*Proceedings of Machine Learning Research* **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

# Cause-Conditioned Multi-Task Learning for Answerable Question Suggestion in MRC

Hadiseh Moradisani[†,*], Fattane Zarrinkalam[†], Julien Serbanescu[†], Zeinab Noorian[‡],
[†] College of Engineering, University of Guelph
[‡] Ted Rogers School of Information Management, Toronto Metropolitan University

**Abstract**

Machine Reading Comprehension (MRC) systems struggle when user questions are unanswerable given the passage: most simply output "no answer", leaving users without guidance on how to recover useful information. We introduce a *cause-conditioned multi-task learning (MTL)* framework that turns failure into follow-up by jointly (1) classifying an input as answerable or as one of six fine-grained unanswerability causes (Entity Swap, Number Swap, Antonym, Negation, Mutual Exclusion, No Information), and (2) generating a revised, context-grounded answerable question conditioned on the predicted cause label and an extracted guidance sentence. Using an ensemble of strong readers plus LLMs-as-judges, we apply majority voting to test whether rewrites become answerable. A human study further assesses fluency, relevance, and usefulness. Our cause-conditioning MTL framework yields better recovery from unanswerable inputs and earns strong human ratings, advancing user-supportive, failure-aware MRC.

**Keywords:** MRC, SQuAD2.0, Multi-Task Learning, Unanswerable Question

## 1. Introduction

Machine Reading Comprehension (MRC) aims to answer natural-language questions using evidence grounded in a provided passage [1, 2]. It is a core capability behind many real-world applications, including virtual assistants, search engines, and customer support systems [1, 3, 4]. Despite steady progress, real deployments continue to face a persistent failure mode: *unanswerable* questions, for which no correct answer can be derived from the given context [1, 5–9]. When such cases are mishandled, by returning a plausible but incorrect answer or by offering no actionable guidance, user trust quickly erodes and overall system utility degrades [10–12].

Most existing MRC pipelines address unanswerability through abstention mechanisms, such as predicting a null span, lowering confidence scores, or incorporating answer verification modules to reduce false positives [10, 11, 13–15]. While abstention is essential for safety and correctness, it is often insufficient from a user-centric perspective. In many practical scenarios, users do not merely want a rejection; they want help *recovering* from failure. If a question cannot be answered as posed, an effective system should ideally suggest a revised question that *can* be answered from the available passage, allowing the interaction to proceed productively.

Motivated by this need, recent work has explored question reformulation and clarification for unanswerable inputs [16–19]. However, reformulating unanswerable questions in paragraph-based MRC remains challenging. Even strong language models frequently produce fluent paraphrases that preserve the original mismatch between the question and the passage, resulting in genuine recovery in only a minority of cases [20, 21]. The core difficulty is that unanswerability is rarely a surface-level phrasing issue; it typically arises from a specific underlying reason the question fails against the given context.

This work is motivated by the observation that unanswerability in MRC is not a single phenomenon but arises from distinct failure causes, such as incorrect entities, numerical mismatches, negation, mutual exclusivity, or missing information. Each failure mode requires a

* hmoradis@uoguelph.ca

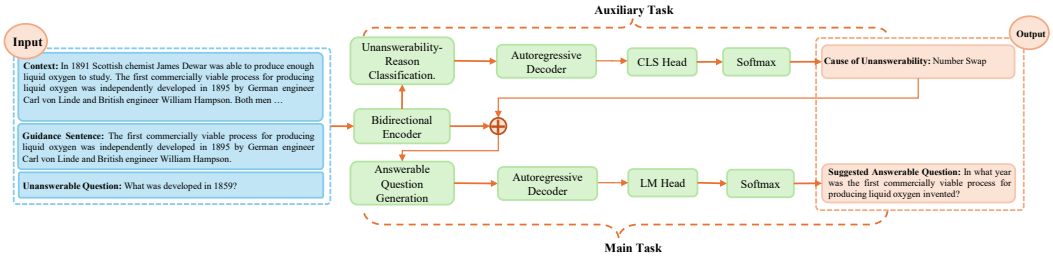


*Figure 1.* Overview of our proposed MTL+Guide framework

different corrective strategy, and treating all unanswerable questions uniformly often leads to generic rewrites that fail to recover answerability.

We therefore formulate answerable question suggestion as a *cause-conditioned* learning problem and propose a multi-task learning (MTL) framework that jointly performs (1) answerability and unanswerability-cause classification and (2) answerable question generation. Given a question and its context, the classifier predicts whether the question is answerable or assigns one of six fine-grained unanswerability causes. The generation module then conditions on the predicted cause, the passage context, and a retrieved guidance sentence to produce a targeted, answerable reformulation. Our use of MTL is motivated by extensive empirical and theoretical evidence that jointly training related tasks improves representation learning and generalization through shared inductive transfer [22–26].

We evaluate our approach on *UnAnswGen*, a large-scale dataset derived from SQuAD2.0 that provides paired answerable–unanswerable questions and supervision over failure causes. Our contributions are threefold: (1) a cause-conditioned MTL framework for unanswerability-aware question generation; (2) integration of a fine-grained unanswerability taxonomy directly into learning for controllable reformulation; and (3) a comprehensive set of single-task and two-stage baselines for systematic comparison. Experiments with BART [27] and T5 [28] encoder–decoder backbones show consistent improvements in fluency, contextual faithfulness, and answerability over alternative formulations. The code and executable workflow are publicly available.[1]

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the proposed cause-conditioned multi-task learning framework. Section 4 describes the experimental setup and evaluation methodology. Section 5 presents results and analyses, and Section 6 concludes the paper.

