Answering Ambiguous Questions through Generative Evidence Fusion and Round-Trip Prediction

Yifan Gao[†][*], **Henghui Zhu**[‡], **Patrick Ng**[‡], **Cicero Nogueira dos Santos**[‡], **Zhiguo Wang**[‡], **Feng Nan**[‡], **Dejiao Zhang**[‡], **Ramesh Nallapati**[‡], **Andrew O. Arnold**[‡], **Bing Xiang**[‡]

[‡]AWS AI [†] The Chinese University of Hong Kong

[†]yifangao95@gmail.com [‡]{henghui,patricng,cicnog,zhiguow}@amazon.com

[‡]{nanfen,dejiaoz,rnallapa,anarnld,bxiang}@amazon.com

Abstract

In open-domain question answering, questions are highly likely to be ambiguous because users may not know the scope of relevant topics when formulating them. Therefore, a system needs to find possible interpretations of the question, and predict one or multiple plausible answers. When multiple plausible answers are found, the system should rewrite the question for each answer to resolve the ambiguity. In this paper, we present a model that aggregates and combines evidence from multiple passages to adaptively predict a single answer or a set of question-answer pairs for ambiguous questions. In addition, we propose a novel round-trip prediction approach to iteratively generate additional interpretations that our model fails to find in the first pass, and then verify and filter out the incorrect question-answer pairs to arrive at the final disambiguated output. Our model, named REFUEL, achieves a new state-of-the-art performance on the AMBIGQA dataset, and shows competitive performance on NQ-OPEN and TriviaQA. The proposed round-trip prediction is a model-agnostic general approach for answering ambiguous open-domain questions, which improves our REFUEL as well as several baseline models. We release source code for our models and experiments at https://github.com/amzn/refuel-open-domain-qa.

Prompt Question (Google search query): What's the most points scored in an NBA game?

Disambiguated QA Pairs:

$\mathbf{Q}_1$: What's the most points scored in an NBA game by combined team? / $\mathbf{A}_1$: 370

$\mathbf{Q}_2$: What's the most points scored in an NBA game by a single team? / $\mathbf{A}_2$: 186

$\mathbf{Q}_3$: What's the most points scored in an NBA game by an individual? / $\mathbf{A}_3$: 100

Relevant Wikipedia Page 1: The highest-scoring regular season game is the triple-overtime game between ... the two teams combined to score 370 points, with the pistons defeating the nuggets 186–184 ...

Relevant Wikipedia Page 2: Wilt Chamberlain scored an nba-record 100 points ...

Figure 1: An example from the AMBIGQA (Min et al., 2020) dataset. The **Prompt Question** is gathered from Google search queries and has three interpretations upon reading Wikipedia. **Disambiguated QA Pairs** are the full set of acceptable answers, paired with the disambiguated rewriting of the prompt question.

1 Introduction

Open-domain Question Answering (QA) is the task of answering questions using a collection of passages with diverse topics (Chen et al., 2017; Guu et al., 2020; Karpukhin et al., 2020). Open-domain questions are highly likely to be ambiguous because people may not have the knowledge of relevant topics when formulating them. For example, in Figure 1, the prompt question *"What's the most points scored in an NBA game?"* is ambiguous because the *score* in this question could be interpreted as *the combined score in a game* ($\mathrm{Q}_1\mathrm{A}_1$), *score from a single team* ($\mathrm{Q}_2\mathrm{A}_2$), or *score from an individual player* ($\mathrm{Q}_3\mathrm{A}_3$). Therefore, a system needs to adaptively predict a single answer, or a set of equally plausible answers when the question has multiple interpretations. When a set of multiple answers is predicted, an unambiguous rewriting of the question that leads to each answer should also be provided to clarify each interpretation.

Min et al. (2020) decompose this problem into two subtasks. Given the prompt question and Wikipedia passages, the first subtask, **Answer Prediction**, consists in predicting one or several plausible answers, depending on whether this question is ambiguous or not. If multiple answers are predicted, the second subtask, **Question Disambiguation**, requires generating a disambiguated question for each of the plausible answers. They propose SPANSEQGEN, which first retrieves and reranks

[*]Work done during an internship at AWS AI.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 3263–3276
August 1–6, 2021. ©2021 Association for Computational Linguistics

passages using the prompt question, and then adopts a BART pre-trained sequence-to-sequence model (Lewis et al., 2020a) to generate all plausible answers, conditioned on the concatenation of the prompt question and top 8 passages. For the question disambiguation subtask, based on BART, they first pre-train a question generation model on NQ-OPEN (Kwiatkowski et al., 2019), a large-scale open-domain QA dataset, to generate the question given the answer and top 8 passages. Then they fine-tune it as a question disambiguation model to generate the disambiguated question conditioned on the prompt question, answer, and passages.

There are three main drawbacks to SPANSEQGEN. Firstly, a complete coverage of all relevant passages is essential for predicting all plausible answers of the ambiguous question. However, SPANSEQGEN only takes 8 passages for answer prediction so some of the most informative passages might be excluded. Secondly, for the question disambiguation subtask, there is a mismatch between question generation pre-training on NQ-OPEN and question disambiguation fine-tuning on AMBIGQA – there is no question to disambiguate in question generation pre-training, which makes the pre-training task somewhat misaligned with fine-tuning. Thirdly, SPANSEQGEN predicts a much smaller average number of answers compared to the ground truth data (1.17 *vs.* 2.19).

To address these issues, we propose REFUEL, Round-trip Evidence FUsion via gEneration with retrievaL, a new framework for answering ambiguous open-domain questions. To ensure a broad coverage of relevant knowledge of the question, REFUEL reads 12 times more passages (100 in our experiments) than SPANSEQGEN by using Fusion-in-Decoder (Izacard and Grave, 2020) that processes each passage individually in the encoder, and then fused their encodings together in the decoder. For the question disambiguation subtask, we propose a *token-deletion pre-training task* to transform NQ-OPEN into an "ambiguous" QA setting by randomly deleting an informative span for each question. Thus, pre-training and fine-tuning tasks are well aligned. Additionally, we add an *insertion-based weighted loss* to emphasize the newly inserted tokens in the disambiguated question, which helps the model on learning to resolve the ambiguity. Finally, we propose a *round-trip prediction approach* to find additional interpretations that REFUEL fails to predict in the first pass. We continuously feed the generated questions into REFUEL until there are no new answers predicted from our model. While this round-trip prediction can improve the recall of answers, we refine the quality of predicted QA pairs by filtering them with the conditional probability of the answers estimated by an answer-generation model.

Our REFUEL achieves a new state-of-the-art on the AMBIGQA dataset, outperforming the previous best model SPANSEQGEN by 9.1% in answer prediction F1 and 4.4% in Edit-F1 score for question disambiguation. When directly doing inference on NQ-OPEN and TriviaQA, REFUEL not only predicts the single answer precisely but also finds multiple interpretations if the question is ambiguous. Moreover, human evaluation shows that REFUEL can correctly generate more QA pairs on all three datasets. Finally, the proposed round-trip prediction is a model-agnostic general approach for answering ambiguous questions, which improves our REFUEL as well as several baseline models up to 3.7% for the overall performance.

The main contributions of this work, which are fundamental to significantly push the state-of-the-art in answering ambiguous questions, can be summarized as follows:

1. We present an evidence aggregation approach that can effectively use a large number of passages to uncover more candidate interpretations of the ambiguous question.
2. We propose a token-deletion pre-training task to reduce the mismatch between pre-training and fine-tuning for question disambiguation. The insertion-based weighted loss further helps to capture answer-relevant constraints.
3. We propose a round-trip prediction approach to find more interpretations missed in the first prediction pass, which we further refine using a conditional-probability-based filtering approach.

