Finally, the size of SQuAD is rather small (80k training examples). It should further improve the system performance if we can collect more training examples.

To overcome these problems, we propose a procedure to automatically create additional training examples from other question answering resources. The idea is to re-use the efficient information retrieval module that we built: if we already have a question answer pair \((q, a)\) and the retrieval module can help us find a paragraph relevant to the question \(q\) and the answer segment \(a\) appears in the paragraph, then we can create a \textit{distantly-supervised} training example in the form of a \((p, q, a)\) triple for training the reading comprehension models:

\[
f : (q, a) \implies (p, q, a)
\]

if \(p \in \text{DocumentRetriever}(q)\) and \(a\) appears in \(p\)

This idea is a similar spirit to the popular approach of using distant supervision (DS) for relation extraction (Mintz et al., 2009)\(^2\). Despite that these examples can be noisy to some extent, it offers a cheap solution to create distantly supervised examples for open-domain question answering and will be a useful addition to SQuAD examples. We will describe the effectiveness of these distantly supervised examples in Section 5.3.

### 5.3 Evaluation

We have all the basic elements of our DRQA systems and let’s take a look at the evaluation.

#### 5.3.1 Question Answering Datasets

The first question is which question answering datasets we should evaluate on. As we discussed, SQuAD is one of the largest general purpose QA datasets currently available for question answering but it is very different from open-domain QA setting. We propose

\(^2\)The idea for relation extraction is to pair textual mentions which contain the two entities which is known as a relation between them in an existing knowledge base.
to train and evaluate our system on other datasets developed for open-domain QA that have been constructed in different ways. We hence adopt the following three datasets:

**TREC** This dataset is based on the benchmarks from the TREC QA tasks that have been curated by Baudiš and Šedivý (2015). We use the large version, which contains a total of 2,180 questions extracted from the datasets from TREC 1999, 2000, 2001 and 2002.\(^3\) Note that for this dataset, all the answers are written in regular expressions, for example, the answer is `Sept(ember)?|Feb(ruary)?` to the question *When is Fashion week in NYC?*, so answers *Sept, September, Feb, February* are all judged as correct.

**WebQuestions** Introduced in Berant et al. (2013), this dataset is built to answer questions from the Freebase KB. It was created by crawling questions through the Google Suggest API, and then obtaining answers using Amazon Mechanical Turk. We convert each answer to text by using entity names so that the dataset does not reference Freebase IDs and is purely made of plain text question-answer pairs.

**WikiMovies** This dataset, introduced in Miller et al. (2016), contains 96k question-answer pairs in the domain of movies. Originally created from the OMDB and MOVIE-LENS databases, the examples are built such that they can also be answered by using a subset of Wikipedia as the knowledge source (the title and the first section of articles from the movie domain).

We would like to emphasize that these datasets are not necessarily collected in the context of answering from Wikipedia. The TREC dataset was designed for text-based question answering (the primary TREC document sets consist mostly of newswire articles), while **WebQuestions** and **WikiMovies** were mainly collected for knowledge-based question answering. We put all these resources in one unified framework, and test how well our system can answer all the questions — hoping that it can reflect the performance of general-knowledge QA.

Table 5.1 and Figure 5.3 give detailed statistics of these QA datasets. As we can see that, the distribution of SQUAD examples is quite different from that of the other QA datasets.

\(^3\)This dataset is available at [https://github.com/brmson/dataset-factoid-curated](https://github.com/brmson/dataset-factoid-curated).
Due to the construction method, SQUAD has longer questions (10.4 tokens vs 6.7–7.5 tokens on average). Also, all these datasets have short answers (although the answers in SQUAD are slightly longer) and most of them are factoid.

Note that there are might be multiple answers for many of the questions in these QA datasets (see the # answers column of Table 5.1). For example, there are two valid answers: English and Urdu to the question What language do people speak in Pakistan? on WEBQUESTIONS. As our system is designed to return one answer, our evaluation considers the prediction as correct if it gives any of the gold answers.