## 2. Related Work

**Unanswerable Question Handling in MRC Systems.** Unanswerable questions have long been recognized as an important challenge in MRC, particularly since the introduction of datasets such as SQuAD2.0 that explicitly include questions whose answers are not supported by the given passage [5, 6]. This line of research primarily focuses on enabling models to recognize when the available context does not justify an answer and to avoid producing unsupported responses.

A common strategy is to predict a null answer span or to rely on confidence scores that allow the model to abstain when evidence is insufficient [10, 11]. Several approaches extend this idea by introducing verification components that assess whether a predicted answer is grounded in the passage. For example, U-Net jointly models answer extraction and answerability verification, allowing the system to reason about both tasks in a unified architecture [13]. Other work explores adversarial training and data augmentation techniques that expose models to challenging unanswerable cases, improving robustness against misleading lexical overlap or spurious cues [8, 14, 15].

[1] https://github.com/Hadis-mrd/cause-conditioned-mtl-mrc

Beyond model architecture, researchers have also investigated lightweight adaptation and regularization techniques for handling unanswerable questions. Prompt- and prefix-tuning methods adapt pretrained language models to settings where contextual evidence is incomplete or ambiguous [7]. Consistency-based learning further encourages stable predictions when questions or contexts are semantically perturbed, strengthening reliability under unanswerable conditions [29]. Additional efforts address interpretability and reliability by incorporating confidence estimation, uncertainty modeling, or user feedback mechanisms that help make abstention decisions more transparent [30].

Large language models have recently been considered within this broader context, often as auxiliary components for validation or fallback decision-making. While these models demonstrate strong language understanding, their use in paragraph-based MRC systems is typically shaped by practical considerations such as limited controllability, the absence of explicit intermediate signals for unanswerability, and deployment constraints related to privacy and auditability. As a result, LLM-based approaches are often integrated in a complementary or post-processing role rather than as structured mechanisms for modeling unanswerability [1, 20, 21, 31].

**Question Reformulation and Clarification for Unanswerable Questions.** MRC systems can respond more helpfully when questions cannot be answered as posed. Rather than abstaining, these approaches aim to reformulate or clarify the user query so that it becomes answerable under the available information. Early studies explore structured rewriting strategies, such as using knowledge graphs to align questions with supported entities and relations [16, 17]. Other work shows that semantic reformulation can recover valid answers for certain classes of unanswerable spoken or conversational queries [18].

Additional research highlights the importance of recognizing flawed assumptions in user questions. For example, models trained to identify and explain false presuppositions can provide more informative responses even when a direct answer is not possible [19]. These studies emphasize that understanding the nature of the mismatch between a question and the available evidence plays a key role in meaningful recovery.

In paragraph-based MRC settings, reformulation remains challenging because the passage is the only source of evidence and unanswerability often arises from subtle inconsistencies, such as incorrect entities, numerical discrepancies, negation, or missing information. Recent analyses suggest that even strong language models frequently generate fluent paraphrases that do not resolve these underlying issues [20, 21].

## 3. Cause-Conditioned MTL for Answerable Question Suggestion

**Overview.** We propose a unified *cause-conditioned Multi-Task Learning (MTL)* framework that combines an auxiliary *Answerability & Unanswerability-Reason Classification* task with a primary *Answerable Question Generation* task. Given a user question and its passage context, the classifier predicts one of seven labels (*Answerable*, *Entity Swap*, *Number Swap*, *Antonym*, *Negation*, *Mutual Exclusion*, or *No Information*) identifying whether and why the original query cannot be answered from the passage. The generation module then conditions on the predicted cause label and contextual evidence to produce a fluent, context-grounded follow-up question that is intended to be answerable. Unlike symmetric MTL approaches that weight tasks equally, we explicitly prioritize the generation objective: the classification head is trained primarily to supply informative causal guidance rather than to maximize its own standalone accuracy, directing model capacity toward user-helpful rewrites that improve recovery from unanswerable inputs.

Our framework, which is shown in Figure 1, is architecture-agnostic and can be instantiated with different pre-trained sequence-to-sequence encoder–decoders. In our experiments we use models from both the BART [27] and T5 [28] families, which have shown strong transfer in multitask settings spanning classification and generation [22–25]. Their

denoising- or span-reconstruction-style pre-training confers robustness to noisy or partially mismatched inputs, a common source of unanswerability, and supports conditional text generation needed for question rewriting. For each backbone we attach two task-specific output layers: a classification (CLS) head that predicts answerability or the unanswerability cause, and a language-model (LM) head that generates the rewritten question. We fine-tune both heads end-to-end so that discriminative signals from the CLS task inform generation, while generation gradients help learn representations that distinguish failure modes.