2 REFUEL

REFUEL answers questions through a three-step process illustrated in Figure 2:

1. The *Passage Retrieval & Reranking* module retrieves question-relevant passages from the whole Wikipedia corpus. Then the retrieved passages are further reranked (Sec. 2.1).
2. Taking the reranked passages and the prompt question as input, our single pass QA pair generation model makes the first prediction

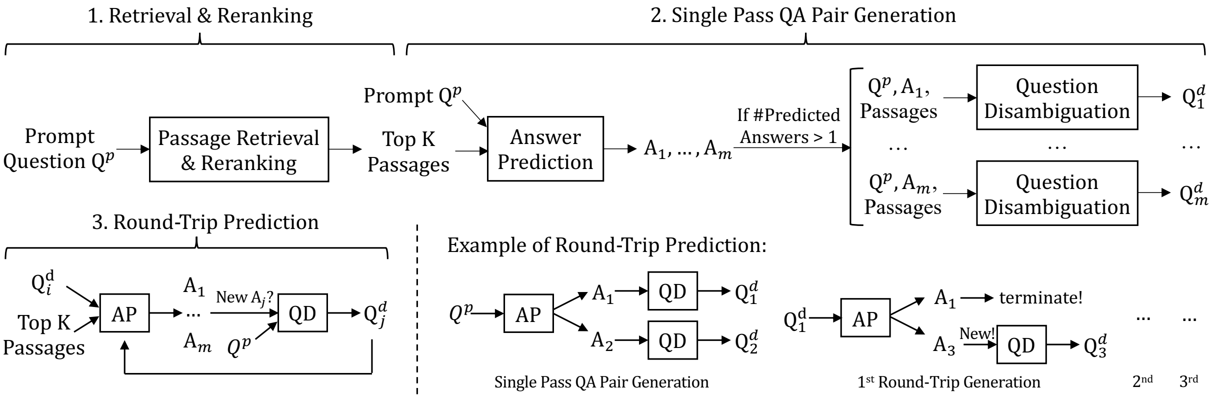


Figure 2: Overall Pipeline of REFUEL. REFUEL firstly retrieves question-relevant passages (Section 2.1). Then it generates first-pass QA pairs through the Answer Prediction (AP) module and Question Disambiguation (QP) module (Section 3). Finally, generated disambiguated questions Q^d are further taken as the input of our pipeline to find more interpretations (Round-Trip Prediction). If the generated question Q^d still has multiple interpretations, the newly predicted answers will receive their own questions (Section 2.3).

pass to predict a single answer or a set of disambiguated QA pairs (Sec. 2.2).

3. Our proposed *Round-Trip Prediction* can find more interpretations missed in the first prediction pass, which we further refine using a conditional-probability-based filtering approach (Sec. 2.3).

2.1 Passage Retrieval & Reranking

We use Dense Passage Retriever (DPR) (Karpukhin et al., 2020) for retrieval. First, we split all Wikipedia pages into 100-token passages, resulting in 24M passages in total. Then DPR maps all passages into d-dimensional vectors, computes the representation of the prompt question, and retrieves N passages whose vectors are closest to the question vector (we use N=1000).

After retrieving N passages for the prompt question, we fine-tune BERT (Devlin et al., 2019) to rerank these passages. Taking the concatenation of the prompt question and each passage as input, the reranker allows a token-level cross-attention between the prompt question and passages. The relevance score is then derived by taking the `[CLS]` vector of the input sequence into a linear layer. After reranking, the QA pair generation model takes the top K passages as inputs (we use K=100).

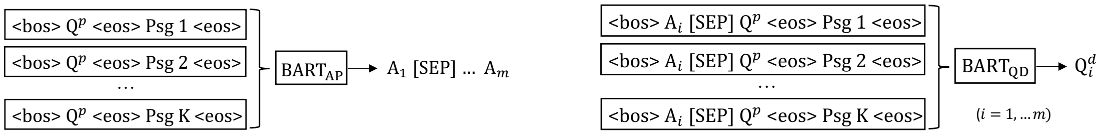
2.2 Single Pass QA Pair Generation

The single pass QA pair generation step includes an Answer Prediction module and a Question Disambiguation module. Firstly, taking the reranked passages and the prompt question Q^p as input, the Answer Prediction module generates one or multiple plausible answers $A_1, ..., A_m$. If multiple plausible answers are found, the prompt question is treated as ambiguous so that the Question Disambiguation module generates a disambiguated question Q_i^d for each predicted answer A_i. Note that our general pipeline in Figure 2 does not limit the implementation of Answer Prediction module and Question Disambiguation module, and it can work for our REFUEL as well as several baselines (shown in Sec. 4.3). Our implementation is detailed in Sec. 3.

2.3 Round-Trip Prediction

During answering ambiguous questions, it might be difficult to find every possible interpretation in the first prediction pass, and existing work (Min et al., 2020) predicts 47% less answers compared with the ground truth. Therefore, we propose *round-trip prediction*, which includes a Round-Trip Generation step and a Language Model Verification Step.

Round-Trip Generation. Keeping the same retrieved passages, we continuously feed the generated disambiguated questions into the Answer Prediction module to check if any new answers are generated, and generate their corresponding disambiguated questions until there are no newly predicted answers. As exemplified in Figure 2, $(\mathrm{Q}_1^d, \mathrm{A}_1), (\mathrm{Q}_2^d, \mathrm{A}_2)$ are two disambiguated QA pairs of the ambiguous prompt question Q^p after the first prediction pass. When feeding Q_1^d to the Answer Prediction module again (1st Round-Trip Prediction), we find that besides the previously predicted answer A_1, a new answer candidate A_3 is predicted. Then we generate its corresponding question Q_3^d accordingly. This loop continues until


<bos> Q^p <eos> Psg 1 <eos>
<bos> Q^p <eos> Psg 2 <eos>
...
<bos> Q^p <eos> Psg K <eos>
BART_AP
A_1 [SEP] ... A_m
<bos> A_i [SEP] Q^p <eos> Psg 1 <eos>
<bos> A_i [SEP] Q^p <eos> Psg 2 <eos>
...
<bos> A_i [SEP] Q^p <eos> Psg K <eos>
BART_QD
Q_i^d
(i = 1, ...m)

(a) Answer Prediction Module

(b) Question Disambiguation Module

Figure 3: The architecture of single pass QA pair generation in REFUEL.

there are no newly predicted answers.

Language Model Verification. Through the Round-Trip Generation, we generate a bunch of QA pairs from the ambiguous prompt question, but some of them are incorrect. Here we adopt a verification process to filter out these incorrect predictions. Recent works in synthetic QA pair generation (Alberti et al., 2019; Puri et al., 2020) use an "Exact Match (EM) Verification" approach to prune the QA pairs. They separately train a QA model as the verification model, and drop the predicted (q, a) when the verification model's answer $a' \neq a$. However, this EM Verification approach is only suitable for factoid reading comprehension tasks such as SQuAD (Rajpurkar et al., 2016), in which the QA model has near-human accuracy so that it will not falsely filter out too many correct QA pairs. In open-domain QA, the current best model can only have 51.4% EM accuracy on the NQ-OPEN dataset (Izacard and Grave, 2020).

Instead of using hard filtering, we employ a "Language Model (LM) Verification" approach that is similar to the *LM filtering* method of Shakeri et al. (2020). LM Verification is a conditional-probability-based approach to filter out QA pairs softly. In "LM Verification", we first train a conditional language model using the gold disambiguated QA pairs from AMBIGQA. The conditional language model is trained to estimate the likelihood of an answer given the golden disambiguated question. Once training is done, it is used to score the generated QA pair (q, a) from REFUEL, which is the likelihood of the answer a given the question q and passages,

$$\text{LM score} = \Sigma_{i=1}^{N_a} \log p(a^i | q, \text{passages}), \qquad (1)$$

where N_a is the length of the generated answer. Finally, we rerank all predicted QA pairs according to the LM score, and drop the QA pairs according to a threshold Th $= 6.1$. The threshold is tuned according using the development set.