Figure 5.3: The average length of questions and answers in our QA datasets. All the statistics are computed based on the training sets.

---

4 As all the answer strings are regex expressions, it is difficult to estimate # of answers. We only simply list the number of alternation symbols | in the answer.
### Table 5.1: Statistics of the QA datasets used for DRQA. DS Train: distantly supervised training data. †: These training sets are not used as is because no passage is associated with each question.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Train</th>
<th># DS Train</th>
<th># Test</th>
<th># answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQUAD</td>
<td>87,599</td>
<td>71,231</td>
<td>N/A</td>
<td>1.0</td>
</tr>
<tr>
<td>TREC</td>
<td>1,486†</td>
<td>3,464</td>
<td>694</td>
<td>3.2⁴</td>
</tr>
<tr>
<td>WEBQUESTIONS</td>
<td>3,778†</td>
<td>4,602</td>
<td>2,032</td>
<td>2.4</td>
</tr>
<tr>
<td>WIKIMOVIES</td>
<td>96,185†</td>
<td>36,301</td>
<td>9,952</td>
<td>1.9</td>
</tr>
</tbody>
</table>

5.3.2 Implementation Details

5.3.2.1 Processing Wikipedia

We use the 2016-12-21 dump⁵ of English Wikipedia for all of our full-scale experiments as the knowledge source used to answer questions. For each page, only the plain text is extracted and all structured data sections such as lists and figures are stripped.⁶ After discarding internal disambiguation, list, index, and outline pages, we retain 5,075,182 articles consisting of 9,008,962 unique uncased token types.

5.3.2.2 Distantly-supervised data

We use the following process for each question-answer pair from the training portion of each dataset to build our distantly-supervised training examples. First, we run our DOCUMENT RETRIEVER on the question to retrieve the top 5 Wikipedia articles. All paragraphs from those articles without an exact match of the known answer are directly discarded. All paragraphs shorter than 25 or longer than 1500 characters are also filtered out. If any named entities are detected in the question, we remove any paragraph that does not contain them at all. For every remaining paragraph in each retrieved page, we score all positions that match an answer using unigram and bigram overlap between the question and a 20 token window, keeping up to the top 5 paragraphs with the highest overlaps. If there is no paragraph with non-zero overlap, the example is discarded; otherwise we add each found

---

⁵[https://dumps.wikimedia.org/enwiki/latest](https://dumps.wikimedia.org/enwiki/latest)

⁶[We use the WikiExtractor script: https://github.com/attardi/wikiextractor](https://github.com/attardi/wikiextractor).
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Example</th>
<th>Article / Paragraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC</td>
<td><strong>Q:</strong> What U.S. state’s motto is “Live free or Die”?&lt;br&gt;A: New Hampshire</td>
<td><strong>Article:</strong> Live Free or Die&lt;br&gt;<strong>Paragraph:</strong> “Live Free or Die” is the official motto of the U.S. state of <strong>New Hampshire</strong>, adopted by the state in 1945. It is possibly the best-known of all state mottos, partly because it conveys an assertive independence historically found in American political philosophy and partly because of its contrast to the milder sentiments found in other state mottos.</td>
</tr>
<tr>
<td>WEBQUESTIONS</td>
<td><strong>Q:</strong> What part of the atom did Chadwick discover?&lt;br&gt;A: neutron</td>
<td><strong>Article:</strong> Atom&lt;br&gt;<strong>Paragraph:</strong> ... The atomic mass of these isotopes varied by integer amounts, called the whole number rule. The explanation for these different isotopes awaited the discovery of the neutron, an uncharged particle with a mass similar to the proton, by the physicist James Chadwick in 1932. ...</td>
</tr>
<tr>
<td>WIKIMOVIES</td>
<td><strong>Q:</strong> Who wrote the film Gigli?&lt;br&gt;A: Martin Brest</td>
<td><strong>Article:</strong> Gigli&lt;br&gt;<strong>Paragraph:</strong> Gigli is a 2003 American romantic comedy film written and directed by Martin Brest and starring Ben Affleck, Jennifer Lopez, Justin Bartha, Al Pacino, Christopher Walken, and Lainie Kazan.</td>
</tr>
</tbody>
</table>