**Answerability and Unanswerability-Reason Classification.** We cast answerability detection as a seven-way classification problem and condition it on an auxiliary *guidance sentence*. Each training instance is encoded as a triplet $(c, q, g)$, where $c$ is the passage context, $q$ the user question, and $g$ a guidance span that highlights the evidence relevant to $q$ (or to its failure). Following [24], the guidance component is used to foreground salient information, disambiguate references, and surface gaps that contribute to unanswerability. Inputs are linearized as

$$\texttt{[CTX]}\ c\ \texttt{[Q]}\ q\ \texttt{[G]}\ g$$

(with model-specific segment tokens) and passed through the shared encoder of our backbone (i.e., BART or T5) to obtain a pooled representation $h$.[2]

The classifier predicts one of seven labels $R = \{r_1, \ldots, r_7\}$: *Answerable*, *Entity Swap*, *Number Swap*, *Antonym*, *Negation*, *Mutual Exclusion*, *No Information*. Each unanswerability category captures a distinct type of mismatch between the question and the context. Specifically, *Entity Swap* refers to questions containing an incorrect entity not supported by the context; *Number Swap* denotes numerical inconsistencies; *Antonym* and *Negation* correspond to semantic contradictions introduced through opposite meaning or negation; *Mutual Exclusion* captures logically incompatible constraints (e.g., asking for a spouse when the subject is described as unmarried); and *No Information* indicates that the required information is not present in the context. A linear layer projects $h$ to logits $z \in \mathbb{R}^{|R|}$; $\mathrm{softmax}(z)$ yields class probabilities $p(r_i \mid c, q, g)$. We optimize standard cross-entropy where $y_i$ is the one-hot gold label:

$$\mathcal{L}_{\text{CLS}} = -\sum_{i=1}^{|R|} y_i \log p(r_i \mid c, q, g) \tag{3.1}$$

**Role of Guidance** The guidance sentence serves as an explicit piece of contextual evidence that highlights either (1) the answer span for answerable questions or (2) the source of the mismatch responsible for unanswerability. In answerable cases, the guidance corresponds to the sentence containing the gold answer. In unanswerable cases, it instead surfaces the relevant context that contradicts or fails to support the user query. For example, if a user asks about *"Eight Ministries"* but the passage only mentions *"Six Ministries"*, a guidance sentence containing the latter enables the model to detect a content mismatch and assign the appropriate unanswerability cause (e.g., *Number Swap* under our schema).

**Answerable Question Generation (AQG).** Given the original (possibly unanswerable) user question $q$, passage context $c$, guidance sentence $g$, and the predicted unanswerability label $r \in R$ from the classification head, the goal is to generate an answerable follow-up question $q_a$ that is supported by $c$. We encode the input as a quadruplet $(q, c, g, r)$, linearized with special delimiters:

$$\texttt{[Q]}\ q\ \texttt{[CTX]}\ c\ \texttt{[G]}\ g\ \texttt{[R]}\ \textsc{Label}(r)$$

where $\textsc{Label}(r)$ is a short textual tag (e.g., `NUMBER_SWAP`) or a learned embedding associated with the class. The concatenated sequence is fed to the shared encoder of our backbone (BART or T5), producing contextual representations that capture interactions among the

[2]We use the encoder's final hidden state corresponding to the first special token; alternatives (mean pooling, attention pooling) produced similar trends.

user query, evidence span, and inferred failure mode. The decoder then conditions on these representations *and generates* a revised question $q_a$ intended to be answerable from $c$. During training we use teacher forcing and optimize the token-level negative log-likelihood (NLL) of the gold reformulated question. Formally, the generation loss is defined as:

$$\mathcal{L}_{\text{GEN}} = -\sum_{t=1}^{|q_a|} \log p\big(y_t \mid y_{<t}, q, c, g, r\big) \tag{3.2}$$

where $y_t$ denotes the gold token at decoding step $t$, and $y_{<t}$ represents the previously generated tokens.

Incorporating the cause label $r$ explicitly helps the model choose an appropriate rewrite strategy (e.g., correct an erroneous entity or number, invert a negation, resolve an antonym conflict, or ask about available information when none matches the original intent), yielding more targeted and fluent suggestions than cause-agnostic generation.

**Loss Weighting.** Training jointly with the classification task enables shared representations that reflect both evidence localization and failure semantics. We optimize a weighted sum of the classification loss $\mathcal{L}_{\text{CLS}}$ (Equation 3.1) and the generation loss (Equation 3.2).

$$\mathcal{L}_{\text{total}} = \alpha\, \mathcal{L}_{\text{CLS}} + \beta\, \mathcal{L}_{\text{GEN}} \tag{3.3}$$

The weights $(\alpha, \beta)$ control the relative influence of discriminative and generative learning signals.

## 4. Experimental Setup

### 4.1. Dataset

Our cause-conditioned MTL framework requires a dataset that provides (1) paired answerable and unanswerable questions within the same passage and (2) fine-grained labels indicating the cause of unanswerability. These properties are necessary to jointly support unanswerability detection and answerable question generation. This section discusses the steps involved in selecting an appropriate dataset.

**Step 1: Dataset Selection.** We use the *UnAnswGen* dataset [32][3], which extends SQuAD2.0 [6] by generating unanswerable questions from existing answerable ones. *UnAnswGen* contains 118,374 paired answerable and unanswerable questions, with each unanswerable instance annotated with one of six causes: *Entity Swap (17,444)*, *Number Swap (2,255)*, *Negation (45,053)*, *Antonym (27,749)*, *Mutual Exclusion (3,221)*, and *No Information (22,652)*. Each instance includes a shared passage, an answerable question, its unanswerable counterpart, the gold answer, and the corresponding cause label. Since UnAnswGen is derived from the SQuAD2.0 training set, we adhere to the original dataset splits and perform evaluation exclusively on the SQuAD2.0 development set augmented with SQuAD2-CR annotations, thereby preventing any potential data leakage.