3 Single Pass QA Pair Generation Details

3.1 Answer Prediction

SPANSEQGEN (Min et al., 2020) concatenates the prompt question and top reranked passages into a single sequence for BART encoding, which is extremely limited by the maximum input sequence length of BART (1024 subwords, equivalent to 8 passages). Consequently, SPANSEQGEN finds fewer interpretations of the prompt question compared to the ground truth (1.17 vs 2.19). To ensure a broad coverage of retrieved & reranked passages, our Answer Prediction module uses the Fusion-in-Decoder approach (Izacard and Grave, 2020), which allows us to scale the number of processed passages. As shown in Figure 3, our BART-based Answer Prediction module BART$_{\text{AP}}$ encodes the concatenation of the prompt question and each passage independently. Then all encoded token-level representations are concatenated into a single sequence, and the BART$_{\text{AP}}$ decoder performs attention over all passages to aggregate and combine evidence. Finally, the BART$_{\text{AP}}$ decoder generates a sequence of plausible answers token-by-token, separated by [SEP]. Since there is no cross-passage attention in the encoder, BART$_{\text{AP}}$ encoder reduces the computation from quadratic in the number of input passages to linear complexity. As a result, it can process 12 times larger number of input passages (up to 100 passages, 16000 subwords) than SPANSEQGEN. Given that AMBIGQA is a small dataset with only 10k training samples, we first pre-train BART$_{\text{AP}}$ on NQ-OPEN to predict a single answer, then fine-tune it on AMBIGQA to predict one or multiple answers.

3.2 Question Disambiguation

If multiple answers are predicted, the Question Disambiguation module is activated to generate a disambiguated rewriting of the prompt question for each predicted answer. Because we do not know which input passage is the key evidence to derive the predicted answer, the Question Disambiguation module takes the same passages in the Answer Prediction stage as inputs. Similar to the Answer Prediction module BART_{AP}, our Question Disambiguation module BART_{QD} processes the inputs under the same fashion except that BART_{QD} encoder additionally takes the predicted answer A_i from BART_{AP} in the input (shown in Figure 3).

Token-Deletion Pre-training. Similar to the training scheme of the Answer Prediction module, we also want to leverage the large-scale NQ-OPEN data for pre-training. One straightforward way is to train a question generation model on NQ-OPEN that generates questions given the passages and answer, and then fine-tune it for question disambiguation on AMBIGQA given the prompt question, answer, and passages. However, there is no input question to disambiguate in the question generation pre-training task, it leads to a mismatch between pre-training and fine-tuning. Ablation study shows this way of pre-training has almost no help for question disambiguation (Section 4.5).

To reduce the mismatch issue between pre-training and fine-tuning, we propose a Token-Deletion Pre-training task. The idea is to construct synthetic ambiguous questions in pre-training to reduce the mismatch. Given a question Q from NQ-OPEN, we randomly delete an *informative span* from it, resulting in a partial question Q^s. This partial question is designed to simulate the ambiguous question Q^p in the fine-tuning stage. Then the token-deletion pre-training target is to recover the complete question Q from the partial question Q^s, answer, and passages. In this way, the token-deletion pre-training aligns the fine-tuning phase.

Prompt questions are usually rewritten by adding new constraints including event/entity references, properties, answer types, etc. For example, the disambiguated question Q_1 in Figure 1 inserts "by a combined team" after the ambiguous prompt question. Therefore, we define the *informative span* as the span containing at least one of the following Part-of-Speech tags: 'ADJ', 'NOUN', 'NUM', 'PROPN', 'SYM', 'VERB'. The length of the span is uniformly sampled in $[1, 5]$.

Insertion-based Weighted Loss. Since the disambiguated question is a small modification from the ambiguous prompt question, most tokens can be directly copied from the input. Here we introduce an insertion-based weighted loss to put more emphasis on the newly added tokens of the disambiguated question, which could be the key to disambiguate the prompt question. Given the prompt question Q^p, we find the newly inserted tokens from the disambiguated question Q^d: $\{q^{in}\}$. The final loss for fine-tuning BART_{QD} is a combination of the original negative log-likelihood loss on all question tokens augmented with a term that adds weight on the likelihood of inserted tokens:

$$\mathcal{L} = \mathcal{L}_{nll} - \lambda \sum_{q_j \in \{q^{in}\}} \log(q_j | A, Q^p, \text{Psg}), \quad (2)$$

where $\mathcal{L}_{nll} = \sum_{i=1}^{n} \log(q_i | A, Q^p, \text{Psg})$, n is the number of tokens in the disambiguated question, $\lambda = 3.5$ is a hyperparameter tuned on the dev. set.

4 Experiments

4.1 Experimental Setup

Dataset. We conduct main experiments on the AMBIGQA dataset (Min et al., 2020). AMBIGQA is constructed to address the ambiguity of questions in open-domain QA. It samples 14,042 questions from NQ-OPEN, a large-scale open-domain QA dataset in which each question has a single answer (Kwiatkowski et al., 2019), and asks annotators to search for, navigate and read multiple Wikipedia pages to find as many interpretations as possible. As a result, each question is annotated with either a single answer or multiple disambiguated QA pairs, depending on how many interpretations can be found. The train, development, and test (not public) dataset sizes are 10036, 2002, 2004, respectively [1]. On average, there are 2.1 distinct answers per question in AMBIGQA. To test the generalization ability of REFUEL on any possibly ambiguous questions, we additionally evaluate it on two open-domain QA datasets: NQ-OPEN and TriviaQA (Joshi et al., 2017).

Implementation Details are in Appendix A. We release source code for our models and experiments at https://github.com/amzn/refuel-open-domain-qa.

Evaluation Metrics. Let $(q_1, a_1), ..., (q_m, a_m)$ be m QA pair predictions, $(\hat{q}_1, \hat{a}_1)$, ..., $(\hat{q}_n, \hat{a}_n)$ be n gold QA pairs, each predicted QA pair (q_i, a_i) is evaluated in order by a *correctness score* towards all gold QA pairs: $c_i = \mathbb{1}(a_i{=}\hat{a}_j) f(q_i, \hat{q}_j)$, where $f(q_i, \hat{q}_j)$ is a similarity function for questions. $(\hat{q}_j, \hat{a}_j)$ will not be further used to evaluate

[1] Leaderboard: https://nlp.cs.washington.edu/ambigqa/leaderboard.html

Model	F1_{ans} (all)		F1_{ans} (multi)		F1_{BLEU}		$\text{F1}_{\text{EDIT-F1}}$		Comb.	
	dev	test	dev	test	dev	test	dev	test	dev	test
Disambig-first (Min et al., 2020)	28.1	24.8	21.9	18.8	4.2	4.0	2.7	2.2	30.8	27.0
DPR Reader (Min et al., 2020)	37.1	32.3	28.4	24.8	13.4	11.3	6.6	5.5	43.7	37.8
SpanSeqGen (Min et al., 2020)	39.7	33.5	29.3	24.5	13.4	11.4	7.2	5.8	46.9	39.3
Refuel w/o RTP (single model)	**48.4**	41.7	37.0	32.7	16.0	14.8	11.2	9.0	59.6	50.7
Refuel (single model)	48.3	**42.1**	**37.3**	**33.3**	**16.2**	**15.3**	**11.8**	**9.6**	**60.1**	**51.7**
SpanSeqGen (ensemble)	41.2	35.2	29.8	24.5	13.6	10.6	7.4	5.7	48.6	40.9
Refuel (ensemble)	**50.4**	**44.3**	**38.7**	**34.8**	**17.0**	**15.9**	**12.5**	**10.1**	**62.9**	**54.4**

