AMBIGQA: Answering Ambiguous Open-domain Questions

Sewon Min[1,2] **Julian Michael**[1] **Hannaneh Hajishirzi**[1,3] **Luke Zettlemoyer**[1,2]
[1]University of Washington [2]Facebook AI Research [3]Allen Institute for Artificial Intelligence
{sewon,julianjm,hannaneh,lsz}@cs.washington.edu

Abstract

Ambiguity is inherent to open-domain question answering; especially when exploring new topics, it can be difficult to ask questions that have a single, unambiguous answer. In this paper, we introduce AMBIGQA, a new open-domain question answering task which involves finding every plausible answer, and then rewriting the question for each one to resolve the ambiguity. To study this task, we construct AMBIGNQ, a dataset covering 14,042 questions from NQ-OPEN, an existing open-domain QA benchmark. We find that over half of the questions in NQ-OPEN are ambiguous, with diverse sources of ambiguity such as event and entity references. We also present strong baseline models for AMBIGQA which we show benefit from weakly supervised learning that incorporates NQ-OPEN, strongly suggesting our new task and data will support significant future research effort. Our data and baselines are available at https://nlp.cs.washington.edu/ambigqa.

1 Introduction

In the open-domain setting, it can be difficult to formulate clear and unambiguous questions. For example, Figure 1 shows a Google search query (Kwiatkowski et al., 2019) that, perhaps surprisingly, has two possible interpretations given the evidence in Wikipedia. Although open-domain question answering (QA) systems aim to answer any factoid question (Voorhees et al., 1999), existing methods assume questions have a single well-defined answer. Nonetheless, ambiguity arises frequently in open-domain QA, where questions are written during information gathering (e.g., search queries) without knowledge of the answer. As we will see in Section 4, over 50% of the questions we sampled from a set of Google search queries are ambiguous. Furthermore, identifying ambiguities is difficult both for humans and machines. As shown in Figure 1, ambiguity is a function of both the question and the evidence provided by a large text corpus.

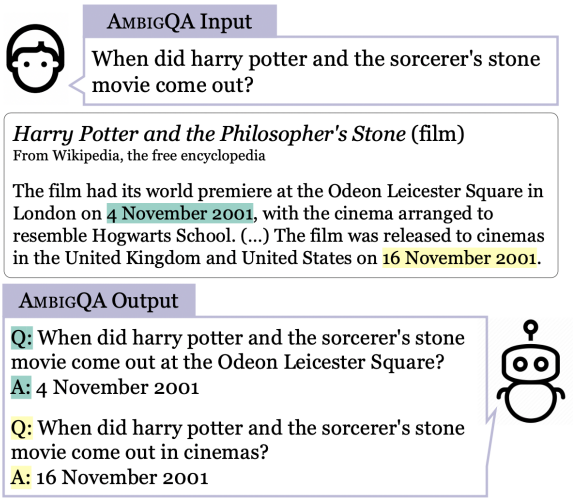


Figure 1: An AMBIGNQ example where the prompt question (top) appears to have a single clear answer, but is actually ambiguous upon reading Wikipedia. AMBIGQA requires producing the full set of acceptable answers while differentiating them from each other using disambiguated rewrites of the question.

To study this challenge, we introduce AMBIGQA (Answering Ambiguous Open-domain Questions), a new task which involves disambiguating and answering potentially ambiguous questions. Specifically, the model must (1) find a set of distinct, equally plausible answers to the question, and (2) provide minimal yet unambiguous rewrites of the question that clarify the interpretation which leads to each answer. Figure 1 shows two such disambiguated questions and their answers.

To support the study of this task, we construct a dataset called AMBIGNQ using 14,042 questions from an open-domain version of NATURAL QUESTIONS (Kwiatkowski et al., 2019), denoted NQ-OPEN. For each question, annotators search for,

Type	Example
Event references (39%)	What season does meredith and derek get married in grey's anatomy? Q: In what season do Meredith and Derek get informally married in Grey's Anatomy? / A: Season 5 Q: In what season do Meredith and Derek get legally married in Grey's Anatomy? / A: Season 7
Properties (27%)	How many episode in seven deadly sins season 2? Q: How many episodes were there in seven deadly sins season 2, not including the OVA episode? / A: 25 Q: How many episodes were there in seven deadly sins season 2, including the OVA episode? / A: 26
Entity references (23%)	How many sacks does clay matthews have in his career? Q: How many sacks does Clay Matthews Jr. have in his career? / A: 69.5 Q: How many sacks does Clay Matthews III have in his career? / A: 91.5
Answer types (16%)	Who sings the song what a beautiful name it is? Q: Which group sings the song what a beautiful name it is? / A: Hillsong Live Q: Who is the lead singer of the song what a beautiful name it is? / A: Brooke Ligertwood
Time-dependency (13%)	When does the new family guy season come out? Q: When does family guy season 16 come out? / A: October 1, 2017 Q: When does family guy season 15 come out? / A: September 25, 2016 Q: When does family guy season 14 come out? / A: September 27, 2015
Multiple sub-questions (3%)	Who was british pm and viceroy during quit india movement? Q: Who was british viceroy during quit India movement? / A: Victor Hope Q: Who was british pm during quit India movement? / A: Winston Churchill

Table 1: Breakdown of the types of ambiguity in 100 randomly sampled items from the AMBIGNQ development data. Each example may fall into multiple categories.

navigate, and read multiple Wikipedia pages to find as many answers as possible. The high prevalence of ambiguity makes the task difficult even for human experts; it is inherently difficult to know if you have found every possible interpretation of a question. Nonetheless, we are able to collect high quality data covering high levels of ambiguity (2.1 distinct answers per question on average) with high estimated agreement (89.0 F1) on valid answers. The types of ambiguity are diverse and sometimes subtle (Table 1), including ambiguous entity or event references, or ambiguity over the answer type; many are only apparent after examining one or more Wikipedia pages.

To establish initial performance levels on this data, we present a set of strong baseline methods. We extend a state-of-the-art QA model (Karpukhin et al., 2020) with three new components: (1) set-based question answering with a sequence-to-sequence model, (2) a question disambiguation model, and (3) a modification to democratic co-training (Zhou and Goldman, 2004) which leverages the partial supervision available in the full NQ-OPEN dataset. We also do an ablation study and qualitative analysis, which suggest there is significant room for future work on this task.

To summarize, our contributions are threefold.

1. We introduce AMBIGQA, a new task which requires identifying all plausible answers to an open-domain question, along with disambiguated questions to differentiate them.
2. We construct AMBIGNQ, a dataset with 14,042 annotations on NQ-OPEN questions containing diverse types of ambiguity.
3. We introduce the first baseline models that produce multiple answers to open-domain questions, with experiments showing their effectiveness in learning from our data while highlighting avenues for future work.