**Step 2: Guidance Sentence Construction.** The paired structure of *UnAnswGen* enables deterministic extraction of guidance sentences. For unanswerable questions, we use the sentence containing the gold answer of the answerable counterpart as the guidance sentence, which highlights the evidence responsible for the mismatch or missing information. For answerable questions, the guidance sentence is the sentence containing the gold answer span. This unified design provides explicit contextual evidence for both classification and generation.

**Step 3: Data Balancing and Answerable Instances.** To address class imbalance across unanswerability causes, we construct a balanced training subset by downsampling each category. Because our framework must also identify answerable inputs, we additionally

[3] https://github.com/Julien-ser/UnAnswGen

*Table 1.* Dataset Statistics.

| | Answerable | Entity Swap | Number Swap | Negation | Antonym | Mutual Exclusion | No Information | Total |
|---|---|---|---|---|---|---|---|---|
| **Train Data** | 3,000 | 3,000 | 2,255 | 3,000 | 3,000 | 3,221 | 3,000 | *20,476* |
| **Dev Set** | 5,928 | 2,394 | 493 | 818 | 1,184 | 401 | 655 | *11,873* |

sample answerable questions from SQuAD2.0 and pair each with itself, using the answer-containing sentence as guidance. This ensures a consistent input format across all training instances.

**Step 4: Training and Evaluation Splits.** The resulting balanced dataset, summarized in Table 1, is used for training. For evaluation, we employ the SQuAD2.0 development set and adopt the fine-grained unanswerability labels provided by SQuAD2-CR [33].

### 4.2. Settings

We fine-tune `facebook/bart-base` and `t5-small` models using the Hugging Face framework. For both architectures, the maximum input sequence length is set to 512 tokens. Models are trained for 6 epochs with a batch size of 2, using the Adam optimizer with a learning rate of $5 \times 10^{-5}$. All experiments are conducted on NVIDIA GeForce RTX 2080 Ti and NVIDIA A30 GPUs.

Our framework jointly optimizes an auxiliary classification objective and a primary generation objective using a weighted multi-task loss (Equation 3.3). We select the loss weights $(\alpha, \beta)$ via 5-fold cross-validation under the constraint $\alpha + \beta = 1$ (5-fold cross-validation is used solely for selecting the loss weights). Among the evaluated configurations, $(\alpha, \beta) = (0.2, 0.8)$ achieves the best average performance across folds and is therefore used in all experiments. The choice $\beta > \alpha$ reflects our emphasis on high-quality answerable question generation, while still leveraging unanswerability detection as a guiding signal. This weighting prevents the model from over-prioritizing the classification task and helps maintain fluent, relevant, and answerable reformulations.

### 4.3. Evaluation Methodology

Because the SQuAD2.0 dev set does not provide ground-truth *answerable reformulations* for its unanswerable questions, we evaluate generation quality using a six-model majority-voting protocol adapted from the UnAnswGen framework [32]. The voter ensemble includes four of the top systems used in UnAnswGen [32]: Retro-Reader [11], mdeberta-v3-base-squad2, electra-base-squad2, and roberta-large-squad2. To capture recent advances in using LLMs as evaluators in NLP [34–38], we augment the ensemble with GPT-4o [39] and Llama-3-8B-Instruct [40]. The evaluation process includes two main steps: **(1)** Each model receives the original (input question, context) pair and assigns it a label (0 if unanswerable, 1 if answerable). **(2)** Each model then assesses the (suggested question, context) pair in the same way, assigning it a 0 if unanswerable and a 1 if answerable. This results in label pairs from the set $\{(1,1),(0,0),(1,0),(0,1)\}$. The suggested question is considered answerable if at least 4 out of 6 models label it as $(0,1)$ when the original question is unanswerable, or as $(1,1)$ when it is answerable. We define the accuracy of our generation task as the ratio of the number of suggested questions classified as answerable to the total number of input questions.

### 4.4. Baselines

We compare our framework against two categories of baselines: **(1) Single-Task Learning (STL).** We train question generation models without multi-task objectives. The simplest variant, *SL*, conditions only on the core inputs (original question + context). The enriched variant, *SL-R*, additionally conditions on the gold unanswerability reason label, enabling us to isolate the effect of explicit reason supervision on generating answerable reformulations. **(2) Two-Stage (TS).** We also implement a cascaded approach. Stage 1 predicts the cause of unanswerability from the question's context (optionally augmented with a guidance sentence). Stage 2 then conditions on the original question, context, and the

*Table 2.* Accuracy of all models on answerable and unanswerable questions.

| Model Name | | SL | | SL-R | | TS | | **MTL** | |
|---|---|---|---|---|---|---|---|---|---|
| | GUIDE | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ |
| BART | Unanswerable | 37.93 | 40.52 | 42.71 | 42.15 | 35.76 | 39.95 | 32.36 | 44.32 |
| | Answerable | 88.11 | 89.40 | 91.58 | 90.35 | 87.64 | 88.70 | 88.76 | 90.32 |
| T5 | Unanswerable | 34.53 | 34.37 | 44.24 | 28.36 | 24.17 | 22.52 | 45.20 | 45.97 |
| | Answerable | 89.94 | 90.71 | 92.34 | 93.15 | 88.30 | 90.17 | 89.43 | 90.62 |