Figure 5.4: Example training data from each QA dataset. In each case we show an associated paragraph where distant supervision (DS) correctly identified the answer within it, which is highlighted.

pair to our DS training dataset. Some examples are shown in Figure 5.4 and the number of distantly supervised examples we created for training are given in Table 5.1 (column # DS Train).

### 5.3.3 Document Retriever Performance

We first examine the performance of our retrieval module on all the QA datasets. Table 5.2 compares the performance of the two approaches described in Section 5.2.2 with that of
Table 5.2: Document retrieval results. % of questions for which the answer segment appears in one of the top 5 pages returned by the method.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Wiki. Search</th>
<th>Document Retriever</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>unigram</td>
<td>bigram</td>
</tr>
<tr>
<td>TREC</td>
<td>81.0</td>
<td>85.2</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>73.7</td>
<td>75.5</td>
</tr>
<tr>
<td>WikiMovies</td>
<td>61.7</td>
<td>54.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70.3</td>
</tr>
</tbody>
</table>

Finally, we assess the performance of our full system DrQA for answering open-domain questions using all these datasets. We compare three versions of DrQA which evaluate the impact of using distant supervision and multitask learning across the training sources provided to Document Reader (Document Retriever remains the same for each case):

- **SQUAD**: A single Document Reader model is trained on the SQUAD training set only and used on all evaluation sets. We used the model that we described in Section 5.2 (the F1 score is 79.0% on the test set of SQUAD).

- **Fine-tune (DS)**: A Document Reader model is pre-trained on SQUAD and then fine-tuned for each dataset independently using its distant supervision (DS) training.

---

**Table 5.3**: Full Wikipedia results. Top-1 exact-match accuracy (%). **FT**: Fine-tune (DS). **MT**: Multitask (DS). The DrQA* results are taken from Raison et al. (2018).

- **Multitask (DS)**: A single DOCUMENT READER model is jointly trained on the SQuAD training set and all the distantly-supervised examples.

For the full Wikipedia setting we use a streamlined model that does not use the CORENLP parsed $f_{token}$ features or lemmas for $f_{exact\_match}$. We find that while these help for more exact paragraph reading in SQuAD, they don’t improve results in the full setting. Additionally, WEBQUESTIONS and WIKIMOVIES provide a list of candidate answers (1.6 million FREEBASE entity strings for WEBQUESTIONS and 76k movie-related entities for WIKIMOVIES) and we restrict that the answer span must be in these lists during prediction.

Table 5.3 presents the results. We only consider top-1, exact-match accuracy, which is the most restricted and challenging setting. In the original paper (Chen et al., 2017), we also evaluated the question/answer pairs in SQuAD. We omit them here because that at least a third of these questions are context-dependent and are not really suitable for open QA.

Despite the difficulty of the task compared to the reading comprehension task (where you are given the right paragraph) and unconstrained QA (using redundant resources), DRQA still provides reasonable performance across all four datasets.

We are interested in a single, full system that can answer any question using Wikipedia. The single model trained only on SQUAD is outperformed on all the datasets by the multitask model that uses distant supervision. However, performance when training on SQUAD
alone is not far behind, indicating that task transfer is occurring. The majority of the improvement from SQUAD to Multitask (DS) learning, however, is likely not from task transfer, as fine-tuning on each dataset alone using DS also gives improvements, showing that is the introduction of extra data in the same domain that helps. Nevertheless, the best single model that we can find is our overall goal, and that is the Multitask (DS) system.

