IIRC: A Dataset of Incomplete Information Reading Comprehension Questions

James Ferguson◇∗   **Matt Gardner**♣   **Hannaneh Hajishirzi**◇♣
Tushar Khot♣   **Pradeep Dasigi**♣
◇University of Washington
♣Allen Institute for AI
{jfferg,hannaneh}@cs.washington.edu
{mattg,tushark,pradeepd}@allenai.org

Abstract

Humans often have to read multiple documents to address their information needs. However, most existing reading comprehension (RC) tasks only focus on questions for which the contexts provide all the information required to answer them, thus not evaluating a system's performance at identifying a potential lack of sufficient information and locating sources for that information. To fill this gap, we present a dataset, **IIRC**, with more than 13K questions over paragraphs from English Wikipedia that provide only partial information to answer them, with the missing information occurring in one or more linked documents. The questions were written by crowd workers who did not have access to any of the linked documents, leading to questions that have little lexical overlap with the contexts where the answers appear. This process also gave many questions without answers, and those that require discrete reasoning, increasing the difficulty of the task. We follow recent modeling work on various reading comprehension datasets to construct a baseline model for this dataset, finding that it achieves 31.1% F1 on this task, while estimated human performance is 88.4%. The dataset, code for the baseline system, and a leaderboard can be found at https://allennlp.org/iirc.

1 Introduction

Humans often read text with the goal of obtaining information. Given that a single document is unlikely to contain all the information a reader might need, the reading process frequently involves identifying the information present in the given document, and what is missing, followed by locating a different source that could potentially contain the missing information. Most recent read-

Wilhelm Müller was born on 7 October 1794 at **Dessau**, the son of a tailor. In 1813-1814 he took part, as a volunteer in the Prussian army, in the national rising against **Napoleon**. He participated in the battles of **Lützen**, **Bautzen**, **Hanau** and **Kulm**. In 1814 he returned to his studies at Berlin. Müller's son, **Friedrich Max Müller**, was an English orientalist who founded the comparative study of religions.

Which battle Wilhelm Müller fought in while in the Prussian army had the most casualties?

Battle of Lützen (1813)
Napoleon lost 19,655 men, while the Prussians lost 8,500 men and the Russians lost 3,500 men

Battle of Bautzen
Losses on both sides totaled around 20,000.

Battle of Hanau
Overall, 4,500 French soldiers and 9,000 allied soldiers were lost in the battle.

Battle of Kulm
The French lost more than half of the pursuing force of 34,000; The allies lost approximately 13,000 soldiers.

Figure 1: An example from **IIRC**. At the top is a context paragraph which provides only partial information required to answer the question. The bold spans in the context indicate links to other Wikipedia pages. The colored boxes below the question show snippets from four of these pages that provide the missing information for answering the question. The answer is the underlined span.

ing comprehension tasks, such as SQuAD 2.0 (Rajpurkar et al., 2018), DROP (Dua et al., 2019b), or Quoref (Dasigi et al., 2019), evaluate models using a relatively simpler setup where all the information required to answer the questions (including judging them as being unanswerable) is provided in the associated contexts. While this setup has led to significant advances in reading comprehension (Ran et al., 2019; Zhang et al., 2020), the tasks are still limited since they do not evaluate the capability of models at identifying precisely what information, if any, is missing to answer a question, and where that information might be found.

On the other hand, open-domain question answering tasks (Chen et al., 2017; Joshi et al., 2017; Dhingra et al., 2017) present a model with a ques-

---
∗Work done as an intern at the Allen Institute for AI.

tion by itself, requiring the model to retrieve relevant information from some corpus. However, this approach loses grounding in a particular passage of text, and it has so far been challenging to collect diverse, complex question in this setting.

Alternatively, complex questions grounded in context can be converted to open-domain or incomplete-information QA datasets such as HotpotQA (Yang et al., 2018). However, they do not capture the information-seeking questions that arise from reading a single document with partial information (Min et al., 2019b; Chen and Durrett, 2019).