*Table 3.* Accuracy of all models across unanswerability types.

| Model Name | | SL | | SL-R | | TS | | MTL | |
|---|---|---|---|---|---|---|---|---|---|
| | GUIDE | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ |
| BART | Entity Swap | 30.158 | 36.34 | 36.006 | 33.5 | 28.362 | 34.586 | 27.401 | 42.731 |
| | Number Swap | 62.88 | 62.88 | 61.663 | 60.243 | 56.186 | 61.663 | 46.044 | 62.474 |
| | Negation | 69.193 | 68.215 | 67.97 | 67.114 | 67.603 | 68.459 | 56.234 | 64.792 |
| | Antonym | 39.949 | 38.766 | 37.077 | 36.655 | 35.304 | 39.358 | 30.489 | 47.381 |
| | Mutual Exclusion | 22.693 | 22.693 | 13.466 | 17.705 | 23.192 | 20.947 | 25.436 | 33.665 |
| | No Info | 14.198 | 18.473 | 49.465 | 53.893 | 16.183 | 20.305 | 22.442 | 29.923 |
| T5 | Entity Swap | 25.438 | 25.396 | 38.596 | 16.165 | 12.03 | 11.612 | 35.923 | 37.134 |
| | Number Swap | 60.649 | 62.88 | 60.851 | 59.634 | 55.578 | 58.417 | 67.545 | 67.342 |
| | Negation | 68.459 | 67.359 | 66.503 | 67.97 | 66.136 | 67.603 | 71.393 | 70.415 |
| | Antonym | 35.557 | 34.375 | 44.932 | 28.125 | 21.621 | 26.604 | 52.449 | 53.125 |
| | Mutual Exclusion | 20.947 | 20.448 | 7.481 | 14.214 | 9.725 | 10.723 | 35.91 | 34.663 |
| | No Info | 12.213 | 12.977 | 45.801 | 9.007 | 5.954 | 6.106 | 22.137 | 25.648 |

Stage 1 predicted cause to generate the suggested answerable question. This design tests whether explicit, pipelined reason prediction improves downstream reformulation quality relative to end-to-end learning.

## 5. Results and Discussion

We evaluated the proposed MTL framework against all baselines described in Section 4.4 (results reported in Tables 2 and 3). Our analysis is organized around the following research questions (RQs).

**RQ1.** *How does the inclusion of unanswerability cause labels for input questions as auxiliary information impact the performance of the Answerable Question Generation task?* As shown in Table 2, adding gold cause labels (SL-R) consistently improves generation accuracy over the unlabeled Single-Task baseline (SL), even *without* a guidance sentence. This demonstrates that explicit unanswerability cause supervision provides a strong and reliable auxiliary signal for answerable question generation across models.

When guidance is additionally provided, the effect of cause labels becomes more nuanced. At the aggregate level (Table 2), BART shows an improvement from SL+Guide to SL-R+Guide, whereas T5 exhibits a decrease in overall accuracy (from 34.37% to 28.36%). However, a closer examination at the unanswerability-type level (Table 3) reveals that this contrast is largely driven by data distribution effects rather than model architecture. In particular, adding guidance on top of cause labels leads to performance degradation across most unanswerability types for both BART and T5. BART's overall improvement arises primarily from a substantial gain on the dominant *No Information* category, which outweighs the degradations observed in other categories. In contrast, T5 does not show a comparable improvement on this category, causing the category-level degradations to be reflected directly in the aggregate score.

These findings indicate that the interaction between cause labels and guidance is sensitive to redundancy and category composition. When guidance already captures the primary reformulation signal, additional cause supervision may introduce overlapping or conflicting constraints. While model architecture can influence sensitivity to such redundancy, for example, T5's text-to-text formulation represents all inputs as a single flat sequence and

may amplify the impact of longer concatenated inputs in smaller variants [41], the observed degradation is primarily explained by the distribution of unanswerability types rather than architectural limitations.

Overall, gold cause labels substantially aid answerable question generation, especially for originally unanswerable items. However, such labels are rarely available at inference time; thus the SL-R and SL-R+GUIDE settings should be viewed as upper bounds under oracle label access. To approximate realistic deployment, we therefore introduce Two-Stage models (TS) that first classify an input question into one of seven unanswerability causes and then condition generation on the predicted label.

Comparing SL to TS in Table 2 shows that TS generally underperforms SL. This suggests that noise in the Stage-1 cause classifier can propagate and mislead the Stage-2 generation module. Adding guidance does not resolve the issue: TS+GUIDE performs similarly to, or worse than, SL+GUIDE (notably in T5-based models), indicating that predicted labels plus guidance do not reliably translate into better reformulations. Finally, TS+GUIDE remains well below the oracle SL-R+GUIDE condition, highlighting a substantial gap attributable to imperfect label prediction. Together, these results point to the need for architectures that integrate cause modeling and generation more tightly, motivating our proposed MTL approach to jointly learn detection and suggestion while mitigating error propagation.