We compare our system to YODAQ (Baudiš, 2015) (an unconstrained QA system using redundant resources), giving results which were previously reported on TREC and WEBQUESTIONS.\(^8\) Despite the increased difficulty of our task, it is reassuring that our performance is not too far behind on TREC (31.3 vs 25.4). The gap is slightly bigger on WEBQUESTIONS, likely because this dataset was created from the specific structure of FREEBASE which YODAQ uses directly.

We also include the results from an enhancement of our model named DRQA*, presented in Raison et al. (2018). The biggest change is that this reading comprehension model is trained and evaluated directly on the Wikipedia articles instead of paragraphs (documents are on average 40 times larger than individual paragraphs). As we can see, the performance has been improved consistently on all the datasets, and the gap from YODAQ is hence further reduced.

\(^8\)The results are extracted from https://github.com/brmson/yodaqa/wiki/Benchmarks.
## CHAPTER 5. OPEN DOMAIN QUESTION ANSWERING

### (a) Question
What is question answering?

**Answer**
a computer science discipline within the fields of information retrieval and natural language processing

**Wiki. article** [Question Answering](https://en.wikipedia.org/wiki/Question_answering)

**Passage**
Question Answering (QA) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

### (b) Question
Which state is Stanford University located in?

**Answer**
California

**Wiki. article** [Stanford Memorial Church](https://en.wikipedia.org/wiki/Stanford_Memorial_Church)

**Passage**
Stanford Memorial Church (also referred to informally as MemChu) is located on the Main Quad at the center of the Stanford University campus in Stanford, California, United States. It was built during the American Renaissance by Jane Stanford as a memorial to her husband Leland. Designed by architect Charles A. Coolidge, a protégé of Henry Hobson Richardson, the church has been called "the University’s architectural crown jewel".

### (c) Question
Who invented LSTM?

**Answer**
Sepp Hochreiter & Jürgen Schmidhuber

**Wiki. article** [Deep Learning](https://en.wikipedia.org/wiki/Deep_learning)

**Passage**
Today, however, many aspects of speech recognition have been taken over by a deep learning method called Long short-term memory (LSTM), a recurrent neural network published by Sepp Hochreiter & Jürgen Schmidhuber in 1997. LSTM RNNs avoid the vanishing gradient problem and can learn “Very Deep Learning” tasks that require memories of events that happened thousands of discrete time steps ago, which is important for speech. In 2003, LSTM started
to become competitive with traditional speech recognizers on certain tasks. Later it was combined with CTC in stacks of LSTM RNNs. In 2015, Google’s speech recognition reportedly experienced a dramatic performance jump of 49% through CTC-trained LSTM, which is now available through Google Voice to all smartphone users, and has become a show case of deep learning.

(d) **Question**

What is the answer to life, the universe, and everything?

**Answer**

42

**Wiki. article**

Phrases from *The Hitchhiker’s Guide to the Galaxy*

**Passage**

The number 42 and the phrase, "Life, the universe, and everything" have attained cult status on the Internet. "Life, the universe, and everything" is a common name for the off-topic section of an Internet forum and the phrase is invoked in similar ways to mean "anything at all". Many chatbots, when asked about the meaning of life, will answer "42". Several online calculators are also programmed with the Question. Google Calculator will give the result to "the answer to life the universe and everything" as 42, as will Wolfram’s Computational Knowledge Engine. Similarly, DuckDuckGo also gives the result of "the answer to the ultimate question of life, the universe and everything” as 42.

In the online community Second Life, there is a section on a sim called "42nd Life.” It is devoted to this concept in the book series, and several attempts at recreating Milliways, the Restaurant at the End of the Universe, were made.

Table 5.4: Sample predictions of our DrQA system.