We present a new dataset of incomplete information reading comprehension questions, **IIRC**, to address both of these limitations. **IIRC** is a crowdsourced dataset of 13441 questions over 5698 paragraphs from English Wikipedia, with most of the questions requiring information from one or more documents hyperlinked to the associated paragraphs, in addition to the original paragraphs themselves. Our crowdsourcing process (Section 2) ensures the questions are naturally information-seeking by decoupling question and answer collection pipelines. Crowd workers are instructed to ask follow-up questions after reading a paragraph, giving links to pages where they would expect to find the answer. This process results in questions like the one shown in Figure 1. As illustrated by the example, this setup results in questions requiring complex reasoning, with an estimated 39% of the questions in **IIRC** requiring discrete reasoning. Moreover, 30% of the questions in **IIRC** require more than one linked document in addition to the original paragraph and 30% of them are unanswerable even given the additional information. When present, the answers are either extracted spans, boolean, or values resulting from numerical operations.

To evaluate the quality of the data, we run experiments with a modified version of NumNet+ (Ran et al., 2019), a state-of-the-art model from DROP (Dua et al., 2019b), chosen because a significant portion of questions in **IIRC** require numerical reasoning similar to that found in DROP. Because DROP uses only a single paragraph of context, we add a two-stage pipeline to retrieve necessary context for the model from the linked articles. The pipeline first identifies which links are pertinent, and then selects the most relevant passage from each of those links, concatenating them to serve as input for the model (Section 3).

This baseline achieves an F_1 score of 31.1% on **IIRC**, while the estimated human performance is 88.4% F_1. Even giving the model oracle pipeline components results in a performance of only 70.3%. Taken together, these results show that substantial modeling is needed both to identify and retrieve missing information, and to combine the retrieved information to answer the question (Section 4). We additionally perform qualitative analysis of the data, and find that the errors of the baseline model are evenly split between retrieving incorrect information, identifying unanswerable questions, and successfully reasoning over the retrieved information.

By construction, all examples in **IIRC** require identifying missing information. Even though current model performance is quite low, a model trained on this data could theoretically leverage that fact to achieve artificially high performance on test data, because it does not have to first determine *whether* more information is needed. To account for this issue, we additionally sample questions from SQuAD 2.0 (Rajpurkar et al., 2018) and DROP (Dua et al., 2019b), which have similar question language to what is in **IIRC**, putting forward this kind of combined evaluation as a challenging benchmark for the community. Predictably, our baseline model performs substantially worse in this setting, reaching only 28% F_1 on the **IIRC** portion of this combined evaluation (Section 5).

2 Building IIRC

We used Wikipedia to build **IIRC** and relied on the fact that entities in Wikipedia articles are linked to other articles about those entities, providing more information about them. Our goals were to build a dataset with naturally information-seeking questions anchored in paragraphs with incomplete information, such that identifying the location of missing information is non-trivial, and answering the questions would require complex cross-document reasoning.

We ensured that the questions are information-seeking by separating question and answer collection processes, and by not providing the question writers access to the contexts where the answers occur. This process also ensured that we get questions that have minimal lexical overlap with the answer contexts. We used Wikipedia paragraphs with many outgoing links to increase the difficulty of identifying the articles that provide the missing information. To ensure complex cross-document

How many different directors did Prada work with in 1976 and 1977?

Jaya Prada
She became a huge star in 1976 with major hit films. Director **K. Balachander's** black-and-white film **Anthuleni Katha** (1976) showcased her dramatic skills; **K. Viswanath's** color film **Siri Siri Muvva** (1976) showed her playing a mute girl with excellent dancing skills; and her title role as Sita in the big-budget mythological film Seetha Kalyanam confirmed her versatility. In 1977, she starred in Adavi Ramudu, which broke box office records and which permanently cemented her star status. Filmmaker **Vijay** introduced her to **Kannada cinema** in his 1977 super-hit movie **Sanadi Appanna** alongside Kannada matinee idol **Raj Kumar**.

> **Seeta Kalyanam**
> Seeta Kalyanam is a 1976 Telugu epic, mythological, drama film directed by Bapu

> **Adavi Ramudu**
> Adavi Ramudu is a 1977 Telugu Action film directed by K. Raghavendra Rao

Answer type: Numeric
Answer: 5 directors

Link prediction: Hard	Retrieval: Bridge	Reasoning: Discrete-numeric

---

In what country did Bain attend the doctoral seminars of Wlad Godzich?

Thomas Bain
Bain was born in **London**. He lived **Kingston upon Thames** attending prep school at Highfield School (**Liphook**, Hampshire). He suffered from **Dyslexia** ... He completed M. Phil at the Geneva-based IUEE (Institute for European Studies), and later attended the doctoral seminars of **Wlad Godzich** in the **University of Geneva**.

