MQR dataset as labeled by our heuristics.

Question	Source Type	Reformulation	Target Type
"how do you know if your local bike club is worth paying for?"	open	"what benefits do bike clubs provide?"	root
"how do you forgive other people?"	open	"what's the best way to forgive people?"	root
"an example of enzyme mimic is ?"	other	"what are examples of enzymes and antibodies ?"	root
"basic difference between compilers and interpreters?"	other	"what are the differences between compilers and interpreters?"	root
"explain the ending of batman arkham city to me"	request	"what happens in the ending of batman arkham city?"	root
"unscrewing sliding window lock"	request	"what tool works with this star-shaped screw with a post in the middle?"	root
"do blackholes exist?"	polar	"why do black holes exist?"	root
"is there any easy way to make money online?"	polar	"what is the easiest way to earn money from online?"	root
"are there any good challenging puzzles?"	polar	"what are some good word puzzles?"	root

Table 8: ROO examples of weakly-labeled pre-training data from the MQR dataset as labeled by our heuristics.

### E.3 Instructions for GEN

GEN reformulations may slightly change the information that is sought in the original question to something more general. This can be done by removing parts of a question (adjuncts or other clauses), and modifying referenced entities. Note that we do not make parallel entity changes (e.g. "Los Angeles" → "San Francisco"), but rather perform vertical generalization (e.g., with hypernyms or holonyms "Los Angeles" → "California"). There are different cases to generalize a question:

- The reformulation is less restricted than the original question w.r.t some entity (e.g., "What do pythons eat?" → "What do snakes eat?");
- The reformulation is more general than the original question regarding conditions/constraints (e.g., "Who is the tallest person in the USA?" → "Who is the tallest person?").

For any given question, multiple distinct generalizations may be possible.

### E.4 Overall Guidelines for Annotators

You will be given questions and asked to generalize them or reshape them into other types. All

your reformulations must be done with respect to the original question. An original question can be generalized up to 3 times. Please complete the following steps for each question:

- Question Validity (prior to any reformulation):
  1. Judge whether the question seeks a valid answer. A question is invalid if you are unable to understand the question's intent. Or, alternatively, you judge that the question is unanswerable. This may be the case for personal questions (e.g. "do I have COVID?"). If the question is invalid, remove question from the training dataset.
- Perform REP reformulation:
  1. Refer to §E.1 to make sure your reformulation adheres to all REP constraints.
- Do ROO reformulation:
  1. Refer to §E.2 to make sure your reformulation adheres to all constraints.
  2. If it is unfeasible to make the reformulation without changing the question's intent, leave blank.
  3. Do not reformulate root questions.

- Do GEN reformulation:
  1. Write down up to 3 generalized reformulation of the original question. If possible, try to perform different types of generalization.
  2. Refer to §E.3 to make sure your reformulation adheres to all constraints.

### E.5 Sampling Strategy

To sample questions for annotation, we first filter questions with fewer than 5 tokens or more than 13 tokens. Then we adopt the unseen strategy (Eck et al., 2005) using bi-grams to select questions that cover diverse topics. For each question, we collect up to 3 different generalized reformulations, given that a question can be generalized in different ways.

### F Extrinsic Evaluation Data

To evaluate the performance of reformulations on our QA system, we take a representative sample of 1M unanswered questions from the real traffic as the test set, where the distribution across different question types is kept to the real traffic distribution. However, due to confidentiality reasons, we cannot reveal the exact question type distribution.

### G Operator Contingency Tables

A natural question is whether different operators are correlated, i.e., they lead to improved answering on the same set of questions, or if they are complementary/orthogonal by improving non-overlapping subsets of questions. To understand this relationship we performed a cross tabulation analysis by building 2x2 contingency tables comparing different operators on our test set. Each operator is represented by a binary variable indicating whether the reformulation by that operator resulted in the unanswered question becoming answered.