Lastly, our DrQA system is open-sourced at [https://github.com/facebookresearch/DrQA](https://github.com/facebookresearch/DrQA) (the Multitask (DS) system was deployed). Table 5.4 lists some sample predictions that we tried by ourselves (not in any of these datasets). As is seen, our system is able to return a precise answer to all these factoid questions and answering some of these questions is not trivial:

(a) It is not trivial to identify that *a computer science discipline within the fields of information retrieval and natural language processing* is the complete noun phrase and the correct answer although the question is pretty simple.
(b) Our system finds the answer in another Wikipedia article *Stanford Memorial Church*, and gives the exactly correct answer *California* as the *state* (instead of *Stanford* or *United States*).

(c) To get the correct answer, the system needs to understand the syntactic structure of the question and the context *Who invented LSTM?* and *a deep learning method called Long short-term memory (LSTM), a recurrent neural network published by Sepp Hochreiter & Jürgen Schmidhuber in 1997*.

Conceptually, our system is simple and elegant, and doesn’t rely on any additional linguistic analysis or external or hand-coded resources (e.g., dictionaries). We think this approach holds great promise for a new generation of open-domain question answering systems. In the next section, we discuss current limitations and possible directions for further improvement.

### 5.4 Future Work

Our DRQA demonstrates that combining information retrieval and neural reading comprehension is an effective approach for open-domain question answering. We hope that our work takes the first step in this research direction. However, our system is still at an early stage and many implementation details can be further improved.

We think the following research directions will (greatly) improve our DRQA system and should be pursued as future work. Indeed, some of the ideas have already been implemented in the following year after we published our DRQA system and we will also describe them in detail in this section.

**Aggregating evidence from multiple paragraphs.** Our system adopted the most simple and straightforward approach: we took the argmax over the unnormalized scores of all the retrieved passages. This is not ideal because 1) It implies that each passage must contain the correct answer (as SQuAD examples) so our system will output one and only one answer for each passage. This is indeed not the case for most retrieved passages. 2) Our current
training paradigm doesn’t guarantee that the scores in different passages are comparable which causes a gap between the training and the evaluation process.

Training on full Wikipedia articles is a solution to alleviate this problem (see the DRQA* results in Table 5.3), however, these models are running slowly and difficult to parallelize. Clark and Gardner (2018) proposed to perform multi-paragraph training with modified training objectives, where the span start and end scores are normalized across all paragraphs sampled from the same context. They demonstrated that it works much better than training on individual passages independently. Similarly, Wang et al. (2018a) and Wang et al. (2018b) proposed to train an explicit passage re-ranking component on the retrieved articles: Wang et al. (2018a) implemented this in a reinforcement learning framework so the re-ranker component and answer extraction components are jointly trained; Wang et al. (2018b) proposed a strength-based re-ranker and a coverage-based re-ranker which aggregate evidence from multiple paragraphs more directly.

**Using more and better training data.** The second aspect which makes a big impact is the training data. Our DRQA system only collected 44k distantly-supervised training examples from TREC, WebQuestions and WikiMovies, and we demonstrated their effectiveness in Section 5.3.4. The system should be further improved if we can leverage more supervised training data — from either TRIVIAQA (Joshi et al., 2017) or generating more data from other QA resources. Moreover, these distantly supervised examples inevitably suffer from the noise problem (i.e., the paragraph doesn’t imply the answer to the question even if the answer is contained) and Lin et al. (2018) proposed a solution to de-noise these distantly supervised examples and demonstrated gains in an evaluation.

We also believe that adding negative examples should improve the performance of our system substantially. We can either create some negative examples using our full pipeline: we can leverage the DOCUMENT RETRIEVAL module to help us find relevant paragraphs while they don’t contain the correct answer. We can also incorporate existing resources such as SQuAD 2.0 (Rajpurkar et al., 2018) into our training process, which contains curated, high-quality negative examples.
Making the **DOCUMENT RETRIEVER** trainable. A third promising direction that has not been fully studied yet is to employ a machine learning approach for the **DOCUMENT RETRIEVER** module. Our system adopted a straightforward, non-machine learning model and further improvement on the retrieval performance (Table 5.2) should lead to an improvement on the full system. A training corpus for the **DOCUMENT RETRIEVER** component can be collected either from other resources or from the QA data (e.g., using whether an article contains the answer to the question as a label). Joint training of the **DOCUMENT RETRIEVAL** and the **DOCUMENT READER** component will be a very desirable and promising direction for future work.