Table 1: Results on the dev. and hidden test set of AmbigQA. "Refuel w/o RTP" is the single pass prediction model without using round-trip prediction. In addition to metrics introduced in Section 4.1, we also show a combined metric "Comb." = F1_{ans} (all) + $\text{F1}_{\text{EDIT-F1}}$ which is used to rank models on the official leaderboard.

other predicted QA pairs as it is used for (q_i, a_i). The overall correctness is calculated by F1 between predictions and references,

$$\text{P}_f = \frac{\sum_{i=1}^{m} c_i}{m}, \text{R}_f = \frac{\sum_{i=1}^{m} c_i}{n}, \text{F1}_f = \frac{2\text{P}_f\text{R}_f}{\text{P}_f + \text{R}_f}.$$

All examples are evaluated for the answer prediction subtask, in which f function always yields 1. This metric is denoted as F1_{ans} (all). For the subset of examples with multiple **gold** QA pairs, both answer prediction subtask and question disambiguation subtask are evaluated. The answer prediction metric only computed on this subset is denoted as F1_{ans} (multi). To evaluate question disambiguation performance, BLEU (Papineni et al., 2002) and EDIT-F1 is used for the function f, denoted as F1_{BLEU} and $\text{F1}_{\text{EDIT-F1}}$, respectively. EDIT-F1 compute the F1 score of added and deleted unigrams from the prompt question to the predicted disambiguated question towards references.

4.2 Experimental Results

Main Results. Performance on the dev. and hidden test set of AmbigQA is shown in Table 1. Even without having round-trip prediction, Refuel (w/o RTP) outperforms SpanSeqGen on both the answer prediction subtask and question disambiguation subtask by a large margin. Moreover, the round-trip prediction indeed further improves the performance by finding more and better QA pairs, going from 1.55 to 1.72 pairs per prompt question on the dev. set. A comprehensive analysis on the round-trip prediction is discussed in Sec 4.3.

Model	N	K	#QAs	F1_{ans}	$\text{F1}_{\text{EDIT-F1}}$
SpanSeqGen	100	≈8	1.17	39.7	7.2
SpanSeqGen*	100	≈8	1.14	41.7	7.1
Refuel (w/o RTP)	100	8	1.42	44.7	10.0
Refuel (w/o RTP)	100	100	1.54	45.4	10.7
Refuel (w/o RTP)	1000	100	1.55	48.4	11.2

Table 2: Dev. set results of AmbigQA as a function of the number of retrieval/reranking (N) and QA input (K) passages. #QAs: the average number of predicted QA pairs per prompt question. *: our replicated results.

Model	NQ-open		TriviaQA	
	EM	Oracle EM	EM	Oracle EM
ORQA (supervised)	33.3	-	45.0	-
HardEM (supervised)	28.1	-	50.9	-
DPR (supervised)	41.5	-	57.9	-
RAG (supervised)	44.5	-	56.8	-
Refuel w/o RTP (NFT)	35.4	45.2	48.2	52.9
Refuel (NFT)	37.3	48.9	49.8	54.3

Table 3: Results on NQ-open and TriviaQA test set. RTP: Round-Trip Prediction. NFT: No Fine-Tuning. ORQA (Lee et al., 2019), HardEM (Min et al., 2019), RAG (Lewis et al., 2020b).

Controlled Comparison with SpanSeqGen. Besides round-trip prediction, Refuel has two advantages over SpanSeqGen in terms of input passages: (1) We retrieve top N=1000 passages (instead of 100 in SpanSeqGen) to get a higher answer recall at top 100 passages (improved from 86.2 to 89.7). (2) Refuel takes K=100 input passages whereas SpanSeqGen takes at most 1024 subwords (K≈8). To establish a controlled and fair comparison, we remove the round-trip prediction part of Refuel, and feed Refuel (w/o RTP) with the same input passages used in SpanSeqGen (N=100, K=8). Results are shown in Table 2. We find (1) Under the same number of passages, Refuel (w/o RTP) (N=100, K=8) still outperforms SpanSeqGen and generates more and better QA pairs; (2) Refuel (w/o RTP) benefits from increasing the answer recall of retrieval stage ($\text{N} = 100 \rightarrow 1000$), as well as allowing more input passages ($\text{K} = 8 \rightarrow 100$).

Generalization to Other Datasets. To test how well does Refuel answer any open-domain questions, we evaluate Refuel on NQ-open and Triv-

Models	#QAs	F1_{ans} (all)	F1_{ans} (multi)	F1_{BLEU}	$\text{F1}_{\text{EDIT-F1}}$	Comb.
REFUEL w/o RTP	1.55	48.4	37.0	16.0	11.2	59.6
+ Round-Trip Generation	2.06 (↑33.5%)	47.6	37.4	16.0	11.4	59.0 (↓0.9%)
+ Round-Trip Generation & LM Verification	1.72 (↑11.1%)	48.3	37.3	16.2	11.8*	60.1 (↑0.7%)
+ Round-Trip Generation & EM Verification	1.43 (↓ 7.7%)	47.6	35.4	15.7	11.6	57.2 (↓4.0%)
DPR Reader	1.62	38.9	29.9	12.5	6.8	45.7
+ Round-Trip Generation & LM Verification	1.81 (↑11.7%)	40.1*	31.6*	13.3*	7.3*	47.4* (↑3.7%)
SPANSEQGEN	1.14	41.7	29.3	12.7	7.1	48.8
+ Round-Trip Generation & LM Verification	1.28 (↑12.3%)	42.4*	29.9*	13.0*	7.4*	49.8* (↑2.1%)

Table 4: Effect of *round-trip prediction* to harvest more interpretations (QA pairs) on the development set of AMBIGQA. "↑ and ↓" denotes the improvement gain over the model without round-trip prediction. *: The model with "Round-Trip Generation & LM Verification" is significantly better than the same model without it under a paired bootstrap test with 10^5 samples (*p-value* <0.05).

iaQA without finetuning on these datasets. When REFUEL predicts multiple answers, we take the first predicted answer for EM evaluation; we also introduce a new **Oracle EM** metric which treat the prediction is correct if the gold answer matches any predicted answers for the current question. Table 3 shows that REFUEL has competitive performance even without dataset-specific finetuning. When REFUEL finds multiple interpretations for questions in NQ-OPEN & TriviaQA, we manually check the quality of disambiguated QA pairs in Section 4.4.

Models	Dataset	#QAs	#C-QAs	#CD-QAs	κ
SPANSEQGEN	AMBIGQA	2.12	1.40	0.46	0.27
REFUEL w/o RTP	AMBIGQA	2.80	1.84	0.98	0.35
REFUEL	AMBIGQA	3.44	2.40	1.24	0.34
REFUEL w/o RTP	NQ-OPEN	2.32	1.30	0.64	0.20
REFUEL	NQ-OPEN	3.20	1.72	0.88	0.21
REFUEL w/o RTP	TriviaQA	2.08	1.02	0.46	0.34
REFUEL	TriviaQA	3.24	1.84	0.82	0.35

Table 5: Human evaluation results. #QAs: the average number of QA pairs per prompt question. #C-QAs & #CD-QAs: the average number of *correct* QA pairs, detailed in Sec. 4.4. κ: Fleiss' kappa score.

4.3 Effect of Round-Trip Prediction

We compare our proposed Round-Trip Prediction (Round-Trip Prediction = Round-Trip Generation + LM Verification) with several alternative approaches, as well as investigate its generalization ability to other models like SPANSEQGEN and DPR Reader. Results are shown in Table 4.

Round-Trip Generation Only. We investigate the necessity of the verification process by conducting only round-trip generation to REFUEL. Results show that Round-Trip Generation can generate 33.5% more QA pairs, but the lower F1_{ans} (all) suggests that this strategy may over-generate QA pairs when the prompt question is not ambiguous. Hence, the verification process is necessary to prune some incorrect QAs.