> **University of Geneva**
> The University of Geneva is a public research university located in Geneva, Switzerland

Answer type: Span
Answer: Switzerland

Link prediction: Medium	Retrieval: Bridge	Reasoning: Non-discrete

---

Was Tip O'Neill working as a politician the year O'Donnell provided testimony to Arlen Specter?

Kenneth O'Donnell
On May 18, 1964, O'Donnell provided testimony to **Norman Redlich** and **Arlen Specter**, assistant counsel for the **Warren Commission**. O'Donnell stated that it was his impression that the shots fired at Kennedy came from the right rear ... In his 1987 autobiography Man of the House, former **House Speaker Tip O'Neill** wrote that he had dinner with O'Donnell and Powers in 1968, and that both men indicated that two shots were fired from behind the fence on the grassy knoll at Dealey Plaza

> **Tip O'Neill**
> Thomas Phillip "Tip" O'Neill Jr. was an American politician, representing northern Boston, Massachusetts, as a Democrat from 1953 to 1987

Answer type: Binary
Answer: Yes

Link prediction: Hard	Retrieval: Cross Context	Reasoning: Discrete-temporal

---

What was the first film Metro-Goldwyn-Mayer released?

Ben Carré
In the 1920s, Carré worked as a freelance art director designing sets for **The Red Lily**, directed by Fred Niblo and starring **Ramon Novarro** and designing the catacombs for **The Phantom of the Opera**. Carré worked on a string of films for the newly formed **Metro-Goldwyn-Mayer**, starting with The Masked Bird and including **La Bohème**, directed by **King Vidor**.

> **Metro-Goldwyn-Mayer**
> MGM produced more than 100 feature films in its first two years.

Answer type: None
Answer: N/A

Link prediction: Easy	Retrieval: Linked context only	Reasoning: Non-discrete

Figure 2: Examples from **IIRC**, labeled with what kinds of processing are required to answer each question. See Table 1 for more details. The passages on the left are the original passage, with bold spans indicating links. The highlighted sections contain the necessary context found in linked articles. Purple highlights indicate either the answer, for the second question, or the information used to compute the answer.

reasoning, we asked the crowd workers to create questions that need information from the seed paragraph as well as one or more linked articles. This constraint resulted in questions that are answerable neither from the original paragraph alone, nor from one of the linked articles alone, often requiring over 3+ passages to answer. The remainder of this section describes our data collection process.

2.1 Seed Paragraphs

We started by collecting paragraphs from Wikipedia articles containing ten or more links to other Wikipedia articles. This resulted in roughly 130K passages. We then created two separate crowdsourcing tasks on Amazon Mechanical Turk[1]; one for collecting questions, and one for collecting answers. Workers for each task were chosen based on a qualification task. Their submissions were manually inspected, and those that produced high quality questions and correct answers, respectively, continued to work on the main annotation tasks.

2.2 Collecting Questions

Given a paragraph with links to other articles highlighted, crowd workers were tasked with writing questions that require information from the paragraph, as well as from one or more of linked articles. Workers could see the links, and the titles of

---
[1]www.mturk.com

	Type	Description	Percentage
Link Prediction	Easy	Link is explicitly mentioned in the question	41%
	Medium	Context is required to determine link target	47%
	Hard	Context is required to determine link targets and number of links	12%
Retrieval	Linked context only	Original passage is not necessary to answer question	14%
	Bridge	Original passage is only necessary to determine links	57%
	Cross context	Original passage is necessary to find relevant information in links	29%
Reasoning	Non-discrete	No discrete reasoning is required	61%
	Discrete-numeric	Discrete reasoning is required	11%
	Discrete-temporal	Discrete reasoning involving time is required	28%
Answer	Span	Answer is one or more spans selected from question or context	45%
	Numeric	Answer is a number (with a unit provided)	17%
	Binary	Answer is either *yes* or *no*	8%
	None	Question cannot be answered given the provided context	30%

Table 1: Frequency of different types of retrieval, reasoning, and answers that appear in **IIRC**.

the articles they pointed to, but not the contents of the linked articles. Since the linked articles were not provided, the workers were asked to frame questions based on the information they think would be contained in the those articles. For each human intelligence task (HIT), workers were presented with a collection of ten paragraphs, and were asked to write a total of ten questions using any of those paragraphs, with two questions requiring following two or more links. For example given a passage about an actor that mentions *Rasulala had roles in Cool Breeze (1972), Blacula (1972), and Willie Dynamite (1973)*, an example of a question requiring multiple links would be *How many different directors did Rasulala work with in 1972?*.