Table 9 shows the results of this analysis for the SURF model. We observe that there is substantial degree of orthogonality between the operators, as evidenced by cases where one operator fails and the other succeeds, e.g., ROO can improve answering on 6.98% of the data where REP fails to do so. The largest correlation is between ROO and ROO+GEN, while the lowest is between ROO and REP. All operators are best complemented by ROO+GEN. The trend is inline with the results shown in Table 2, where ROO+GEN has the highest number of answered reformulated questions.

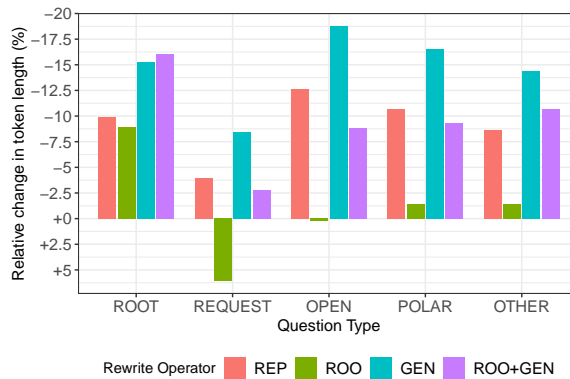


Figure 3: Relative change in token length after applying the different reformulation operators.

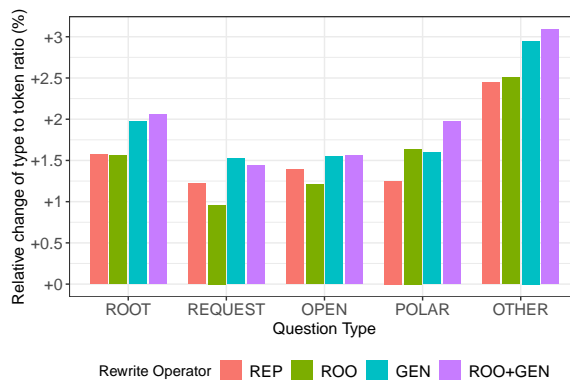


Figure 4: Relative increase in type-token ratio after applying the different reformulation operators.

### H Analysis of Reformulations Changes

We also consider how our reformulation operators change the original questions in terms of length and type-token ratio (TTR). Previously, in Table 4 of Appendix A we showed that these question characteristics are correlated with answer rate. As a follow up, we examined how the SURF reformulations change these variables.

Figure 3 shows that SURF reformulations from all operators significantly shorten the input questions, indicating that they result in simplified questions. The micro-averaged length reduction across all question types for each operator is 9.9% for REP, 4.7% for ROO, 15.5% for GEN, and 12.6% for ROO+GEN. The average length of a question reformulation by ROO increases only for open and request question types, while it decreases in all other cases. However, for open and request question types, ROO makes the question more specific (e.g., “explain how to play football” is reformulated into “what is the best way to play football?” by ROO). Sometimes, ROO also makes polar ques-

	<b>REP = 0</b>	<b>REP = 1</b>		<b>ROO+GEN = 0</b>	<b>ROO+GEN = 1</b>
<b>GEN = 0</b>	83.80%	1.87%	<b>GEN = 0</b>	77.81%	7.85%
<b>GEN = 1</b>	6.79%	7.54%	<b>GEN = 1</b>	4.27%	10.07%
<b>ROO = 0</b>	83.61%	3.21%	<b>ROO = 0</b>	80.28%	6.55%
<b>ROO = 1</b>	6.98%	6.20%	<b>ROO = 1</b>	1.80%	11.38%

	<b>GEN = 0</b>	<b>GEN = 1</b>		<b>ROO+GEN = 0</b>	<b>ROO+GEN = 1</b>
<b>ROO = 0</b>	80.27%	6.55%	<b>REP = 0</b>	79.25%	11.34%
<b>ROO = 1</b>	5.39%	7.79%	<b>REP = 1</b>	2.83%	6.58%

Table 9: Comparing the percentage of answered (1) and unanswered (0) questions between two operations in crosstables.

tion more specific (e.g., “*is aspartame safe?*” → “*what are the health risks of eating aspartame?*”).