Related to this, Clark and Gardner (2018) also built an open-domain question answering system on top of a search engine (Bing web search) and demonstrated superior performance compared to ours. We think the results are not directly comparable and the two approaches (using a commercial search engine or building an independent IR component) both have pros and cons. Building our own IR component gets rid of an existing API call and can run faster and easily adapt to new domains.

**Better DOCUMENT READER module.** For our DrQA system, we used the neural reading comprehension model which achieved F1 of 79.0% on the test set of SQUAD 1.1. With the recent development of neural reading comprehension models (Section 3.4), we are confident that if we replace our current **DOCUMENT READER** model with the state-of-the-art models (Devlin et al., 2018), the performance of our full system will be improved as well.

**More analysis is needed.** Another important missing work is to conduct an in-depth analysis of our current systems: to understand which questions they can answer, and which they can’t. We think it is important to compare our modern systems to the earlier TREC QA results under the same conditions. It will help us understand where we make genuine progress and what techniques we can still use from the pre-deep learning era, to build better question answering systems in the future.

Concurrent to our work, there are several works in a similar spirit to ours, including **\(^9\)**The demo is at [https://documentqa.allenai.org](https://documentqa.allenai.org).
SEARCHQA (Dunn et al., 2017) and QUASAR-T (Dhingra et al., 2017b), which both collected relevant documents for trivia or JEOPARDY! questions — the former one retrieved documents from CLUEWEB using the LUCENE index and the latter used GOOGLE search. TRIVIAQA (Joshi et al., 2017) also has an open-domain setting where all the retrieved documents from Bing web search are kept. However, these datasets still focus on the task of question answering from the retrieved documents, while we are more interested in building an end-to-end QA system.
Chapter 6

Conversational Question Answering

In the last chapter, we discussed how we built a general-knowledge question-answering system from neural reading comprehension. However, most current QA systems are limited to answering isolated questions, i.e., every time we ask a question, the systems return an answer without the ability to consider any context. In this chapter, we set out to tackle another challenging problem, Conversational Question Answering, where a machine has to understand a text passage and answer a series of questions that appear in a conversation.

Humans gather information by engaging in conversations involving a series of interconnected questions and answers. For machines to assist in information gathering, it is therefore essential to enable them to answer conversational questions. Figure 6.1 shows a conversation between two humans who are reading a passage, one acting as a questioner and the other as an answerer. In this conversation, every question after the first is dependent on the conversation history. For instance, $Q_5$ Who? is only a single word and is impossible to answer without knowing what has already been said. Posing short questions is an effective human conversation strategy, but such questions are really difficult for machines to parse. Therefore, conversational question answering combines the challenges from both dialogue and reading comprehension.

We believe that building systems which are able to answer such conversational questions will play a crucial role in our future conversational AI systems. To approach this problem, we need to build effective datasets and conversational QA models and we will describe both of them in this chapter.
Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80. Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie’s husband Josh were coming as well. Jessica had . . .

Q1: Who had a birthday?
A1: Jessica
R1: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.

Q2: How old would she be?
A2: 80
R2: she was turning 80

Q3: Did she plan to have any visitors?
A3: Yes
R3: Her granddaughter Annie was coming over

Q4: How many?
A4: Three
R4: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie’s husband Josh were coming as well.

Q5: Who?
A5: Annie, Melanie and Josh
R5: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie’s husband Josh were coming as well.

Figure 6.1: A conversation from our CoQA dataset. Each turn contains a question ($Q_i$), an answer ($A_i$) and a rationale ($R_i$) that supports the answer.