2 Related Work

Open-domain Question Answering requires a system to answer any factoid question based on evidence provided by a large corpus such as Wikipedia (Voorhees et al., 1999; Chen et al., 2017). Existing benchmarks use questions of various types, from open-ended information-seeking (Berant et al., 2013; Kwiatkowski et al., 2019; Clark et al., 2019) to more specialized trivia/quiz (Joshi et al., 2017; Dunn et al., 2017). To the best of our knowledge, all existing formulations assume each question has a single clear answer.

Our work is built upon an open-domain version of NATURAL QUESTIONS (Kwiatkowski et al., 2019), denoted NQ-OPEN, composed of questions posed by real users of Google search, each with an answer drawn from Wikipedia. NQ-OPEN has promoted several recent advances in open-

domain question answering (Lee et al., 2019; Asai et al., 2020; Min et al., 2019a,b; Guu et al., 2020; Karpukhin et al., 2020). Nonetheless, Kwiatkowski et al. (2019) report that the answers to such questions are often debatable, and the average agreement rate on NQ-OPEN test data is 49.2%,[1] in large part due to ambiguous questions. In this work, we embrace this ambiguity as inherent to information seeking open-domain QA, and present the first methods for returning sets of answers paired with different interpretations of the question.

Clarification Questions have been used to study question ambiguity in other settings. Research on community Q&A (Braslavski et al., 2017; Rao and Daumé III, 2018, 2019) studies finding underspecification in the question, but it does not find the answer to the original question. In recent work, Xu et al. (2019) study clarification of questions that are intentionally annotated with pre-specified entity reference ambiguities. Aliannejadi et al. (2019) and Zamani et al. (2020) use clarification questions to refine intents of simple query logs without immediately apparent information needs (e.g., single keywords like *dinosaur*[2]).

In contrast, we study open-domain factoid questions asked by real users: these present clear information needs, but carry diverse naturally occurring ambiguities (see Table 1). Furthermore, instead of prolonging the user's information-seeking session with clarification questions, our task formulation provides a complete and immediate solution with unambiguous rewrites of the original question.

Question Rewriting is a novel, well-defined task which we propose for differentiating distinct answers. To the best of our knowledge, it has not been studied for resolving ambiguity; we are only aware of Elgohary et al. (2019) which use question rewriting to convert conversational questions into self-contained questions.

3 Task: AMBIGQA

3.1 AMBIGQA Setup

Figure 1 depicts the AMBIGQA task. The input is a prompt question q, and the output is a list of n question-answer pairs $(x_1, y_1), \ldots, (x_n, y_n)$, where each y_i is an equally plausible answer to q, and each x_i is a minimally edited modification of q whose answer is unambiguously y_i. We consider two subtasks.

Multiple Answer Prediction. Given a question q, output a set of semantically distinct and equally plausible answers $y_1, \ldots, y_n$, where n is unknown.

Question Disambiguation. Given q and a set of answers $y_1, \ldots, y_n$, generate *disambiguated questions* $x_1, \ldots, x_n$, where each x_i is a *minimal edit* of q which makes it unambiguous so that y_i is a correct answer and all y_j for all $j \neq i$ are incorrect. When $n = 1$, this task is trivial, as $x_1 = q$.

We choose to represent ambiguity with a set of disambiguated questions because it is well-defined, immediately human-interpretable, and allows for straightforward annotation of a wide range of ambiguities without complex guidelines.

3.2 Evaluation Metrics

To evaluate model performance, we present several ways to compare a model prediction with m question-answer pairs $(x_1, y_1), \ldots, (x_m, y_m)$ with a gold reference set with n pairs $(\bar{x}_1, \bar{\mathcal{Y}}_1), \ldots, (\bar{x}_n, \bar{\mathcal{Y}}_n)$. Since there may be more than one way to refer to a single answer (e.g., *Michael Jordan* and *Michael Jeffrey Jordan*) each gold answer $\bar{\mathcal{Y}}_i$ is a set of acceptable answer strings, where all $\bar{\mathcal{Y}}_i$ are disjoint.

We assign each predicted question-answer pair (x_i, y_i) a *correctness score* based on a string similarity function f valued in $[0, 1]$.

$$c_i = \max_{1 \leq j \leq n} \mathbb{I}[y_i \in \bar{\mathcal{Y}}_j] f(x_i, \bar{x}_j).$$

Intuitively, c_i considers (1) the correctness of the answer and (2) the similarity $f(x_i, \bar{x}_j)$ between the predicted and reference question. We calculate F1 treating the c_i as measures of correctness:

$$\text{prec}_f = \frac{\sum_i c_i}{m}, \qquad \text{rec}_f = \frac{\sum_i c_i}{n},$$
$$\text{F1}_f = \frac{2 \times \text{prec}_f \times \text{rec}_f}{\text{prec}_f + \text{rec}_f}.$$

We consider three choices of F_f. $\text{F1}_{\textbf{ans}}$ is the F1 score on answers only, where f always yields 1. This may be used without the question disambiguation step. $\text{F1}_{\textbf{BLEU}}$ accounts for string similarity between questions, calculating f with BLEU (Papineni et al., 2002). $\text{F1}_{\textbf{EDIT-F1}}$ uses EDIT-F1 as f, where EDIT-F1 is a new measure that represents each disambiguated question by its added and

[1] The NQ-OPEN test data has 5-way annotations; we compute their pairwise agreement based on string match.

[2] The average query length in Zamani et al. (2020) is 2.6.

deleted unigrams compared to the prompt question, and computes the F1 score between them. For example, consider the prompt question "Who made the play the crucible?", the reference "Who wrote the play the crucible?" and the prediction "Who made the play the crucible in 2012?". The gold edits[3] here are ⦃ -made , +wrote ⦄ while the predicted edits are ⦃ +in , +2012 ⦄. Their EDIT-F1 is thus zero, even though the questions are similar. Unlike BLEU which we use to directly measure similarity to the gold question, this metric only gives credit for getting the key semantic differences correct between the original question and the clarification.

4 Data: AMBIGNQ

4.1 Data Collection

We construct AMBIGNQ using prompt questions from NQ-OPEN and English Wikipedia as the evidence corpus. We use Amazon Mechanical Turk for crowdsourcing.

The crucial annotation challenge is maximizing recall: finding all possible distinct answers to a question. This is difficult, as ambiguities are often only apparent after carefully searching the evidence for multiple possible answers. However, we can collect high quality data with high levels of ambiguity using careful worker selection and a two stage pipeline: *generation* and *validation*.

Generation. Workers in the first stage are given a prompt question and a search box that uses the Google Search API restricted to English Wikipedia. Allowing annotators to find Wikipedia pages on their own closely approximates the real process people use to answer open-ended questions—an approach with no existing large-scale dataset.[4]

Workers find all plausible answers to the question; when there are multiple, each answer is paired with a minimal edit of the prompt question which differentiates it from the other answers, in line with our task requirements. A distinct answer may be annotated as multiple possible spans (e.g., *Michael Jordan* and *Michael Jeffrey Jordan*).