**RQ2.** *Does jointly modeling classification and generation in a multi-task learning (MTL) framework improve Answerable Question Generation (AQG) relative to Single-Task and Two-Stage approaches?*

As shown in Table 2, our MTL+GUIDE models consistently outperform all guided baselines (i.e., SL+GUIDE, SL-R+GUIDE, and TS+GUIDE) under both BART- and T5-based implementations. These gains indicate that learning to predict unanswerability causes *jointly* with question generation provides more useful training signal than supplying labels as static inputs (Single-Task) or passing noisy predictions in a pipeline (Two-Stage), thereby yielding more effective reformulations of unanswerable questions. Notably, all models achieve above 85% accuracy on answerable questions, indicating that the MTL framework enhancements do not compromise performance on answerable inputs, thereby demonstrating its overall effectiveness.

**RQ3.** *Which unanswerability classes benefit most from the predicted label signal in MTL+GUIDE framework, and which classes remain challenging despite its inclusion?* The effectiveness of incorporating predicted unanswerability labels within the Answerable Question Generation (AQG) step in the MTL+GUIDE framework varies notably across different unanswerability types, as evidenced by the results in Table 3. Categories such as *Entity Swap*, *Antonym*, *Mutual Exclusion*, and *No Information* consistently benefit from the joint modeling of classification and generation, with label supervision substantially improving the quality of question reformulation. Notably, in the *No Information* category, the highest performance is achieved by the SL-R models, which assume access to gold unanswerability labels. While this represents an idealized scenario not feasible in real-world applications, it underscores the potential upper bound of performance when precise labels are available. Although the MTL models, operating under realistic constraints with predicted labels, outperform all other settings, it still falls short of surpassing the SL-R models in this specific category. In contrast, for the *Number Swap* class, the T5-based MTL models outperform their SL, SL-R, and TS counterparts, suggesting improved handling of numerical reasoning under multi-task learning. In contrast, the BART-based MTL+GUIDE model shows only marginal gains over SL-R and performs similarly to SL models, indicating limited benefit in this setting. For Negation, the T5-based MTL models again achieve the highest accuracy among T5 variants, reflecting its advantage with guided multi-source inputs. However, the BART-based MTL models underperform compared to all other BART configurations, revealing their relative weakness in capturing negation-driven unanswerability. Collectively,

*Table 4.* Human evaluation.

| | **Ans.** | **Flu.** | **Rel.** | **Add.** |
|---|---|---|---|---|
| Entity Swap | 100 | 4.82 | 4.13 | 3.48 |
| Number Swap | 100 | 4.83 | 4.57 | 3.77 |
| Negation | 100 | 4.92 | 4.62 | 4.40 |
| Antonym | 100 | 4.60 | 4.58 | 3.97 |
| Mutual Exclusion | 100 | 4.97 | 4.48 | 3.95 |
| No Information | 95 | 4.88 | 4.58 | 4.80 |
| *All* | 99.17 | 4.84 | 4.49 | 4.06 |

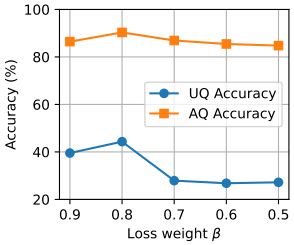


*Figure 2.* Sensitivity analysis.

these findings suggest that the effectiveness of guided multi-task learning is influenced by both the model architecture and the nature of the unanswerability. T5 consistently benefits from this setup, particularly in challenging cases like *Negation* and *Number Swap*, whereas BART exhibits more limited and task-sensitive gains.

**RQ4.** *What is the impact of incorporating the guidance sentence into the Answerable Question Generation task?* As shown in Table 2, the comparison of each model variant with and without the guidance sentence indicates that removing the guidance sentence decreases accuracy in the answerable question generation task for unanswerable questions, as evidenced by the SL model's performance drop without guidance. Moreover, the enhanced results for the TS and MTL models with guidance further emphasize its significance in generating answerable questions.

### 5.1. Qualitative Analysis

Table 5 presents representative qualitative examples comparing MTL+GUIDE and SL+GUIDE. In the example#a, where unanswerability arises from contextual mismatches such as incorrect temporal references, MTL+GUIDE applies targeted edits that resolve the underlying inconsistency, producing fluent and context-supported reformulations. In contrast, SL+GUIDE often preserves the original mismatch. Example#b illustrates scenarios where SL+GUIDE performs comparably or better, typically when the error is simple and the guidance sentence alone is sufficient for correction. Overall, these examples suggest that explicit cause conditioning is most beneficial for more complex unanswerability cases, while guidance-only reformulation remains effective for simpler instances.