Regarding TTR, from Figure 4 we observe a consistent TTR increase for all reformulation types. This is consistent with the findings from Table 4, where it was shown that TTR is positively correlated with answer rate. The increase of TTR means the reformulated questions are less likely to have repeated tokens.

## I Question Reformulation Examples

We show synthetic examples of question reformulations output by our model in Table 10.

Task	Origin	Reformulation
REP	“why why are the dinosaurs go away from them”	“why did dinosaurs go away?”
REP	“how do tiger how do baby tigers speak”	“how to communicate with baby tigers?”
REP	“korean food new jersey”	“what is the best korean food in new jersey”
REP	“hey find me some vinegar potato salad recipes”	“how to make vinegar potato salad”
REP	“are bill pullman have a son”	“does bill pullman have a son”
REP	“what time of the year do all the lightning bugs come out”	“when do lightning bugs come out”
REP	“when is new movie about minions”	“when will the next movie about minions be released”
REP	“how much does it how long does it take to boil one and a half cups of water”	“how long does it take to boil water”
REP	“what’s the macros of rotisserie chicken”	“what are the macros of a rotisserie chicken”
ROO	“composer dona zetti”	“who is the composer dona zetti”
ROO	“does a lot of sugar will hurt your teeth”	“why does sugar hurt your teeth”
ROO	“hey find me vinegar potato salad recipes”	“what are some good vinegar potato salad recipes?”
ROO	“look up some macron recipes”	“what are some macron recipes”
ROO	“hey show me a marinade for pepper steak”	“what marinade should be used for pepper steak”
ROO	“are bill pullman have a son”	“who is bill pullman’s son”
ROO	“in an air fryer at what temperatures do i heat fully cooked bratwurst and for how long”	“how long should bratwurst be cooked in an air fryer”
ROO	“yes did steven spielberg go to films school”	“where did steven spielberg go to film school”
ROO	“how do you do reminder for my phone”	“how to set up a reminder for a phone?”
ROO	“again show me the recipe for pan seared steak”	“what are some recipes for pan seared steak”
GEN	“how long can fleas live without host”	“how long can fleas live”
GEN	“in 1901 who won the first nobel prize for physics”	“who won the first nobel prize for physics”
GEN	“are large feet assigned beauty some cultures”	“are large feet assigned beauty”
GEN	“did kamala harris ever move to canada than back to america”	“did kamala harris move to canada”
GEN	“how many playable characters are in lego star wars the skywalker saga”	“how many playable characters are in lego star wars”
GEN	“tell me what is the total number of people who ever lived on earth”	“how many people have ever lived on earth”
GEN	“price of a wheel for a two thousand five volvo s eighty”	“what is the price of a wheel for a car”
GEN	“is it good to put a watermelon that is not been peeled in the ice box”	“is it good to put watermelon in the ice box”
GEN	“how is it another way to defeat hurricane”	“how to defeat a hurricane”
ROO+GEN	“in an air fryer at what temperatures do i heat fully cooked bratwurst and for how long”	“how long should bratwurst be cooked”
ROO+GEN	“in 1901 who won the first nobel prize for physics”	“who won the first nobel prize”
ROO+GEN	“what is the average salary of a taxi driver from san francisco”	“how much does a taxi driver earn in california”
ROO+GEN	“in feet how long was the largest shark ever on the record”	“what is the longest shark”
ROO+GEN	“who is the actor’s name who plays eleven in stranger things”	“who is the actor that plays eleven in stranger things”
ROO+GEN	“what is a apple or lemon that starts with letter f.”	“what is a fruit that start with f”
ROO+GEN	“does hair grow faster when it is warm or cold outside”	“how fast does hair grow”
ROO+GEN	“can you tell us some facts about patrick mahomes achievements”	“what are facts about patrick mahomes”
ROO+GEN	“if i workout for about ten minutes everyday how many days will it take for me to lose a pound”	“how long does it take to lose weight?”

Table 10: Synthetic examples of reformulated questions according to the different reformulation operators.