As a special case, some questions contain *temporal deixis* which depends on the time of writing, e.g., "When does the new family guy season come out?". To avoid unmanageably many answers, we instruct workers to remove the time-dependence by rewriting the prompt question for up to three most recent events before Jan 1, 2018, e.g., "When does family guy season 16 come out?" (see Table 1).

[3]Represented as multisets, written using ⦃bag⦄ notation.

[4]For instance, answers in NQ-OPEN are annotated over pre-specified Wikipedia pages from the Google search engine.

Split	# data	# QAs %			
		1	2	3	4+
Train	10,036	53	24	14	10
Dev	2,002	49	23	14	13
Test	2,004	44	24	16	16

Table 2: Data statistics. For the number of QA pairs (# QAs), the minimum is taken when there are more than 1 accepted annotations.

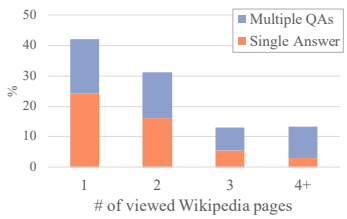
Validation. Workers in the validation stage review the annotations provided by multiple generators. Validators mark each generator's annotations as correct or incorrect, or provide a new set of question-answer pairs by combining the valid ones from each generator. They search Wikipedia as generators do, and are additionally given Wikipedia pages that generators viewed to speed up the process. Validation is skipped when annotated answers from all generators exactly match (37% of cases).

Quality control. We recruit highly qualified workers through a qualification test (details in Appendix A). Although the task was difficult for most workers, we found that our highly qualified full-time workers, given quick and detailed feedback on their work, produced high accuracy and recall. For development and test data, we use two generators and one validator per prompt question. For training data, we skip validation and only use one generator per question.

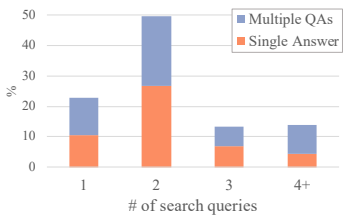
Inter-annotator agreement. Evaluating generators against each other on the development set yields 60.8 $\mathrm{F1}_{\mathrm{ans}}$. All annotations passed validation for 76% of questions, while validators made changes (edits or exclusions) in the remaining 24%. The average $\mathrm{F1}_{\mathrm{ans}}$ between co-authors and workers on a sample of 50 validations was 89.0%. This indicates that, despite the intrinsic difficulty and subjectivity of the task, humans agree on the boundary between valid and invalid answers in most cases.

4.2 Data Analysis

The final dataset contains 14,042 annotated examples, split consistently with NQ-OPEN. As shown in Table 2, over 50% of development and test examples contain multiple question-answer pairs. This



(a) Number of unique Wikipedia pages visited by crowdworkers.†



(b) Number of search queries written by crowdworkers.

(c) Word cloud of the edits made in questions; ■ and ■ indicate added and deleted unigrams, respectively.

Figure 2: Data Analysis on the development data. †This is actually an underestimate; we could not track when annotators viewed pages by following hyperlinks for technical reasons.

indicates a high rate of ambiguity in NQ-OPEN, even though previous work has studied it with the assumption that each question has a single answer. We also find a discrepancy between development and test; this is likely due to the way in which NQ-OPEN is constructed, which over-samples difficult questions in the test set (see Appendix B for details). The training set contains relatively fewer ambiguous examples (47%), presumably because using only one worker per training example yielded slightly lower recall.

Types of ambiguity. Table 1 shows a breakdown of the types of ambiguity in AMBIGNQ. They are diverse, including ambiguity in entity references, event references, properties, and answer types, with a relatively uniform distribution between them. In comparison to Xu et al. (2019), who intentionally elicit questions with ambiguous entity references, our analysis shows that *unintended* ambiguity comes from diverse sources. In many cases, ambiguity is not apparent from the prompt question alone, but only after researching the question on Wikipedia, as evidenced by differences in model performance (Section 6.2).

Annotator behavior. Figures 2a and 2b show the number of unique Wikipedia pages and the number of search queries used by workers during annotation. More often than not, workers used multiple queries and navigated multiple Wikipedia pages, showing how our setup captures ambiguity in the *retrieval* step of open-domain question answering, which is missed in approaches that assume a pre-specified evidence document.

Distribution of edits. Figure 2c shows unigram edits made to questions in the development data, where we remove stopwords except wh-words and group numeric values by the number of digits. Adding numerals such as years is common, as they can easily disambiguate entity or event references or remove time dependence. Wh-word changes are also common, especially for specifying the answer type (e.g., from *who* to *which group*; see Table 1). The distribution of edits is fairly long-tailed, with the 100 most frequent edits covering 36% of the total, and the top 1,000 covering 69%.

5 Model

To set initial performance levels on AMBIGNQ, we present a baseline AMBIGQA model combining ideas from recent advances in open-domain QA (Karpukhin et al., 2020) and generation (Lewis et al., 2020). Given a prompt question q, our model predicts answers $y_1..y_n$, and generates corresponding questions $x_1..x_n$ conditioning on q, the answers $y_1..y_n$, and the evidence passages. A novel co-training step also allows the model to leverage the partial supervision available in NQ-OPEN.

Multiple Answer Prediction. Here we describe SPANSEQGEN, our model for multiple answer prediction. Following Karpukhin et al. (2020), a state-of-the-art model on NQ-OPEN, SPANSEQGEN first retrieves 100 passages with a BERT-based (Devlin et al., 2019) dual encoder, and reranks them using a BERT-based cross encoder. Then, instead of predicting an answer span from the top 1 passage as Karpukhin et al. (2020) does, SPANSEQGEN uses another sequence-to-sequence model based on BART (Lewis et al., 2020). Specifically, it conditions on the concatenation of q and the top passages in order up to 1024 tokens, and sequentially generates distinct answers token-by-token, separated by [SEP]. We pretrain SPANSEQGEN on NQ-OPEN and finetune it on AMBIGNQ.

We develop SPANSEQGEN primarily because Karpukhin et al. (2020) is designed for generating a single answer, but SPANSEQGEN also boosts the

Algorithm 1 Democratic co-training with weak supervision (Section 5).

// Each question in D_{full} has an answer list annotated
// Each question in $D_{partial}$ has one answer annotated
$\hat{D}_{\text{full}} \leftarrow D_{\text{full}}$
for $iter \in \{1..N\}$ **do**
  // Train C sequence-to-sequence QA models
  for $i \in \{1..C\}$ **do**
    $\phi_i \leftarrow train(\hat{D}_{\text{full}})$
  $\hat{D}_L \leftarrow D_{\text{full}}$
  for $(q^j, y^j) \in D_{\text{partial}}$ **do**
    // Get predictions by using y_j as prefix
    $\hat{Y}^j \leftarrow \{\hat{y} \mid \hat{y} \neq y^j$, and
      $|\{i \mid \hat{y} \in \phi_i(q^j|y^j), 1 \leq i \leq C\}| > \frac{C}{2}$
    }
    if $|\hat{Y}^j| > 0$ **then**
      // Add it as a multiple answer case
      $\hat{D}_{\text{full}} \leftarrow \hat{D}_L \cup \{(q^j, \{y^j\} \cup \hat{Y}^j)\}$
    else if $\forall i = 1..C, |\phi_i(x^j) - \{y^j\}| = 0$ **then**
      // Add it as a single answer case
      $\hat{D}_{\text{full}} \leftarrow \hat{D}_L \cup \{(q^j, \{y^j\})\}$

performance on NQ-OPEN (41.5→42.2 on the test data). We include ablations on different approaches and models in Section 6.2.