In order to minimize questions with shortcut reasoning, we provided workers extensive instructions along with examples of good and bad questions to ask. Examples of bad questions included questions that did not require any links - *Who did the Arakanese kings compare themselves to?* when the context included *They compared themselves to Sultans*; and questions that did not require information from the original passage - *What was Syed Alaol's most famous work?* when the context included *Syed Alaol was a renowned poet*.

In addition to writing questions, workers also provided the context from the original paragraph that they thought would be necessary to answer the question, as well as the links they expected to contain the remaining necessary information. Workers were paid $4.00 per set of ten questions, and reported taking 25 minutes on average, coming out to $9.60 per hour. 40 workers passed the qualification and worked on the main task.

2.3 Collecting Answers

For the answer task, workers were given a collection of ten questions, their respective original paragraphs, and the context/links selected by the question writer. For each paragraph, workers were able to see the links, and could follow them to view the text, not including tables or images, of the linked document.

They were then asked to select an answer from one of four types: a span of text from either the question or a document, a number and unit, yes/no, or *no answer*. For answerable questions, i.e. any of the first three types, they were additionally asked to provide the minimal context span(s), necessary to answer the question. For unanswerable questions, there is typically no indication that the answer is not given, so no such context can be provided. For example, the following question was written for a passage about a ship called the Italia: *Who was the mayor of New York City when Italia was transferred to Genoa-NYC?* Following the link to New York City mentions the current mayor, but not past mayors, making it unanswerable.

Annotators were also given the option of labeling a question as bad if it didn't make sense, and these bad questions were then filtered out. For example, if an annotator misinterpreted the passage when writing the question as in the case of the following question written about a horse, Crystal Ocean, and St Leger, which the annotator thought was a horse, but is actually a horse race: *Is Crystal Ocean taller than St Leger?*. Additionally, A small percentage of questions that can be answered from the original paragraph alone were also marked as being bad.

For the training set, comprising 80% of the data, each question was answered by a single annotator.

Number of questions	13441
Number of passages	5698
Average number of links per passage	14.5
Average passage length (words)	197.5
Average question length (words)	13.6

Table 2: Statistics of **IIRC**.

For the development and test sets, comprising 10% each, three annotators answered each question, and only questions where at least two annotators agreed on the answer were kept. Workers were paid $3.00 per set of ten answers, and reported taking 20 minutes on average, coming out to $9.00 per hour. 33 workers passed the qualification and worked on the main task.

2.4 Dataset Analysis

In Figure 2 we show some examples from **IIRC**, labeled with different kinds of processing required to solve them. The types are described in detail in Table 1. These types and percentages were computed from a manual analysis of 100 examples.

In Table 2 we provide some global statistics of the dataset. In total, there are 13441 questions over 5698 passages. Each passage contains an average of 14.5 outgoing links. Using the context provided by the answer annotators, we are able to compute a distribution of the number of links required to answer questions in the dataset, included in Table 4. While the majority of questions require information from only one linked document in addition to the original paragraph, 30% of questions require two or more, with some requiring reasoning over as many as 12 documents to reach the answer. This variability in the number of context documents adds an extra layer of complexity to the task.

We also analyzed the initial trigrams of questions to quantify the diversity of questions in the dataset. We found that the most common type of questions, those related to time (eg "How old was", "How long did"), make up 15% of questions. There are 3.5k different initial trigrams across the 10.8k questions in the training set.

3 Modeling IIRC

3.1 Task Overview

Formally, a system tackling **IIRC** is provided with the following inputs: a question Q; a passage P ; a set of links contained in the passage, $L = \{l_i\}_{i=1}^N$; and the set of articles those links lead to, $\mathbf{A} = \{a^i\}_{i=1}^N$. The surface form of each link, l_i is a

sequence of tokens in P and is linked to an article a^i. The target output is either a number, a sequence of tokens in one of P, Q, or a^i, Yes, No, or NULL (for unanswerable questions).