LM Verification *vs.* EM Verification. As described in section 2.3, we compare the existing EM Verification approach (Alberti et al., 2019; Puri et al., 2020) with our LM Verification. Results demonstrate that EM Verification prunes too many QA pairs – the number of remaining QA pairs (1.43) is even smaller than not doing round-trip prediction (1.55). This validates our intuition in section 2.3 that EM Verification is not suitable for open-domain QA tasks because of the low performance of current open-domain QA models.

Generalization to Other Models. We show that round-trip prediction is a model-agnostic general approach for answering possibly ambiguous open-domain questions by using it on our replicated baseline models: DPR Reader and SPANSEQGEN. With the help of round-trip prediction, DPR Reader and SPANSEQGEN generates 11.7% and 12.3% more QA pairs, which result in a boost of 3.7% and 2.1% for the overall performance (Comb.).

4.4 Human Evaluation

Since the answers collected in AMBIGQA are not necessarily exhaustive, there is a possibility that a model generates correct interpretations but they are missed in AMBIGQA. Therefore, we hire 3 workers from `MTurk.com` to evaluate the correctness of the answer given the generated disambiguated question and retrieved passages (instructions in Appendix C). Let $(q_1, a_1), ..., (q_n, a_n)$ be n generated QA pairs from the same prompt question, we define two levels of correctness as follows: **#C-QAs**: (q_i, a_i) is considered **C**orrect if a_i is a correct answer of q_i; **#CD-QAs**: (q_i, a_i) is considered correct iff. (1) a_i is a correct answer of q_i *and* (2) any $a_j (j \neq i)$ is a wrong answer of q_i. #CD-QAs is designed to examine the **C**orrectness of ques-

Pre-train Method + Fine-tune Method	F1$_{\text{BLEU}}$	F1$_{\text{EDIT-F1}}$
Prompt Baseline	18.9	0.0
None + QDF	16.2	10.1
None + QDF (w/ filtered passages)	16.4	9.4
QGP + QDF	15.9	10.3
TDP + QDF	16.5	10.9
TDP + QDF (w/ insertion-based loss)	16.0	**11.2**

Table 6: Ablation Study of REFUEL for the question disambiguation subtask on the dev. set. QDF: Question Disambiguation Fine-tuning, QGP: Question Generation Pre-training, TDP: Token-Deletion Pre-training.

tion **D**isambiguation because ambiguous questions can have multiple valid answers. We take the majority judgement from 3 annotators for each QA pair. For each dataset, we randomly sample 50 prompt questions which have multiple predicted answers, and apply the QA swapping strategy in #CD-QAs, resulting 960 question-answer-passages triples in total. Results in Table 5 show that REFUEL (w/o RTP) can correctly generate 113% more QA pairs than SPANSEQGEN on #CD-QAs. In addition, round-trip prediction (RTP) can find more correct interpretations across all datasets.

4.5 Ablations on Question Disambiguation

Table 6 compares our question disambiguation model with the prompt baseline and several ablations. The prompt baseline directly takes the prompt question as the disambiguated prediction, so its F1$_{\text{EDIT-F1}}$ is zero. However, F1$_{\text{BLEU}}$ score of the prompt baseline is higher than REFUEL. This suggests that F1$_{\text{EDIT-F1}}$ captures the effectiveness of question disambiguation better than F1$_{\text{BLEU}}$.

For our ablations, we start from only using AMBIGQA dataset (None+QDF), and investigate whether it is helpful to only use answer-containing passages as inputs (None+QDF w/ filtered passages). The worse result of the latter approach suggests that we should keep all passages for question disambiguation. Second, we examine the effectiveness of pre-training. We try the question generation pre-training (QGP+QDF) and compare it with the ablation without any pre-training (None+QDF). Results show that the question generation pre-training has little help for fine-tuning. By replacing the question generation pre-training QGP with our proposed token-deletion pre-training TDP, we see the results (TDP+QDF) are better than the no pre-training ablation (None+QDF), which implies the mismatch between pre-training and fine-tuning are somewhat reduced. Finally, the insertion-based loss enables REFUEL to capture the key disambiguation phrase with less copying the prompt question, resulting in a lower BLEU but higher Edit-F1.

Prompt question #1: What's the most points scored in an nba game?
Reference:
Q1: What is the highest amount of points scored by a single team in regular season NBA games? / A1: 186
Q2: What is the highest amount of points scored by a single team in regular season games in regulation? / A2: 162
Q3: What is the highest amount of points scored by a single team in playoff games? / A3: 153
REFUEL w/o RTP: (QA1-QA4: F1$_{\text{ans}}$=57.1, F1$_{\text{EDIT-F1}}$=44.9)
Q1: What's the most points scored in a regular season nba game by combined? / A1: 370
Q2: What's the most points scored in an nba playoff game by combined? / A2: 304
Q3: What's the most points scored in an nba game by individual? / A3: 100
Q4: What player scored the most points in an NBA game? / A4: wilt chamberlain
REFUEL: (QA1-QA6: F1$_{\text{ans}}$=66.7, F1$_{\text{EDIT-F1}}$=57.1)
Q5: What's the most points scored in an NBA game by single team? / A5: 186
Q6: What's the most points scored in an nba playoff game by single team? / A6: 153
Relevant Passages: (w/ rank from retrieval & reranking)
Rank 1: ... the highest-scoring regular season game is ... the two teams combined to score 370 points, with the pistons defeating the nuggets 186–184 ...
Rank 3: wilt chamberlain scored an nba-record 100 points. the highest-scoring playoff game is the double-overtime game between ... the two teams combined to score 304 points, with the trail blazers defeating the suns 153–151 ...

Figure 4: Predictions generated by REFUEL w/o round-trip prediction (QA1-QA4) and REFUEL (QA1-QA6).

4.6 Case Study

Figure 4 provides example question-answer pairs generated by crowd-workers, REFUEL (w/o RTP), and REFUEL. The annotator find three interpretations from the prompt question, while our single pass model REFUEL (w/o RTP) finds in total four interpretations (QA1-4). Although QA2 predicted from our model is not included in the references, it is indeed a correct interpretation of the prompt question. In addition, the Round-Trip Prediction approach finds two correct interpretations (QA5, QA6) which the model fails to predict on the first generation pass. More cases are shown in Appendix F.

5 Related Work

Open-Domain Question Answering is answering factoid questions using a huge collection of documents such as Wikipedia pages (Voorhees, 1999;

Chen et al., 2017; Yang et al., 2019; Lee et al., 2019; Wang et al., 2019). We are motivated by the recent proposed question ambiguity problem in open-domain QA (Min et al., 2020). Different from the existing formulation of open-domain QA that each question only has a single answer, the proposed AMBIGQA task requires to predict a single answer or a set of disambiguated QA pairs depending on the ambiguity of the input question. They also propose the first model SPANSEQGEN to this task, which firstly uses the dense passage retriever (Karpukhin et al., 2020) to retrieve question-relevant passages, and then adopts a retrieval-augmented generation method (Lewis et al., 2020b) to disambiguated QA pairs.

Our REFUEL follow Min et al. (2020)'s task formulation and overall pipeline, but there are three differences between our REFUEL and SPANSEQGEN: (1) REFUEL takes the architecture of Fusion-in-Decoder (Izacard and Grave, 2020) that can effectively use a large number of passages to uncover more candidate interpretations of the ambiguous question. (2) We propose a token-deletion pretraining task to reduce the mismatch between pretraining and fine-tuning for question disambiguation. The insertion-based weighted loss further helps to capture answer-relevant constraints. (3) We propose a model-agnostic round-trip prediction approach to find more interpretations missed in the first prediction pass, which we further refine using a conditional-probability-based filtering approach.