Question Disambiguation. We design a question disambiguation (QD) model based on BART. The model generates each question x_i ($i = 1..n$) conditioning on the concatenation of q, the target answer y_i, other answers $y_1..y_{i-1}, y_{i+1}..y_n$, and the top passages as used by SPANSEQGEN. We pretrain on NQ-OPEN to generate questions given an answer and passage, and then finetune it on the full task data in AMBIGNQ. We include ablations on different variants of the model in Section 6.2.

Co-training with weak supervision. Given the prevalence of unlabelled ambiguity in NQ-OPEN, we introduce a method that treats the NQ-OPEN annotations as weak supervision and learns to discover potential ambiguity in the data. We modify a democratic co-training algorithm (Zhou and Goldman, 2004) as described in Algorithm 1. We iteratively grow the training set $\hat{D}_{\text{full}}$ from AMBIGNQ (D_{full}) with silver data from NQ-OPEN (D_{partial}) predicted by a majority of a set C of SPANSEQGEN models trained on $\hat{D}_{\text{full}}$. The key step is injecting the known answer y^j from NQ-OPEN as a prefix to SPANSEQGEN's output during prediction. In each step, if a majority of C predict an additional answer, we assume we have found a false negative and add the result to the training set $\hat{D}_{\text{full}}$. If all models predict no additional answer, we add the example to $\hat{D}_{\text{full}}$ with y^j as a single answer.

6 Experiments

We describe the baseline models used in our experiments, followed by results and ablations. Implementation details and hyperparameters of all models are provided in Appendix D.

6.1 Baselines

DISAMBIG-FIRST. This baseline disambiguates the prompt question without any context from plausible answers or reference passages. Specifically, it implements the following pipeline: (1) Feed the prompt question q into a BERT-based binary classifier to determine whether it is ambiguous. (2) If q is ambiguous, pass it into a BART-based model which generates a sequence of disambiguated questions $x_1..x_n$ ($n > 1$), separated by `[SEP]`; otherwise, consider only $x_1 = q$. (3) Feed each x_i into a state-of-the-art model on NQ-OPEN (Karpukhin et al., 2020) to produce its answer y_i.

Thresholding + QD. We also include a model based on Karpukhin et al. (2020), with thresholding for multiple answer prediction and our question disambiguation (QD) model. Karpukhin et al. (2020) outputs a likelihood score for each span; we obtain $y_1..y_n$ by taking valid spans with likelihood larger than a hyperparameter γ. The model is trained to maximize the marginal likelihood of any span in the gold answer set $\bar{y}_1..\bar{y}_n$. As with SPANSEQGEN, we pretrain on NQ-OPEN and finetune on AMBIGNQ. We then produce disambiguated questions using our BART-based QD model (Section 5).

6.2 Results

Table 3 reports the performance of our baselines; example model outputs are provided in Table 5.

Main results. We first find that DISAMBIG-FIRST is significantly worse than other models. In particular, classification accuracy on whether the prompt question is ambiguous is 67%, close to the majority baseline (60%). When the model does identify an ambiguous question, its rewrites often look reasonable on the surface, but do not match the facts. For instance, in example 1 of Table 5, it asks about filming in 2017 and during season 1 for *Snow White and the Huntsman*, which was actually a film released in 2012. This shows that reading evidence documents is crucial for identifying and characterizing ambiguities.

While SPANSEQGEN outperforms Karpukhin et al. (2020) with thresholding, the difference is

Model	$F1_{ans}$ (*all*)		$F1_{ans}$ (*multi*)		$F1_{BLEU}$		$F1_{EDIT\text{-}F1}$	
	dev	test	dev	test	dev	test	dev	test
DISAMBIG-FIRST	28.1	24.8	21.9	18.8	4.2	4.0	2.7	2.2
Thresholding + QD	37.1	32.3	28.4	24.8	13.4	11.3	6.6	5.5
SPANSEQGEN + QD	39.7	33.5	29.3	24.5	13.4	11.4	7.2	5.8
SPANSEQGEN† + QD	41.2	35.2	29.8	24.5	13.6	10.6	7.4	5.7
SPANSEQGEN† (Co-training) + QD	**42.3**	**35.9**	**31.7**	**26.0**	**14.3**	**11.5**	**8.0**	**6.3**

Table 3: Results on AMBIGNQ. The *multi* measure only considers examples with multiple question-answer pairs. † indicates ensemble. See Appendix B for details on the discrepancy between development and test.

Model	q	y_i	$y_1..y_{i-1}$, $y_{i+1}..y_n$	*Full task*		*Gold answers given*	
				$F1_{BLEU}$	$F1_{EDIT\text{-}F1}$	$F1_{BLEU}$	$F1_{EDIT\text{-}F1}$
QD model	✓	✓	✓	14.3	**8.0**	40.1	**19.2**
- prompt question	-	✓	✓	6.7	7.7	15.1	**19.2**
- untargeted answers	✓	✓	-	14.2	7.3	41.2	17.2
Always prompt question	✓	-	-	**15.9**	0.0	**47.4**	0.0

Table 4: Ablations on question disambiguation (development data, multiple answers only). QD model refers to the question disambiguation model described in Section 5. For multiple answer prediction, we use SPANSEQGEN† with co-training (*Full task*) or the gold answers (*Gold answers given*).

not as great as we expected. This suggests two things. First, thresholding may be a surprisingly effective baseline for outputting multiple answers, even though the answers must compete with each other for probability mass in order to surpass the threshold γ. Second, maximizing likelihood in a sequence-to-sequence model like SPANSEQGEN may not produce well-calibrated results. For instance, the model seems to suffer due to variation in the length of the output sequence, outputting shorter sequences on average (3.0 tokens) than gold (6.7).[5] This leads to low recall when there are multiple answers; our best model achieves a precision of 49.6 and recall of 25.3 for its $F1_{ans}$ of 31.7 on such questions.

Overall, SPANSEQGEN achieves reasonable $F1_{ans}$ scores. $F1_{ans}$ on examples with multiple question-answer pairs (*multi*) are lower, indicating that predicting all plausible answers is more challenging than predicting a single answer, as expected. SPANSEQGEN also obtains the best performance in $F1_{BLEU}$ and $F1_{EDIT\text{-}F1}$, although their absolute values are low in general; we discuss this in our question disambiguation ablations below.