*Table 5.* Examples where the MTL+Guide model outperforms the SL+Guide model (a) and vice versa (b). Note: Guidance sentences are shown in italics.

| | |
|---|---|
| **Example#a** | |
| **Input** | **Context.** *In the 1910s, New York-based filmmakers were attracted to Jacksonville's warm climate, exotic locations, excellent rail access, and cheap labor.* Over the course of the decade, more than 30 silent film studios were established, earning Jacksonville the title of .... |
| | **Unanswerable Question.** Is what industry was attracted to Jacksonville in the early nineteenth century? |
| **Output** | **MTL+Guide.** Is what industry was attracted to Jacksonville in the 1910s?<br>**SL+Guide.** Is what industry was attracted to Jacksonville in the early nineteenth century? |
| **Example#b** | |
| **Input** | **Context.** *NASA's CALIPSO satellite has measured the amount of dust transported by wind from the Sahara to the Amazon: an average 182 million tons of dust are windblown out of the Sahara each year*, at 15 degrees west longitude, across 1,600 miles (2,600 km) over the Atlantic Ocean ... |
| | **Unanswerable Question.** How many tons of dust are blown from the rain forest each year? |
| **Output** | **MTL+Guide.** How many tons of dust are blown from the rain forest each year?<br>**SL+Guide.** How many tons of dust are blown from the Sahara each year? |

### 5.2. Sensitivity Analysis

We conduct a sensitivity analysis to examine the effect of loss-weight configurations $(\alpha, \beta)$ under the constraint $\alpha + \beta = 1$. Figure 2 reports accuracy on unanswerable (UQ) and

answerable (AQ) questions while varying $\beta$, the weight of the generation objective. UQ performance peaks at $\beta = 0.8$, indicating that prioritizing generation improves recovery from unanswerable inputs, while AQ accuracy remains relatively stable across configurations. We exclude configurations where $\alpha$ dominates $\beta$, as our design treats unanswerability classification as an auxiliary task and assigning it a higher weight was observed to detract from generation quality. Overall, these results demonstrate that MTL+GUIDE is robust to reasonable variations in loss weighting, and that the selected configuration $(\alpha, \beta) = (0.2, 0.8)$ strikes an effective balance between classification guidance and generation quality.

### 5.3. User Study

To assess the quality of the answerable questions generated by the best-performing MTL+GUIDE model, we conducted a human evaluation with three fluent and native English-speaking annotators. The evaluation considered four criteria: *Answerability (Ans.)*, *Fluency (Flu.)*, *Relevance (Rel.)*, and *Addressing the cause of unanswerability (Add.)*. Annotators were provided with written guidelines and illustrative examples and completed the task independently. We randomly sampled 120 unanswerable questions for which the model generated answerable reformulations, ensuring balanced coverage across unanswerability types (20 per class). For each instance, annotators were shown the original question, its context, the unanswerability cause, and the generated reformulation. Fluency, relevance, and effectiveness were rated on a five-point Likert scale, while answerability was determined via majority voting. Final scores were obtained by averaging annotator ratings and are reported in Table 4. Overall, the generated questions were judged answerable in 99.17% of cases and achieved consistently high scores across all evaluation dimensions.

## 6. Conclusion

In this work, we address a key limitation of current MRC systems: the lack of constructive recovery when user questions are unanswerable. We propose a cause-conditioned multi-task learning framework that jointly models fine-grained unanswerability classification and answerable question generation. By conditioning generation on predicted failure causes and contextual guidance, the model produces targeted, context-supported reformulations rather than superficial paraphrases. Experiments across automatic metrics, LLM-based evaluation, and human judgments show that the proposed framework consistently outperforms single-task and two-stage baselines without compromising performance on answerable questions. These results demonstrate that explicitly modeling unanswerability causes provides a practical inductive bias for generating fluent and effective follow-up questions.

## References


[1] H. Moradisani, F. Zarrinkalam, Z. Noorian, and F. Ensan. "Exploring unanswerability in machine reading comprehension: approaches, benchmarks, and open challenges". In: *Artificial Intelligence Review* 59.1 (2025), p. 23.

[2] R. Baradaran, R. Ghiasi, and H. Amirkhani. "A survey on machine reading comprehension systems". In: *Natural Language Engineering* 28.6 (2022), pp. 683–732.

[3] H. A. Pandya and B. S. Bhatt. "Question answering survey: Directions, challenges, datasets, evaluation matrices". In: *arXiv preprint arXiv:2112.03572* (2021).

[4] S. Ouyang, Z. Zhang, and H. Zhao. "Fact-Driven Logical Reasoning for Machine Reading Comprehension". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 17. 2024, pp. 18851–18859.

[5] C. Tan, F. Wei, Q. Zhou, N. Yang, W. Lv, and M. Zhou. "I know there is no answer: Modeling answer validation for machine reading comprehension". In: *Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part I 7*. Springer. 2018, pp. 85–97.

[6] P. Rajpurkar, R. Jia, and P. Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 784–789.

[7] J. Liao, X. Zhao, J. Zheng, X. Li, F. Cai, and J. Tang. "Ptau: Prompt tuning for attributing unanswerable questions". In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022, pp. 1219–1229.

[8] P. Faldu, I. Bhattacharya, et al. "RetinaQA: A Robust Knowledge Base Question Answering Model for both Answerable and Unanswerable Questions". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2024, pp. 6643–6656.

[9] N. Kim, P. M. Htut, S. R. Bowman, and J. Petty. "$(QA)^2$: Question Answering with Questionable Assumptions". In: *The 61st Annual Meeting Of The Association For Computational Linguistics*. 2023.

[10] M. Hu, F. Wei, Y. Peng, Z. Huang, N. Yang, and D. Li. "Read+ verify: Machine reading comprehension with unanswerable questions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6529–6537.

[11] Z. Zhang, J. Yang, and H. Zhao. "Retrospective reader for machine reading comprehension". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 16. 2021, pp. 14506–14514.