3.2 Baseline Model

To evaluate the difficulty of **IIRC**, we construct a baseline model adapted from a state-of-the-art model built for DROP. We choose a DROP model due to the inclusion of numerical reasoning questions in our dataset. Because the model was not originally used for data requiring multiple paragraphs and retrieval, we first predict relevant context to serve as input to the QA model using a pipeline with three stages:

1. Identify relevant links
2. Select passages from linked articles
3. Pass the concatenated passages to a QA model

3.2.1 Identifying Links

To identify the set of relevant links, L', in a passage, P, for a question, Q, the model first encodes the concatenation of the question and original passage using BERT (Devlin et al., 2019). It then concatenates the encoded representations of the first and last tokens of each link as input to a scoring function, following the span classification procedure used by Joshi et al. (2013), selecting any links that score above a threshold g.

$$P' = \text{BERT}([Q||P])$$
$$\text{Score}(l) = f([p'_i||p'_j]), \quad l = (p_i...p_j, a)$$
$$L' = \{l : \text{Score}(l) > g\}$$

where l is a link covering tokens $p_i...p_j$ linking to article a.

3.2.2 Selecting Context

Given the set, L' from the previous step, the model then must select relevant context passages from the documents. For each document, it first splits the document into overlapping windows[2], $w_0, w_1...w_n$. Each window is then concatenated with the question and prepended with a CLS token, and encoded with BERT. The encoded CLS tokens are then passed through a linear predictor to score each window, and the highest scoring sections from each document are concatenated as context for the final

---

[2]See section 4.2 for more details.

model, C.

$$c_{a_i} = \max_{w_j \in \text{Split}(a_i)} f(\text{BERT}([Q||w_j]))$$
$$C = [c_{a_i} : a_i \in L']$$

3.2.3 QA Model

As mentioned above, the final step in the pipeline is passing the concatenated context, along with the question and a selected window from the original passage, as input to a QA model. For our experiments, we use NumNet+, because it is the best performing model on the DROP leaderboard with publicly available code. At a high level, NumNet+ encodes the input using RoBERTa (Liu et al., 2019), as well as a numerical reasoning component. It then passes these into a classifier to determine the type of answer expected by the question, which we modified by adding binary and unanswerable as additional answer types. This model is trained using the gold context for answerable questions, and predicted context for unanswerable questions. We do this because by definition, unanswerable questions do not have annotated answer context.

4 Experiments

4.1 Evaluation Metrics

We use two evaluation metrics to compare model performance: Exact-Match (EM), and a numeracy-focused (macro-averaged) F1 score, which measures overlap between a bag-of-words representation of the gold and predicted answers. Due to the number of numeric answers in the data, we follow the evaluation methods used by DROP (Dua et al., 2019b).

Specifically, we employ the same implementation of Exact-Match accuracy as used by SQuAD (Rajpurkar et al., 2016), which removes articles and does other simple normalization, and our F1 score is based on that used by SQuAD. We define F1 to be 0 when there is a number mismatch between the gold and predicted answers, regardless of other word overlap. When an answer has multiple spans, we first perform a one-to-one alignment greedily based on bag-of-word overlap on the set of spans and then compute average F1 over each span. For numeric answers, we ignore the units. Binary and unanswerable questions are both treated as span questions. In the unanswerable case, the answer is a special NONE token, and in the binary case, the answer is either *yes* or *no*.

4.2 Implementation Details

For the link selection model, we initialized the encoder with pretrained BERT-base, and fine-tuned it during training. For the scoring function, we used a single linear layer with a sigmoid activation function. The model was trained using Adam, and the score threshold to select links was set to 0.5. Additionally, we truncated any passages longer than 512 tokens to 512. This occurred in less than 1% of the data. This model is trained using a cross-entropy objective with the information provided in the gold context by annotators. Any links pointing to articles with an annotated context span are labeled 1, and all other links are labeled 0.

For the passage selection model, we again initialized the encoder with pretrained BERT-base, and fine-tuned it during training. We set the window size such that the concatenation of all selected contexts, along with the question and a selection from the original passage, has max length 512. More specifically, using the number of links, N_l selected in the previous step, for a question with N_Q tokens, we set the window size to be $\frac{512-(N_Q)}{N_l+1}$. We set the stride to be $\frac{1}{4}$ the window size, i.e. if the first window contains tokens $[0, 200]$, the second window would contain $[50, 250]$. We used a single linear layer with a sigmoid activation as the scoring function. We train this model with a cross-entropy objective. We use the gold context provided by annotators, labeling sections that contain the entirety of the annotated context 1, and all other sections 0.