There is a substantial difference in performance between development and test overall, likely due to distributional differences in the original questions in NQ-OPEN; detailed discussion is in Appendix B.

Effect of co-training. The last two rows of Table 3 reports the effect of our co-training method. As co-training requires multiple trained models, we compare with a naive ensemble. While we see gains from ensembling alone, an ensemble trained with the co-training method achieves the best performance on all metrics. This result demonstrates the potential of jointly using AMBIGNQ and partial supervision from NQ-OPEN.

Ablations on question disambiguation. Table 4 reports results of an ablation experiment on question disambiguation (QD). Among our ablations, we include models without the prompt question or untargeted answers as input, and a naive baseline that always outputs the prompt question. We report the metrics both in the scenarios of the full task and the gold answers given, to see the performance dependent on and independent from multiple answer prediction, respectively.[6]

Simply copying the prompt question gives high $F1_{BLEU}$, which is natural since the questions were disambiguated using minimal edits. This justifies using $F1_{EDIT\text{-}F1}$ to evaluate semantic differences from the prompt question. In addition, we find that

[5] This problem has also been reported in other conditional generation tasks (Sountsov and Sarawagi, 2016; Stahlberg and Byrne, 2019); we leave it for future work.

[6] Note that a high $F1_{ans}$ and low $F1_{EDIT\text{-}F1}$ may not indicate bad question disambiguation. For instance, if a model correctly predicts one out of two answers and does not perform any edits to the question, it obtains high $F1_{ans}$ and zero $F1_{EDIT\text{-}F1}$, despite the error being in answer prediction.

Prompt question #1: Where was snow white and the huntsman filmed? **Reference:** Q: Where were beach scenes for snow white and huntsman predominantly filmed? / A: Marloes Sands Beach Q: Where was principal photography for snow white and huntsman filmed? / A: United Kingdom Q: Where was castle in snow white and huntsman filmed? / A: Gateholm island **Prediction of DISAMBIG-FIRST:** ($F1_{ans}$=0.40, $F1_{EDIT\text{-}F1}$=0.00) Q: Where was snow white and the huntsman filmed in 2017? / A: Marloes Sands Beach Q: Where was snow white and the huntsman filmed during the filming of Season 1 of the TV series? / A: Marloes Sands Beach **Prediction of SPANSEQGEN:** ($F1_{ans}$=0.80, $F1_{EDIT\text{-}F1}$=0.69) Q: Where was snow white and huntsman principal photography filmed / A: United Kingdom Q: Where were beach scenes for snow white and huntsman mostly filmed / A: Marloes Sands Beach
Prompt question #2: When was the city of new york founded? **Reference:** Q: When was city of new york founded by dutch and initially called new amsterdam? / A: 1624 Q: When was city of new york under english control and renamed to new york? / A: 1664 **Prediction of SPANSEQGEN:** ($F1_{ans}$=1.00, $F1_{EDIT\text{-}F1}$=0.67) Q: When was city of new york city founded with dutch protection? / A: 1624 Q: When was city of new york city founded and renamed with english name? / A: 1664

Table 5: Model predictions on samples from the development data. **(#1)** DISAMBIG-FIRST generates questions that look reasonable on the surface but don't match the facts. SPANSEQGEN produces the reasonable answers and questions, although not perfect. **(#2)** SPANSEQGEN produces correct answers and questions.

Reference has multiple answers	
Multiple answer prediction is correct	2%
Multiple answer prediction is partially correct†	40%
Multiple answer prediction is incorrect	14%
Reference has one answer	
Over-generated predictions	2%
Correct single answer prediction	26%
Incorrect single answer prediction	12%
Reference is incorrect	4%

Table 6: Analysis of predictions made by SPANSEQGEN with co-training, on 50 samples from the development data. Examples shown in Appendix (Table 10).
†In 15 out of 20 cases, the model generates only one answer.

Model	NQ-OPEN EM	$F1_{ans}$ (*all*)	$F1_{ans}$ (*multi*)
Dev			
Min et al. (2019b)	34.7	30.8	20.4
Asai et al. (2020)	31.7	29.7	19.7
Karpukhin et al. (2020)	39.8	35.2	**26.5**
SPANSEQGEN	**42.0**	**36.4**	24.8
Test			
Min et al. (2019b)	34.5	27.5	17.0
Asai et al. (2020)	32.6	27.9	17.7
Karpukhin et al. (2020)	41.5	30.1	**23.2**
SPANSEQGEN	**42.2**	**30.8**	20.7

Table 7: Zero-shot performance on multiple answer prediction of the models trained on NQ-OPEN. We report Exact Match (EM) on NQ-OPEN and $F1_{ans}$ on AMBIGNQ.

our QD model conditioned on all available context is better than other variants in overall metrics.

Performance is low overall, even given the gold answers, highlighting the challenge of the task. We think there are two major reasons. First, maximizing the likelihood of the output sequence can miss the importance of *edits* to the prompt question, leading the QD model to miss the information that is most important to differentiate one answer from the others. Second, there is a lack of annotated data, especially for question disambiguation which does not benefit from weakly supervised learning with NQ-OPEN; future work can explore how to maximize the use of supervision from other available data. It is also worth noting that the metric may miss edits that are semantically correct, but phrased differently (see Table 5, example 2).

6.3 Zero-shot results

Since AMBIGNQ provides an evaluation set with explicit sets of multiple answers, we can also test if models trained on partial supervision only (NQ-OPEN) are capable of producing full answer sets. In fact, the problem of ambiguity already exists in previous QA tasks, and a single labeled answer can be viewed as a sample from a multi-modal distribution of answers. This setting is important for modeling in domains where single-answer datasets are available but full annotations like in AMBIGNQ are not. To this end, we present a zero-shot setting where a system predicts multiple distinct answers without using AMBIGNQ training data. We include four NQ-OPEN models including ours, consisting of diverse approaches and model architec-

tures, as baselines. These models, when trained on NQ-OPEN, may be made to predict multiple answers via thresholding as described in Section 6.1.[7] Table 7 reports zero-shot performance. Although SPANSEQGEN outperforms Karpukhin et al. (2020) in the standard setting, it is worse in zero-shot $F1_{ans}$ (*multi*), potentially because thresholding exacerbates the problems that SPANSEQGEN has with long sequences (Section 6.2).

6.4 Error Analysis

Table 6 reports an analysis of predictions by SPANSEQGEN with co-training, based on 50 random samples from the development data; examples can be found in the Appendix (Table 10). When there are multiple reference answers, the model rarely gets all correct answers, although often generates a subset of them. In 15 out of 20 partially correct cases, the model produces only one answer, consistent with the under-generation we found in Section 6.2. In four out of those 15 cases, the model prediction is arguably the most likely answer,[8] but in the other 11 cases, it hard to argue for one answer over the other(s). It is also worth noting that accuracy on examples with a single answer is quite high, being correct in 13 out of 20 cases. This estimated accuracy on unambiguous questions is higher than state-of-the-art levels on NQ-OPEN (42 EM), suggesting that NQ-OPEN may substantially underestimate performance due to the prevalence of unmarked ambiguity. Together with our experimental results, this seems to indicate that recall of multiple answers is one of the primary challenges in AmbigQA.