6 Conclusion

In this paper, we present REFUEL to answer ambiguous open-domain questions. REFUEL is a generative approach to aggregate and combine evidence from multiple passages for multiple rounds which can find more and better interpretations. REFUEL achieves a new state-of-the-art on AMBIGQA, and shows competitive performance on NQ-OPEN and TriviaQA. The proposed round-trip prediction is a general approach for answering ambiguous open-domain questions, which improves our REFUEL as well as several baseline models.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *International Conference on Machine Learning*.

Gautier Izacard and E. Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *ArXiv*, abs/2007.01282.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer.

2020a. 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.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, V. Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Thirty-third Conference on Neural Information Processing Systems*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

R. Puri, Ryan Spring, M. Patwary, M. Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *ArXiv*, abs/2002.09599.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

E. Voorhees. 1999. The trec-8 question answering track report. In *TREC*.

Zhiguo Wang, Yue Zhang, Mo Yu, Wei Zhang, Lin Pan, Linfeng Song, Kun Xu, and Yousef El-Kurdi. 2019. Multi-granular text encoding for self-explaining categorization. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 41–45, Florence, Italy. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

A Implementation Details

Evidence Corpus. We keep the version of English Wikipedia Dump consistent to the annotation timestep of NQ-OPEN and AMBIGQA, which is 2018-12-20 and 2020-01-20 respectively. Models pre-trained on NQ-OPEN use passages from dump 2018-12-20 while models fine-tuned on AMBIGQA take dump 2020-01-20. We use the AMBIGQA processed passages of these dumps, which takes the plain text and split Wikipedia pages into 100-word passages. As a result, there are 22M passages of Wikipedia Dump 2018-12-20 and 24M passages of Wikipedia Dump 2020-01-20.

Retrieval & Reranking. We use the multiset version of Dense Passage Retriever (DPR) (Karpukhin et al., 2020), which is jointly trained on five open-domain QA datasets. For the reranker, we fine-tune a `bert-large-cased` model with a batch size 16, learning rate 1e-5, training epoch 10 on the NQ-OPEN dataset. We sample 1 positive and 31 negative passages in training to maximize log-likelihood of the positive passage. The best reranker model is selected according to the answer recall in top 100 reranked passages. The trained reranker model is used for both NQ-OPEN and AMBIGQA dataset (we tried to finetune this model on AMBIGQA but did not receive any sensible improvement). The total training takes 10 hours and we tune the learning rate from 1e-5 to 5e-5 and select the best one.

Answer Prediction. We train a $\text{BART}_{\text{large}}$ model on NQ-OPEN with a batch size 64, epoch 10, and learning rate 5e-5. Then we finetune the trained model on AMBIGQA with a batch size 64, epoch 30, and learning rate 3e-5. According to empirical results, we discard training samples which the gold answers do not appear in any input passages for training on both NQ-OPEN and AMBIGQA (in the case of AMBIGQA, we discard training examples only when none of gold answers are found). All models are selected according to the performance (EM for NQ-OPEN, F1_{ans} (all) for AMBIGQA) on the development set.

Question Disambiguation. We train a $\text{BART}_{\text{large}}$ model on NQ-OPEN with a batch size 64, epoch 10, and learning rate 1e-5. Then we finetune the trained model on AMBIGQA with a batch size 64, epoch 30, and learning rate 5e-5. Different from training in answer prediction, we do not filter training samples which the answer does not appear in any input passages according to empirical results. The best model is selected according to $\text{F1}_{\text{EDIT-F1}}$ for both NQ-OPEN and AMBIGQA on the development set.

LM Verification. Based on the best QA model on NQ-OPEN trained in the Answer Prediction, we finetune it using the gold *disambiguated* QA pairs from AMBIGQA, in which each disambiguated question is only paired with one answer. We use a batch size 64, epoch 30, and learning rate 3e-5 for finetuning, and select the best model according to the EM score on the dev. set of AMBIGQA.

All the experiments are conducted on a single machine with 8 V100 GPUs. The pre-training on NQ-OPEN takes 60 hours for models in Answer Prediction, Question Disambiguation and LM Verification, and the fine-tuning takes 10 hours on AMBIGQA.

B Error Analysis

Answer Prediction Error. In the development set of AMBIGQA, 22.9% of examples actually have multiple interpretations but REFUEL only predicts one answer. In 12.0% examples, REFUEL wrongly predicts multiple answers on the unambiguous prompt questions. In the rest 65.1% examples, REFUEL aligns with annotators in terms of the ambiguity. Since REFUEL tends to wrongly think the prompt question is unambiguous, it predicts fewer answers than ground truth (1.55 *vs.* 2.02 on average). In effect, the predicted answers have a relatively high precision 55.6% but low recall 48.0%. By localizing where the errors come from, we find that in 2.3% of examples, REFUEL fails to retrieve any relevant passage which contains gold answers. In 27.0% of examples, retrieved passages only contain part of gold answers. In 38.6% of examples, retrieved passages can cover all gold answers but REFUEL fails to make correct predictions.

Question Disambiguation Error. We analyze the quality of disambiguated questions when the predicted answers are correct. We select 100 samples from the development data and summarize errors into five categories in Figure 5. We see that 42% of generated questions are totally wrong and 15% of them are identical to the prompt ones. Besides, there are in total 31% of generated questions (*Correct but Different Constraints*, *Correct but Paraphrase*) are actually correct but do not get credits under the current matching based evalua-

Error Type	%	Example
Wrong Disambiguation	42	Prompt Q: How long do contestants get to answer on jeopardy? Answer: 30 seconds Disamb. Q (g): How long do contestants have to answer during the last round of Jeopardy!? Disamb. Q (p): How long do contestants get to answer on the electronic display on jeopardy?
Correct but Different Constraints	19	Prompt Q: Who is the administrator of the small business administration? Answer: mickey thomas Disamb. Q (g): Who is the administrator of the small business administration from 2014 to 2017? Disamb. Q (p): Who is the 24th administrator of the small business administration?
Correct but Paraphrase	13	Prompt Q: Who are the kane county cougars affiliated with? Answer: arizona diamondbacks Disamb. Q (g): Who have the Kane County Cougars been affiliated with since 2015? Disamb. Q (p): Who are the Kane County Cougars affiliated with from 2015-present?
Annotation Error	11	Prompt Q: Who played tony in only fools and horses? Answer: christopher papazoglou Disamb. Q (g): Who played tony driscoll in only fools and horses? Disamb. Q (p): Who played Tony in Only Fools and Horses from 1981-1983?
No Disambiguation	15	Prompt Q: Who has the most nascar wins in history? Answer: richard petty Disamb. Q (g): Who has the most nascar super series wins in all-time history? Disamb. Q (p): Who has the most NASCAR wins in history?

Figure 5: Types of question disambiguation errors and their proportions in the dev. data based on 100 samples. "Disamb. Q (g)/(p)": Gold/Predicted Disambiguated Question. "*Correct but Different Constraints*": Predicted questions are correct interpretations of the answers but expressed through different constraints. "*Correct but Paraphrase*": Predicted questions are paraphrases of gold questions. The difference between disambiguated questions and prompt questions is highlighted.

tion metric $\text{F1}_{\text{EDIT-F1}}$. This suggests that a better evaluation metric should be incorporated in future to mitigate the variability of language generation, such as using a trained QA model for evaluation.

C Details of Human Evaluation

Instruction Details. Figure 6 shows the instruction and interface for human evaluation. We have three choices for each QA pair: "Answer is correct", "Answer is incorrect" and "Insufficient evidence". Since each QA pair has 100 retrieved passages, we show 5 retrieved passages (with answer highlighted) at a time. If the worker select "Insufficient evidence", we will show the next 5 retrieved passages until this QA pair receives a "correct/incorrect" decision. If "Insufficient evidence" is still select after showing all 100 passages, then we mark this QA pair as "incorrect".