For NumNet+, we followed the hyperparameter and training settings specified in the original paper (Ran et al., 2019). We trained the model on gold context provided by annotators when available, i.e. for answerable questions, and predicted context from the previous steps otherwise.

4.3 Results and Discussion

Full Task Results Table 3 presents the performance of the baseline model. It additionally shows the results of using gold information at each stage of the pipeline, as well as human performance on computed on a subset of 200 examples from the test set. The model achieves $31.1\% \ F_1$, which is well below the human performance of 88.4%. Even with the benefit of the gold input, there is still room for improvement on reasoning over multiple contexts, as performance is still 18% absolute below human levels. The model does a good job of predicting the relevant links, as evidenced by the fact that us-

Model	Dev		Test	
	EM	F1	EM	F1
Full model	29.6	33.0	27.7	31.1
Oracle L	30.9	34.7	29.0	32.5
Oracle L+C	63.9	69.2	65.6	70.3
Human	-	-	85.7	88.4

Table 3: Baseline and oracle results on **IIRC**. Human evaluation was obtained from a subset of 200 examples from the test set. We evaluate the model when given oracle links (L) and retrieved contexts (C). Retrieving the correct contexts is a significant challenge, but even given oracle contexts there is a substantial gap between model and human performance.

Number of links	EM	F1
1 (70%)	33.2	36.7
2 (23%)	27.0	30.6
3 (4%)	25.9	31.5
4+ (3%)	40.9	43.4

Table 4: Exact match and F1 of the baseline model on the **IIRC** dev set broken down by number of links necessary to answer the question. The numbers in parentheses are the percentage of questions in the full dataset that require that number of context documents.

ing the gold links only improves performance by 1 point, but still struggles to identify the appropriate context within the linked documents. This is likely due to annotators not being able to see the linked context when the questions are written. This makes this step more difficult by not providing the model with surface-level lexical cues in the question that it could use to easily select the appropriate context.

Analysis of Number of Linked Documents Table 4 shows the results of running the full pipeline broken down according to the number of linked documents required to answer the question. These performance differences are the result of a few factors. The first is the fact that the more links required to answer a question, the more chances there are for failure to retrieve the necessary information. This is exacerbated by the pipeline nature of our baseline model. However, the spike in performance for questions requiring four or more links is caused by the number of unanswerable questions. Nearly half of the questions in that category are unanswerable, and the model largely predicts *No Answer* on those questions. Finally, the distribution of question types is different conditioned on the number of links. Questions that require more links often also require some form of discrete reasoning, which is more difficult for the model to handle.

Answer Type	EM	F1
Span	24.0	29.1
Number	20.4	-
Binary	56.5	-
No Answer	32.4	-

Table 5: Exact match and F1 of the baseline model on the **IIRC** dev set broken down by answer type. F1 equals EM for non-span types, so is not repeated.

Input	P	R	F1
Constant baseline	26.7	100.0	42.1
Question only	61.8	54.9	58.1
Question + Passage	64.2	54.9	59.2
Question + Pred Context	62.3	70.1	66.0

Table 6: Precision, recall, and F1 of identifying unanswerable questions in the dev set with various baselines that use different combinations of the question, original passage, and predicted context.

Analyzing Different Answer Types Table 5 shows the performance broken down according to the type of answer each question has. The model performs worst on questions with numeric answers. This is due to the fact that these questions often require the model to do arithmetic to solve, which, as discussed above, the model struggles with relative to other types of questions.

Unanswerable Questions Table 6 shows how well a simple model can identify unanswerable questions with varying amounts of information. We set this up as a binary prediction, either answerable or not, and use a linear classifier that takes the BERT `CLS` token as input. We also include the result of always predicting unanswerable as a baseline. When the model can only see the question, it improves over the baseline by around 10 F1, meaning that there is some signal in the question alone, without any context.

Some types of questions are more likely to be unanswerable, such as those asking for information with regards to a specific year, i.e. *What was the population of New York in 1989?*. This is caused by Wikipedia more generally including current statistics, but not including a specific information for all previous years. Additionally adding the original passage does not significantly improve performance. This is not surprising, as the original passage always contains information relevant to the question, and the question annotators could see that text when writing the question.