7 Conclusion & Future Work

We introduced AMBIGQA, a new task that involves providing multiple possible answers to a potentially ambiguous open-domain question, and providing a disambiguated question corresponding to each answer. We constructed AMBIGNQ, a dataset with 14,042 annotations on NQ-OPEN questions. Our analysis shows the dataset contains diverse types of ambiguity, often not visible from the prompt question alone. We also introduced a first baseline model for producing multiple answers to open-domain questions, with experiments showing its effectiveness in learning from our data while highlighting possible areas for improvement.

Future research developing on AmbigQA models may include explicitly modeling ambiguity over events and entities or in the retrieval step, as well as improving performance on the difficult problems of answer recall and question disambiguation. Furthermore, future work may build on the AmbigQA task with more open-ended approaches such as (1) applying the approach to QA over structured data (such as ambiguous questions that require returning tables), (2) handling questions with no answer or ill-formed questions that require inferring and satisfying more complex ambiguous information needs, and (3) more carefully evaluating usefulness to end users.

[7] We allow using development data to tune the threshold γ, although this arguably makes our setting not zero-shot in the strictest sense.

[8] For example, a question "Who did `<title-of-the-song>`" is ambiguous, but a well-known performer of the song may be argued to be a more likely answer than its little-known songwriter.

Acknowledgments

This research was supported by ONR N00014-18-1-2826, DARPA N66001-19-2-403, the NSF (IIS-1252835, IIS-1562364), an Allen Distinguished Investigator Award, and the Sloan Fellowship. We thank Mandar Joshi, H2Lab members and the anonymous reviewers for their helpful comments and suggestions.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *SIGIR*.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *ICLR*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*.

Pavel Braslavski, Denis Savenkov, Eugene Agichtein, and Alina Dubatovka. 2017. What do you mean exactly? Analyzing clarification questions in CQA. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *ACL*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Matthew Dunn, Levent Sagun, Mike Higgins, Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Ahmed Elgohary, Denis Peskov, and Jordan L. Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *EMNLP*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *ICML*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Change, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a benchmark for question answering research. *TACL*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *ACL*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing question-answer meaning representations. In *NAACL*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard EM approach for weakly supervised question answering. In *EMNLP*.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *ACL*.

Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. In *NAACL*.

Pavel Sountsov and Sunita Sarawagi. 2016. Length bias in encoder decoder models and a case for global conditioning. In *EMNLP*.

Felix Stahlberg and Bill Byrne. 2019. On nmt search errors and model errors: Cat got your tongue? In *EMNLP*.

Ellen M Voorhees et al. 1999. The TREC-8 question answering track report. In *Trec*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and SUN Xu. 2019. Asking clarification questions in knowledge-based question answering. In *EMNLP*.

Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. Mimics: A large-scale data collection for search clarification. In *ACM International on Conference on Information and Knowledge Management*.

Y. Zhou and S. Goldman. 2004. Democratic co-learning. In *IEEE International Conference on Tools with Artificial Intelligence*.

A Data Collection Details

We use Amazon Mechanical Turk[9] and Spacro (Michael et al., 2018)[10] for crowdsourcing. All data was collected in February and March of 2020. We use the Google Search API[11] restricted to English Wikipedia for the search tool.

Crowdsourcing details. Figure 3 shows the interface used for generation and validation. We use an iframe to render Wikipedia pages in a mobile view, in order to provide the document format that they are familiar with, rather than the plain text with no formatting. When workers write the questions and the answers in the generation stage, we show appropriate error messages (e.g. when the written question is the same as the prompt question) or warning messages (e.g., when the answer is composed of more than 20 words) in order to give tight feedback. Workers produce free text answers which we instruct them to copy and paste from Wikipedia.

We pay 0.75 and 0.15 USD per prompt question for generation and validation, respectively. Generators may skip the prompt question if the answer is not found in Wikipedia, or the question is ill-formed, too subjective or too ambiguous, e.g., "When did the new tax cuts go into effect?"

Quality control. We only recruit full-time workers that are dedicated to our task. We were able to recruit full-time workers by requiring the minimum number of HITs that can be achieved by working 40 hours a week. We also host a public website for them to monitor the validated statuses, ask questions on examples that they do not understand the validated result, or claim on the validation which is incorrect in their opinion. We found it very useful to communicate with workers, give feedback, and fix the incorrect annotations.

Inter-annotator agreement. When two independent generators are evaluated on the answer list from each other, they obtain 60.8 $\mathrm{F1}_{\mathrm{ans}}$. Specifically, for 76% of questions, all annotations passed validation, either automatically because they exactly matched (37%) or because they were both accepted by validators (39%). In the remaining 24% of cases, one annotator missed a possible question-answer pair that the other one found, or included an invalid question-answer pair.

To assess validation quality, two co-authors annotated a random sample of 50 validations. The average $\mathrm{F1}_{\mathrm{ans}}$ between the co-authors and workers was 89.0%.

B Discrepancy between development and test in NQ-OPEN

In our experiments on AMBIGNQ, we found a significant discrepancy between the development and test sets. Upon further investigation, we identified that this is at least in part due to a distributional difference between the development and test sets of NQ-OPEN, upon which we built the data. As this may be important for other researchers working on NQ-OPEN, we detail our findings here.

Following Lee et al. (2019), NQ-OPEN is constructed by filtering NATURAL QUESTIONS to questions where at least one annotator provided a non-null short answer to the question.[12] While the training and development sets of NQ-OPEN were all drawn from the training set of NATURAL QUESTIONS, in which one annotator answered each question, the test set of NQ-OPEN is taken from its development set, which had five annotators per question.

This difference in number of annotators introduces a sampling bias: questions for which an annotator is less likely to find an answer are overrepresented in the NQ-OPEN test set, in comparison to training and development. Suppose, for example, that a randomly sampled annotator has a 50% chance of producing a short answer for some question q. Then q has a 50% chance of making it into NQ-OPEN's development set, but a $(1-.5^5 =)$ 97% chance of making it into test. Concretely, when each annotator is considered independently, 34.6% of the short answer annotations in the test set of NQ-OPEN are null answers, and the majority of annotations are null for 33.9% of questions.

As a consequence, there is a significant gap in model performance between development and test when they are evaluated under the same conditions. The official evaluation protocol for NQ-OPEN counts a prediction as correct if it matches any of the gold reference answers. Under these conditions, the gap between development and test appears marginal (Table 8, first two columns). However, as the NQ-OPEN test set was more compre-

[9] www.mturk.com

[10] github.com/julianmichael/spacro

[11] developers.google.com/custom-search/

[12] NATURAL QUESTIONS annotators answered each question with a set of short answers, which could be empty if there was no reasonable short answer. We refer to the empty cases as *null answers*. See Kwiatkowski et al. (2019) for details.