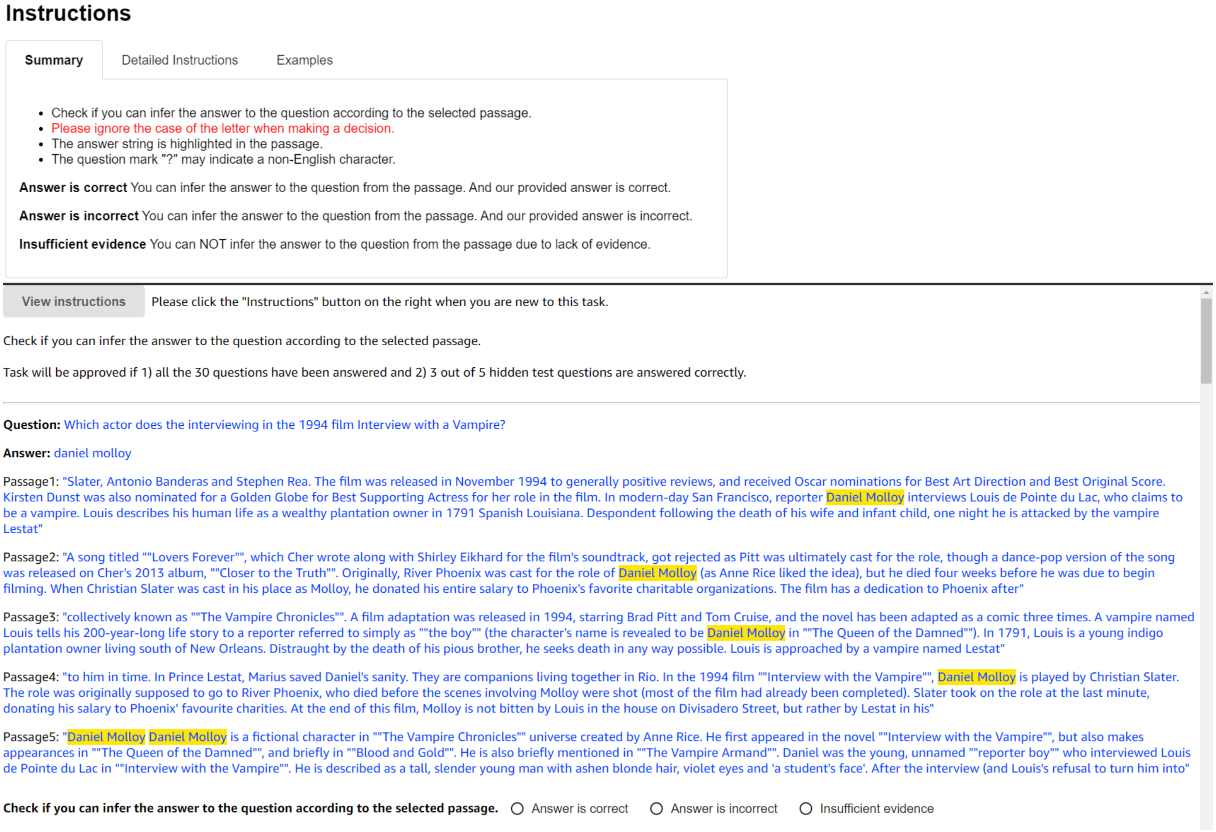
Evaluation Metrics & Quality Control. Let $(q_1, a_1), ..., (q_n, a_n)$ be n generated QA pairs from the same prompt question, we define two levels of correctness as follows: **#C-QAs**: (q_i, a_i) is considered **C**orrect if a_i is a correct answer of q_i; **#CD-QAs**: (q_i, a_i) is considered correct iff. (1) a_i is a correct answer of q_i *and* (2) any $a_j (j \neq i)$ is a wrong answer of q_i. #CD-QAs is designed to examine the **C**orrectness of question **D**isambiguation because ambiguous questions can have multiple valid answers. Moreover, it reduce the priming effect so that workers won't have a tendency to mark all samples as correct. During annotation, workers do not know each question q_i is paired with its answer a_i or other answers $a_j (j \neq i)$ under the same prompt question.

We only recruit workers based in the United States and pay 0.2 USD per QA pair on Mturk. For quality control, we have manually annotate 15 correct QA pairs and 15 wrong QA pairs (pair q_i with $a_j (j \neq i)$, and randomly select 5 of them to examine the quality of annotation. The task will be approved only when 3 out of 5 hidden test QA pairs receive correct annotations.

D Discussion on Problem Formulation

REFUEL follows the problem formulation of SPANSEQGEN to firstly predict one or multiple answers, and then generate the disambiguated question for each answer. We also tried/considered different formulations of this problem as follows:

QGen-AGen. We swap the order of answer prediction and question disambiguation in the problem formulation – firstly a QD model generates several disambiguated questions in a sequence, or predicts EOS if the question is not ambiguous; Then a QA model predicts a single answer for each predicted disambiguated question. This approach does not work in our experiments with poor performance. We think the major reason is generating multiple disambiguated question from the prompt question as the first step is much harder than the original



Instructions

- Check if you can infer the answer to the question according to the selected passage.
- Please ignore the case of the letter when making a decision.
- The answer string is highlighted in the passage.
- The question mark "?" may indicate a non-English character.

Answer is correct You can infer the answer to the question from the passage. And our provided answer is correct.

Answer is incorrect You can infer the answer to the question from the passage. And our provided answer is incorrect.

Insufficient evidence You can NOT infer the answer to the question from the passage due to lack of evidence.

Please click the "Instructions" button on the right when you are new to this task.

Check if you can infer the answer to the question according to the selected passage.

Task will be approved if 1) all the 30 questions have been answered and 2) 3 out of 5 hidden test questions are answered correctly.

Question: Which actor does the interviewing in the 1994 film Interview with a Vampire?

Answer: daniel molloy

Passage1: "Slater, Antonio Banderas and Stephen Rea. The film was released in November 1994 to generally positive reviews, and received Oscar nominations for Best Art Direction and Best Original Score. Kirsten Dunst was also nominated for a Golden Globe for Best Supporting Actress for her role in the film. In modern-day San Francisco, reporter Daniel Molloy interviews Louis de Pointe du Lac, who claims to be a vampire. Louis describes his human life as a wealthy plantation owner in 1791 Spanish Louisiana. Despondent following the death of his wife and infant child, one night he is attacked by the vampire Lestat"

Passage2: "A song titled ""Lovers Forever"", which Cher wrote along with Shirley Eikhard for the film's soundtrack, got rejected as Pitt was ultimately cast for the role, though a dance-pop version of the song was released on Cher's 2013 album, ""Closer to the Truth"". Originally, River Phoenix was cast for the role of Daniel Molloy (as Anne Rice liked the idea), but he died four weeks before he was due to begin filming. When Christian Slater was cast in his place as Molloy, he donated his entire salary to Phoenix's favorite charitable organizations. The film has a dedication to Phoenix after"

Passage3: "collectively known as ""The Vampire Chronicles"". A film adaptation was released in 1994, starring Brad Pitt and Tom Cruise, and the novel has been adapted as a comic three times. A vampire named Louis tells his 200-year-long life story to a reporter referred to simply as ""the boy"" (the character's name is revealed to be Daniel Molloy in ""The Queen of the Damned""). In 1791, Louis is a young indigo plantation owner living south of New Orleans. Distraught by the death of his pious brother, he seeks death in any way possible. Louis is approached by a vampire named Lestat"

Passage4: "to him in time. In Prince Lestat, Marius saved Daniel's sanity. They are companions living together in Rio. In the 1994 film ""Interview with the Vampire"", Daniel Molloy is played by Christian Slater. The role was originally supposed to go to River Phoenix, who died before the scenes involving Molloy were shot (most of the film had already been completed). Slater took on the role at the last minute, donating his salary to Phoenix' favourite charities. At the end of this film, Molloy is not bitten by Louis in the house on Divisadero Street, but rather by Lestat in his"

Passage5: "Daniel Molloy Daniel Molloy is a fictional character in ""The Vampire Chronicles"" universe created by Anne Rice. He first appeared in the novel ""Interview with the Vampire"", but also makes appearances in ""The Queen of the Damned"", and briefly in ""Blood and Gold"". He is also briefly mentioned in ""The Vampire Armand"". Daniel was the young, unnamed ""reporter boy"" who interviewed Louis de Pointe du Lac in ""Interview with the Vampire"". He is described as a tall, slender young man with ashen blonde hair, violet eyes and 'a student's face'. After the interview (and Louis's refusal to turn him into"

Check if you can infer the answer to the question according to the selected passage. ○ Answer is correct ○ Answer is incorrect ○ Insufficient evidence

Figure 6: Instructions and interface for human evaluation. *(best viewed in color)*

formulation which only requires to generating multiple plausible answers from the prompt question.