4.4 Error Analysis

In order to better understand the challenges of the dataset, we manually analyzed 100 erroneous predictions by the model.

Incorrect context (39%) These are the cases where the model identified the correct links but selected the wrong portion of the linked document. It often selects semantically similar context but misses the crucial information, e.g. selecting the duration instead of end date.

Modeling errors (32%) These are the cases in which the context passed to the final QA model contained all of the necessary information, but the model failed to predict the correct answer. This occurred most commonly for questions requiring math, with the model including unnecessary dates in the computation, resulting in predictions that were orders of magnitude off. For example, predicting *-1984* when the question was asking for the age of a person.

Identifying unanswerable questions (24%) In these cases, the QA model was provided with related context that was missing crucial information, similar to the first class of errors. However, in this case, the full articles also did not contain the necessary information. In these cases the model often selected a related entity, ie for a question asking *In which ocean is the island nation located?*, the model predicted the island nation, *Papua New Guinea* as opposed to the ocean, which was not mentioned.

Insufficient Links (5%) These are cases where insufficient links were selected from the original passage, thus not providing enough information to answer the question. While the model can handle over-selection of links, we found that the vast majority of the time, the system correctly identified both the necessary and sufficient links, rarely over-predicting the required links.

5 Combined Evaluation

By construction, all the questions in **IIRC** require more than the original paragraph to answer. This means that a reading comprehension model built for **IIRC** does not actually have to detect *whether* more information is required than what is in the given paragraph, as it can always assume that this is true. In order to combat this bias, we recommend an additional, more stringent evaluation that

| Training | Links | | | QA | |
Data	P	R	F1	EM	F1
IIRC	88	98	93	32.0	35.6
IIRC + S + D	85	79	82	24.6	28.0

Table 7: Results for link identification and QA when training the baseline model on **IIRC** and sampled questions from SQuAD (S) and DROP (D).

combines **IIRC** with other reading comprehension datasets that do not require retrieving additional information. This is in line with recently-recommended evaluation methodologies for reading comprehension models (Talmor and Berant, 2019; Dua et al., 2019a).

In this section, we present the results of one such evaluation. Noting that **IIRC** has similar properties to both SQuAD 2.0 (Rajpurkar et al., 2018) and DROP (Dua et al., 2019b), and even similar question language in places, we sample questions from these datasets to form a combined dataset for training and evaluating our baseline model.

Sampling from SQuAD 2.0 and DROP To construct the data for the combined evaluation, we sample an additional 3360 questions from SQuAD 2.0 and DROP, so that they make up 20% of the questions in the new data. We sample from SQuAD 2.0 and DROP with a ratio of $3 : 1$ in order to match the distribution of numeric questions in **IIRC** and used a Wikifier (Cheng and Roth, 2013) to identify the links to Wikipedia articles in them.

Results We train the full baseline on **IIRC** augmented with sampled DROP and SQuAD data, and evaluate it on the **IIRC** dev set without any additional sampled data. We don't include any sampled data in the evaluation in order to make a direct comparison to **IIRC** to see how adding questions that don't require external context affects the model's ability to identify necessary context. We also include the results of running just the link identification model trained under each setting. We show the results in table 7. Adding the extra dimension of determining whether extra information is necessary causes the model to become less confident, significantly hurting recall on link selection. These missed predictions then propagate down the pipeline, resulting in a loss of almost 8% F_1 when compared to a model trained on just **IIRC**.

We also evaluated the combination model on a dev set with sampled SQuAD and DROP data to see how well the model learned to identify that

no external information was necessary. Given that none of the SQuAD or DROP data requires external links, this evaluation could only negatively impact precision. We find that precision dropped by 8 points, compared to a drop of 28 points when the model trained only on **IIRC** was used, indicating that the model is able to learn to identify when no external information is required.

6 Related Work

Questions requiring multiple contexts Prior multi-context reading comprehension datasets were built by starting from discontiguous contexts, and forming compositional questions by stringing multiple facts either by relying on knowledge graphs as in QAngaroo (Welbl et al., 2018), or by having crowdworkers do so, as in HotpotQA (Yang et al., 2018). It has been shown that many of these questions can be answered by focusing on just one of the facts used for building the questions (Min et al., 2019b). In contrast, each question in **IIRC** was written by a crowdworker who had access to just one paragraph, with the goal of obtaining information missing in it, thus minimizing lexical overlap between questions and the answer contexts. Additionally, **IIRC** provides a unique question type: questions requiring aggregating information from many related documents, such as the second question in Figure 2.