Model	Any		First	
	dev	test	dev	test
Min et al. (2019b)	34.7	34.5	32.4	25.7
Asai et al. (2020)	31.7	32.6	28.9	23.8
Karpukhin et al. (2020)	39.8	41.5	37.0	29.8
SPANSEQGEN	42.0	42.2	38.8	31.1

Table 8: Exact Match (EM) on NQ-OPEN of different models, counting a prediction as correct if it matches *Any* gold reference, or only the *First* non-null one.

hensively annotated than development, it has a more generous evaluation; the number of unique reference answers is 1.2 and 1.8 on development and test, respectively. In order to make the evaluation more consistent, we try evaluating models against the first reference answer only, and find a significant gap between development and test (5–8%) across all models (Table 8, last two columns).[13]

Despite this discrepancy, AMBIGNQ follows the setup and data split from NQ-OPEN providing consistency with prior work. Since the AMBIGNQ development and test sets were annotated under the same conditions, this discrepancy now shows up in the metrics. We leave the distribution shift of questions on the test data as one of challenges on AMBIGNQ.

C Data Analysis Details

Mismatches with NQ-OPEN. 29.4% of AMBIGNQ development examples do not include the NQ-OPEN answer. We analyze a random sample of 50 such questions, and present a breakdown in Table 9. We find that our answers are correct in 92% of cases, among which 44% of disagreements are due to mismatched spans, 22% are due to the NQ-OPEN answer being incorrect, and 14% are due to time-dependence in the question. Of the 8% of cases where our answer is incorrect, the NQ-OPEN answers are also incorrect over half the time, indicating that these may be difficult questions.

D Baseline Implementation Details

Evidence corpus. We use English Wikipedia dump from 2018-12-20 and 2020-01-20 for NQ-OPEN and AMBIGNQ, respectively. Following Karpukhin et al. (2020), we take the plain text and split passages to be up to 100 words each.

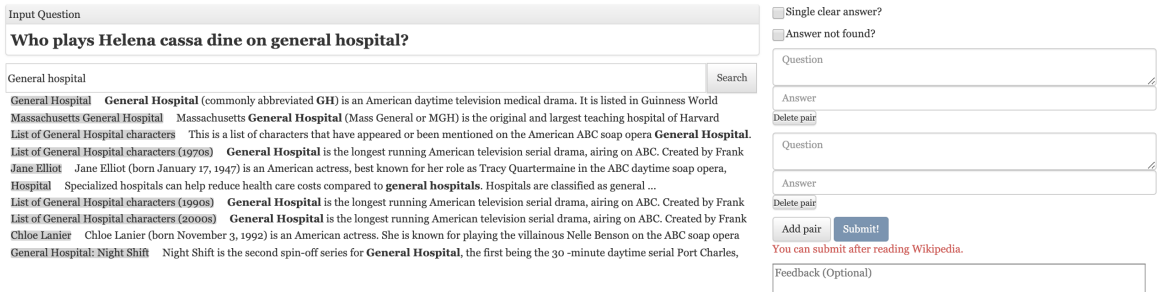
Model implementation. All models are implemented in PyTorch (Paszke et al., 2017), PyTorch-Transformers (Wolf et al., 2019) (for BERT) and fairseq (Ott et al., 2019) (for BART). We use $\text{BERT}_{\text{BASE}}$ and $\text{BART}_{\text{LARGE}}$ for all models. We use the exact same setup and hyperparameters for any process that we follow Karpukhin et al. (2020). For the passage retrieval through a dual encoder, we use the provided multi-setting trained model. For all BART-based models, we follow the default hyparameters from BART summarization code in fairseq, using one 32GB gpu. For finetuning, we change the learning rate to be $5e-6$ on both tasks. We use beam search for decoding the sequence. We train the model for 4 epochs (when trained on NQ-OPEN or pseudo-labelled data) or 15 epochs (when trained on AMBIGNQ), and take the best checkpoint based on the development data. Note that the perplexity of the output sequence does not correlate with the metric of interest (Exact Match, F1_{ans} or $\text{F1}_{\text{EDIT-F1}}$) as briefly discussed in Section 6.2, so using the metric of interest instead of perplexity is important for hyperparamter tuning or the choice of the best checkpoint.

Details in ensemble and co-training. We use an ensemble based on voting; the answers that are predicted by the highest number of models are chosen as the final answers. The number of models used in ensemble (C) is $C = 5$ before cotraining and $C = 4$ after cotraining. For co-training, we use $N = 2$ and $C = 6$, where N is the number of iteration and C is the number of models, in line with Algorithm 1. The choice of C is determined by taking the best combination of the models as follows. We train sixteen different models, using different hyperparamers including checkpoints from NQ-OPEN, learning rates, the order of the answers in the output sequence and the random seed. We then measure the development F1_{ans} on different combinations of the models with varying C ($4 \leq C \leq 6$) and take the best one.

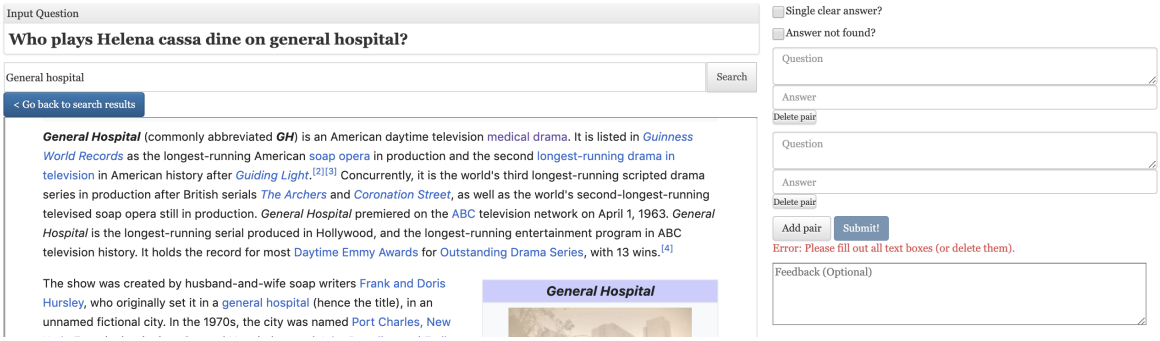
E Error Analysis of SPANSEQGEN

Table 10 reports an analysis of predictions by SPANSEQGEN, on 50 random samples from the development set. We refer to Section 6.4 for the discussions.