QAGen. Another possible approach is using a single model to predict disambiguated question-answer pairs where each answer right precedes its disambiguated question. This is certainly a possible way but it is even more challenging than **QGen-AGen**. We did not try this way after receiving poor performance from **QGen-AGen**.

E Baselines for Round-Trip Prediction

Since the current round-trip prediction requires several iteration between the answer prediction module and the question disambiguation module, it would be better to over-generate many answers in one pass. One straightforward way to generate more QA pairs is setting a minimum length of generation for the answer prediction model, and then go through the LM Verification process to drop the low-quality predictions. We set two minimum lengths of generation (L=8/16) for our answer prediction model. As shown in Table 7, although setting a minimum length effectively increases the number of predicted QA pairs (2.10/2.88 for L=8/16), the over-generated answers are extremely noisy which in turn hurts the effectiveness of the LM Verification model, resulting in far worse performance across all metrics. Presumably, one major disadvantage of the Min-Length Generation approach is that REFUEL loses the flexibility to decide the number of possible interpretations based on the passages. Instead, it always generates multiple answers according to the minimum length.

F More Cases from REFUEL: Figure 7

Models	#QAs	F1$_{\text{ans}}$ (all)	F1$_{\text{ans}}$ (multi)	F1$_{\text{BLEU}}$	F1$_{\text{EDIT-F1}}$	Comb.
REFUEL w/o RTP	1.55	**48.4**	37.0	16.0	11.2	59.6
+ Round-Trip Generation	2.06	47.6	**37.4**	16.0	11.4	59.0
+ Round-Trip Generation & LM Verification	1.72	48.3	37.3	**16.2**	**11.8**	**60.1**
+ Min-Length Generation (L=8)	2.10	40.8	36.2	15.5	11.1	51.9
+ Min-Length Generation (L=8) & LM Verification	1.69	42.9	36.3	15.9	11.4	54.4
+ Min-Length Generation (L=16)	2.88	37.2	34.1	14.6	10.3	47.5
+ Min-Length Generation (L=16) & LM Verification	1.46	43.1	34.5	15.2	11.1	54.2

Table 7: Dev. set results on different approaches to harvest more interpretations (QA pairs) towards the ambiguous questions. "#QAs" denotes the average number of generated QA pairs per prompt question.

Prompt question #1: Who played lead guitar for the rolling stones?
Reference:
Q1: Who played lead guitar for the rolling stones from 1962-1969? / A1: brian jones
Q2: Who played lead guitar for the rolling stones from 1969-1974? / A2: mick taylor
Q3: Who played lead guitar for the rolling stones from since 1962? / A3: keith richards
Q4: Who played lead guitar for the rolling stones from since 1975? / A4: ronnie wood
Prediction of REFUEL w/o Round-Trip Prediction: (QA1-QA3: F1$_{\text{ans}}$=57.1, F1$_{\text{EDIT-F1}}$=8.2)
Q1: Who played electric guitar for the Rolling Stones from 1962-present? / A1: keith richards
Q2: Who primarily played guitar for the Rolling Stones? / A2: mick jagger and keith richards
Q3: Who originally played slide guitar for the Rolling Stones? / A3: brian jones
Prediction of REFUEL: (QA1-QA4: F1$_{\text{ans}}$=75.0, F1$_{\text{EDIT-F1}}$=15.5)
Q4: Who played bass guitar for the Rolling Stones from 1969-1975? / A4: mick taylor
Relevant Snippets of Passages: (w/ rank from retrieval & reranking module)
Rank 2: ... the original lineup consisted of multi-instrumentalist brian jones, lead vocalist mick jagger, guitarist keith richards, bass guitarist bill wyman, drummer charlie watts, and keyboardist ian stewart. ... following jones' death in 1969, mick taylor took over lead guitar duties until 1974.
Rank 4: mick jagger sir michael philip jagger (born 26 july 1943) is an english singer, ... his distinctive voice and energetic live performances, along with keith richards' guitar style, have been the trademark of the rolling stones ...
Rank 10: song, as the stones are generally known for their guitar interplay of rhythm and lead ("weaving") between richards and the other guitarist in the band – brian jones (1962–1969), mick taylor (1969–1975), and ronnie wood (1975–present) ...

Prompt question #2: When does the ration shop open in india?
Reference:
Q1: When did the ration shop open in india for the first time? / A1: February 1944
Q2: When did the ration shop in its current form open in india? / A2: June 1947
Prediction of REFUEL w/o Round-Trip Prediction: (QA1: F1$_{\text{ans}}$=66.7, F1$_{\text{EDIT-F1}}$=28.6)
Q1: When does the Indian Food Security System open in its current form? / A1: june 1947
Prediction of REFUEL: (QA1-QA2: F1$_{\text{ans}}$=100.0, F1$_{\text{EDIT-F1}}$=71.4)
Q2: When does the first ration shop open in india? / A2: february 1944
Relevant Snippets of Passages: (w/ rank from retrieval & reranking module)
Rank 3: public distribution system the indian food security system was established by the government ... this scheme was first started in february 1944, during the second world war, and was launched in the current form in june 1947.

Prompt question #3: When is the new christopher robin coming out?
Reference:
Q1: When did the new Christopher Robin come out in Burbank? A1: July 30, 2018
Q2: When did the new Christopher Robin come out throughout the United States? A2: August 3, 2018
Prediction of REFUEL w/o Round-Trip Prediction: (QA1-QA2: F1$_{\text{ans}}$=50.0, F1$_{\text{EDIT-F1}}$=28.6)
Q1: When did the new christopher robin film come out in the US? A1: august 3, 2018
Q2: When did the new christopher robin film come out at the Disneyland Resort? A2: july 17, 2018
Prediction of REFUEL: (QA1-QA3: F1$_{\text{ans}}$=80.0, F1$_{\text{EDIT-F1}}$=53.6)
Q3: When did the new christopher robin film come out in California? A3: july 30 2018
Relevant Snippets of Passages: (w/ rank from retrieval & reranking module)
Rank 1: "christopher robin" had its premiere in burbank, california on july 30, 2018. released in the united states on august 3, 2018, by walt disney studios motion pictures, the film grossed over $197 million.
Rank 2: "christopher robin" premiered in burbank, california on july 30, 2018, and was released on august 3, 2018 by walt disney studios motion pictures.
Rank 18: for the first time as a disney movie club exclusive on july 17, 2018 to coincide with its belated 20th anniversary and the live-action "christopher robin" film, released over two weeks later.

Figure 7: Predictions generated by REFUEL from the development data. We also manually check all the 100 retrieved and reranked passages, and list the answer-relevant passages here. However, the listed passages might be different from the passages that annotators search and read during annotation.