Separation of questions from answer contexts Many prior datasets (e.g.: WhoDidWhat (Onishi et al., 2016), NewsQA (Trischler et al., 2016), DuoRC (Saha et al., 2018), Natural Questions (Kwiatkowski et al., 2019), TyDiQA (Clark et al., 2020)) have tried to remove simple lexical heuristics from reading comprehension tasks by separating the contexts that questions are anchored in from those that are used to answer them. **IIRC** also separates the two contexts, but is unique given that the linked documents elaborate on the information present in the original contexts, naturally giving rise to follow-up questions, instead of open-ended ones.

Open-domain question answering In the open-domain QA setting, a system is given a question without any associated context, and must retrieve the necessary context to answer the question (Chen et al., 2017; Joshi et al., 2017; Dhingra et al., 2017; Yang et al., 2018; Seo et al., 2019; Karpukhin et al., 2020; Min et al., 2019a). **IIRC** is similar in that it

also requires the retrieval of missing information. However, the questions are grounded in a given paragraph, meaning that a system must examine more than just the question in order to know what to retrieve. Most questions in **IIRC** do not make sense in an open-domain setting, without their associated paragraphs.

Unanswerable questions Unlike SQuAD 2.0 (Rajpurkar et al., 2018) where the unanswerable questions were written to be close to answerable questions, **IIRC** contains naturally unanswerable questions that were not written with the goal of being unanswerable, a property that our dataset shares with NewsQA (Trischler et al., 2016), Natural Questions (Kwiatkowski et al., 2019), and TyDi QA (Clark et al., 2020). Results shown in Section 4.3 indicate that these questions cannot be trivially distinguished from answerable questions.

Incomplete Information QA A few prior datasets have explored question answering given incomplete information, such as science facts (Mihaylov et al., 2018; Khot et al., 2019). However, these datasets contain multiple choice questions, and the answer choices provide hints as to what information may be needed. Yuan et al. (2020) explore this as well using a POMDP in which the context in existing QA datasets is hidden from the model until it explicitly searches for it.

7 Conclusion

We introduced **IIRC**, a new dataset of incomplete-information reading comprehension questions. These questions require identifying what information is missing from a paragraph in order to answer a question, predicting where to find it, then synthesizing the retrieved information in complex ways. Our baseline model, built on top of state-of-the-art models for the most closely related existing datasets, performs quite poorly in this setting, even when given oracle retrieval results, and especially when combined with other reading comprehension datasets. **IIRC** both provides a promising new avenue for studying complex reading and retrieval problems and demonstrates that much more research is needed in this area.

References

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

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *NAACL*.

Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *ArXiv*, abs/2003.05002.

Pradeep Dasigi, Nelson F Liu, Ana Marasovic, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5927–5934.

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.

Bhuwan Dhingra, Kathryn Mazaitis, and William W. Cohen. 2017. Quasar: Datasets for question answering by search and reading. *ArXiv*, abs/1707.03904.

Dheeru Dua, Ananth Gottumukkala, Alon Talmor, Sameer Singh, and Matt Gardner. 2019a. ORB: An open reading benchmark for comprehensive evaluation of machine reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 147–153, Hong Kong, China. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019b. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL-HLT*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2013. Relational Inference for Wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2019. What's missing: A knowledge gap guided approach for multi-hop question answering. In *EMNLP*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

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

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019b. Compositional questions do not necessitate multi-hop reasoning. In *ACL*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David A. McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *EMNLP*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

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.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of EMNLP*.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *ACL*.

Min Joon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur P. Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *ACL*.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. NewsQA: A Machine Comprehension Dataset. *ArXiv*, abs/1611.09830.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Xingdi Yuan, Jie Fu, Marc-Alexandre Cote, Yi Tay, Christopher Pal, and Adam Trischler. 2020. Interactive machine comprehension with information seeking agents. In *ACL*.

Zhuosheng Zhang, Jun jie Yang, and Hai Zhao. 2020. Retrospective reader for machine reading comprehension. *ArXiv*, abs/2001.09694.