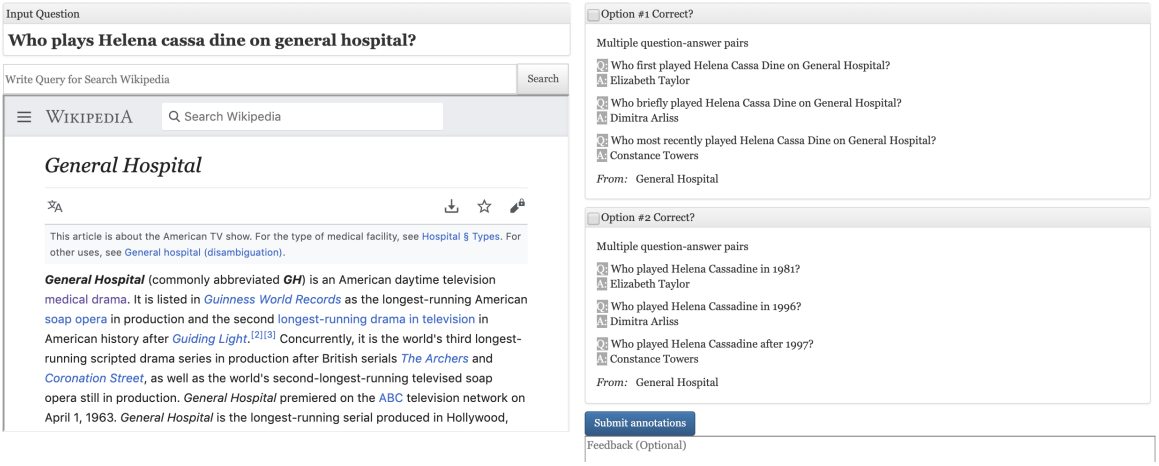
[13] It is unlikely that this discrepancy is due to overfitting on development, because the effect is consistent across models and not present on the other datasets that they are evaluated on.



(a) Interface in the generation stage when the workers write a query and see the search results.



(b) Interface in the generation stage when the workers click and read one of Wikipedia pages from the search results.



(c) Interface in the validation stage when the workers are given annotations from two generation workers and click the Wikipedia page that the generation workers have read.

Figure 3: Interface for crowdsourcing.

Answer span mismatch (44%)
Q: Who did the artwork for pink floyd's wall? NQ-OPEN answer: Gerald Anthony Scarfe AMBIGNQ answer: Q: Who did the art work for the album cover of Pink Floyd's The Wall? / A: Gerald Scarfe Q: Who was the cinematographer for Pink Floyd - The Wall (1982 film)? / A: Peter Biziou
NQ-OPEN answer incorporated as a question (2%)
Q: What award did leonardo dicaprio won for the revenant? NQ-OPEN answer: BAFTA Award; Academy Award for Best Actor; Golden Globe Award AMBIGNQ answer: Q: What British Academy Film Awards award did leonardo dicaprio won for the revenant? / A: Best Actor in a Leading Role Q: What Academy award did leonardo dicaprio won for the revenant? / A: Best Actor Q: What Golden Globe award did leonardo dicaprio won for the revenant? / A: Best Actor in a Motion Picture – Drama (Other question-answer pairs omitted)
NQ-OPEN answer less specific (10%)
Q: When was the nba 3 point line introduced? NQ-OPEN answer: 1979 AMBIGNQ answer: June 1979
NQ-OPEN answer incorrect and our answers include all possible answers (22%)
Q: Who was inducted into the national inventors hall of fame first? NQ-OPEN answer: John Fitch AMBIGNQ answer: Thomas Edison *Comment:* Thomas Edison inducted in 1973, John Fitch inducted in 2006. John Fitch is mentioned as the earliest born inventor inducted.[†]
Mismatch from time-dependence (14%)
Q: Who has the most home runs in the home run derby? NQ-OPEN answer: Todd Frazier AMBIGNQ answer: Q: Who has the most home runs in the the TV show the home run derby? / A: Mickey Mantle; Mickey Charles Mantle Q: Who has the most home runs in the annual competition the home run derby? / A: Joc Russell Pederson; Joc Pederson
NQ-OPEN answer is reasonable and our answers miss it (4%)
Q: Who was the first person to settle dodge city? NQ-OPEN answer: civilians AMBIGNQ answer: Henry J. Sitler
NQ-OPEN answer incorrect but our answers miss another possible answer (4%)
Q: In which year were chips used inside the computer for the first time? NQ-OPEN answer: 1975 AMBIGNQ answer: 1962 *Comment:* The years that the chips were used for the first time in the prototype and the production are 1962 and 1974, respectively, and can be both included.[‡]

Table 9: Breakdown of cases that NQ-OPEN answer is not included in AMBIGNQ answers.

[†]en.wikipedia.org/wiki/List_of_National_Inventors_Hall_of_Fame_inductees

[‡]en.wikipedia.org/wiki/History_of_computing_hardware_(1960s%E2%80%93present)

Reference has multiple answers; Multiple answer prediction is correct (2%) **Prompt question:** Who was england's prime minister during ww1? **Reference:** H. H. Asquith (*beginning of WW1*), David Lloyd George (*end of WW1*) **Prediction:** ($F1_{ans}$=1.00) H. H. Asquith, David Lloyd George
Reference has multiple answers; Multiple answer prediction is partially correct (40%) **Prompt question:** Who played kelly on the drew carey show? **NQ-OPEN answer:** Cynthia Watros **Reference:** Cynthia Watros (*as Kellie N.*), Jenny McCarthy (*as M. Kelly*), Brett Butler (*as G. Kelly*), Anna Gunn (*as Kelly W.*) **Prediction:** ($F1_{ans}$=0.40): Brett Butler
Reference has multiple answers; Multiple answer prediction is incorrect (14%) **Prompt question:** Who plays the white queen in alice through the looking glass? **Reference:** Amelia Crouch (*young White Queen*), Anne Hathaway (*adult White Queen*) **Prediction:** ($F1_{ans}$=0.00): Helena Bonham Carter[†]
Reference has one answer; over-generated predictions (2%) **Prompt question:** How many times csk reached final in ipl? **Reference:** eight **Prediction:** ($F1_{ans}$=66.7): eight, seven[‡]
Reference has one answer; correct single answer prediction (26%) **Prompt question:** When did the 5th circuit became the 11th circuit? **Reference:** October 1, 1981 **Prediction:** ($F1_{ans}$=100.0): October 1, 1981
Reference has one answer; incorrect single answer prediction (12%) **Prompt question:** Who is considered the home team for super bowl 52? **Reference:** New England Patriots **Prediction:** ($F1_{ans}$=0.0): Atlanta Falcons
Reference is incorrect (4%) **Prompt question:** Who has won the most trophies man utd or liverpool? **Reference:** Man utd (*trophies*), Liverpool (*FIFA and UEFA Cups*) **Prediction:** ($F1_{ans}$=66.7): Manchester United

Table 10: Analysis of multiple answer predictions made by SPANSEQGEN with co-training, on 50 samples from the development data. Rewrites are omitted but differentiation of multiple answers is denoted as a keyword in *italic*.

[†]Helena Bonham Carter played Red Queen.

[‡]In fact, the model may have found time-dependency, because the eighth event happened only in 2019.