[12] D. Fu, A. Godbole, and R. Jia. "SCENE: Self-Labeled Counterfactuals for Extrapolating to Negative Examples". In: *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[13] F. Sun, L. Li, X. Qiu, and Y. Liu. "U-net: Machine reading comprehension with unanswerable questions". In: *arXiv preprint arXiv:1810.06638* (2018).

[14] S. Back, S. C. Chinthakindi, A. Kedia, H. Lee, and J. Choo. "NeurQuRI: Neural question requirement inspector for answerability prediction in machine reading comprehension". In: *International Conference on Learning Representations*. 2020.

[15] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang. "SG-Net: Syntax-guided machine reading comprehension". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05. 2020, pp. 9636–9643.

[16] A.-Z. Yen, H.-H. Huang, and H.-H. Chen. "Unanswerable question correction in question answering over personal knowledge base". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 16. 2021, pp. 14266–14275.

[17] A.-Z. Yen, H.-H. Huang, and H.-H. Chen. "Unanswerable Question Correction and Explanation over Personal Knowledge Base". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, pp. 4645–4649.

[18] P. Faustini, Z. Chen, B. Fetahu, O. Rokhlenko, and S. Malmasi. "Answering Unanswered Questions through Semantic Reformulations in Spoken QA". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*. 2023, pp. 729–743.

[19] S. Hu, Y. Luo, H. Wang, X. Cheng, Z. Liu, and M. Sun. "Won't Get Fooled Again: Answering Questions with False Premises". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 5626–5643.

[20] W. Zhao, G. Gao, C. Cardie, and A. M. Rush. "I Could've Asked That: Reformulating Unanswerable Questions". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 4207–4220.

[21] Q.-W. Zhang, F. Li, J. Wang, L. Qiao, Y. Yu, D. Yin, and X. Sun. "FactGuard: Leveraging Multi-Agent Systems to Generate Answerable and Unanswerable Questions for Enhanced Long-Context LLM Extraction". In: *arXiv preprint arXiv:2504.05607* (2025).

[22] S. Kongyoung, C. Macdonald, and I. Ounis. "Multi-Task Learning of Query Generation and Classification for Generative Conversational Question Rewriting". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pp. 13667–13678.

[23] Y. Deng, W. Zhang, W. Xu, W. Lei, T.-S. Chua, and W. Lam. "A unified multi-task learning framework for multi-goal conversational recommender systems". In: *ACM Transactions on Information Systems* 41.3 (2023), pp. 1–25.

[24] C. Ding, Y. Hong, and J. Yao. "SGCM: Salience-Guided Context Modeling for Question Generation". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 2024, pp. 14755–14762.

[25] T. Ide and D. Kawahara. "Multi-Task Learning of Generation and Classification for Emotion-Aware Dialogue Response Generation". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. 2021, pp. 119–125.

[26] J. Kim, J. Park, C. Jeon, J. Choi, K. Kim, M. Hong, and S. Kim. "Chunk Knowledge Generation Model for Enhanced Information Retrieval: A Multi-task Learning Approach". In: *arXiv e-prints* (2025), arXiv–2509.

[27] M Lewis. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461* (2019).

[28] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of machine learning research* 21.140 (2020), pp. 1–67.

[29] Y. Zhang and B. Shen. "Prefix-tuning-based Consistency Learning Framework for Machine Reading Comprehension with unanswerable questions". In: *Computers and Electrical Engineering* 127 (2025), p. 110567.

[30] G. Gao, H.-T. Chen, Y. Artzi, and E. Choi. "Continually Improving Extractive QA via Human Feedback". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 406–423.

[31] P. Kirichenko, M. Ibrahim, K. Chaudhuri, and S. J. Bell. "AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions". In: *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

[32] H. Moradisani, F. Zarrinkalam, J. Serbanescu, and Z. Noorian. "UnAnswGen: A Systematic Approach for Generating Unanswerable Questions in Machine Reading Comprehension". In: *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 2024, pp. 280–286.

[33] G. Lee, S.-w. Hwang, and H. Cho. "SQuAD2-CR: Semi-supervised annotation for cause and rationales for unanswerability in SQuAD 2.0". In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. 2020, pp. 5425–5432.

[34] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, et al. "A survey on llm-as-a-judge". In: *arXiv preprint arXiv:2411.15594* (2024).

[35] X. Ho, J. Huang, F. Boudin, and A. Aizawa. "LLM-as-a-Judge: Reassessing the Performance of LLMs in Extractive QA". In: *arXiv preprint arXiv:2504.11972* (2025).

[36] S. Ma, H. Peng, L. Hou, and J. Li. "MRCEval: A Comprehensive, Challenging and Accessible Machine Reading Comprehension Benchmark". In: *arXiv preprint arXiv:2503.07144* (2025).

[37] J. Guan, J. Dodge, D. Wadden, M. Huang, and H. Peng. "Language models hallucinate, but may excel at fact verification". In: *Proceedings of the 2024 conference of the North American chapter of the association for computational linguistics: human language technologies (volume 1: long papers)*. 2024, pp. 1090–1111.

[38] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 46595–46623.

[39] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. "Gpt-4o system card". In: *arXiv preprint arXiv:2410.21276* (2024).

[40] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[41] S. Guo, L. Liao, C. Li, and T.-S. Chua. "A survey on neural question generation: Methods, applications, and prospects". In: *arXiv preprint arXiv:2402.18267* (2024).