This chapter is organized as follows. We first discuss related work in Section 6.1 and then we introduce CoQA (Reddy et al., 2019) in Section 6.2, a Conversational Question Answering challenge for measuring the ability of machines to participate in a question-answering style conversation.¹ Our dataset contains 127k questions with answers, obtained from 8k conversations about text passages from seven diverse domains. We define our task

¹We launch CoQA as a challenge to the community at https://stanfordnlp.github.io/coqa/.
and describe the dataset collection process. We also analyze the dataset in depth and show that conversational questions have challenging phenomena not present in existing reading comprehension datasets, e.g., coreference and pragmatic reasoning. Next we describe several strong conversational and reading comprehension models we built for COQA in Section 6.3 and present experimental results in Section 6.4. Finally, we discuss future work of conversational question answering (Section 6.5).

6.1 Related Work

Conversational question answering is directly related to dialogue. Building conversational agents, or dialogue systems to converse with humans in natural language is one of the major goals of natural language understanding. The two most common classes of dialogue systems are: task-oriented, and chit-chat (or chatbot) dialogue agents. Task-oriented dialogue systems are designed for a particular task and set up to have short conversations (e.g., booking a flight or making a restaurant reservation). They are evaluated based on task-completion rate or time to task completion. In contrast, chit-chat dialogue systems are designed for extended, casual conversations, without a specific goal. Usually, the longer the user engagement and interaction, the better these systems are.

Answering questions is also a core task of dialogue systems, because one of the most common needs for humans to interact with dialogue agents is to seek information and ask questions of various topics. QA-based dialogue techniques have been developed extensively in automated personal assistant systems such as Amazon’s ALEXA, Apple’s SIRI or GOOGLE ASSISTANT, either based on structured knowledge bases, or unstructured text collections. Modern dialogue systems are mostly built on top of deep neural networks. For a comprehensive survey of neural approaches to different types of dialogue systems, we refer readers to (Gao et al., 2018).

Our work is closely related to the Visual Dialog task of (Das et al., 2017) and the Complex Sequential Question Answering task of (Saha et al., 2018), which perform conversational question answering on images and a knowledge graph (e.g. WIKIDATA) respectively, with the latter focusing on questions obtained by paraphrasing templates. Figure 6.2 demonstrates an example from each task. We focus on conversations over a passage of text,
which requires the ability of reading comprehension.

Another related line of research is sequential question answering (Iyyer et al., 2017; Talmor and Berant, 2018), in which a complex question is decomposed into a sequence of simpler questions. For example, a question What super hero from Earth appeared most recently? can be decomposed into the following three questions: 1) Who are all of the super heroes?, 2) Which of them come from Earth?, and 3) Of those, who appeared most recently?. Therefore, their focus is how to answer a complex question via sequential question answering, while we are more interested in a natural conversation of a variety of topics while the questions can be dependent on the dialogue history.

### 6.2 CoQA: A Conversational QA Challenge

In this section, we introduce CoQA, a novel dataset for building Conversational Question Answering systems. We develop CoQA with three main goals in mind. The first concerns the nature of questions in a human conversation. As an example seen in Figure 6.1, in this conversation, every question after the first is dependent on the conversation history. At present, there are no large scale reading comprehension datasets which contain questions
that depend on a conversation history and this is what COQA is mainly developed for.\textsuperscript{2}

The second goal of COQA is to ensure the naturalness of answers in a conversation. As we discussed in the earlier chapters, most existing reading comprehension datasets either restrict answers to a contiguous span in a given passage, or allow free-form answers with a low human agreement (e.g., NARRATIVEQA). Our desiderata are 1) the answers should not be only span-based so that anything can be asked and the conversation can flow naturally. For example, there is no extractive answer for \textit{Q4 How many?} in Figure 6.1. 2) It still supports reliable automatic evaluation with a strong human performance. Therefore, we propose that the answers can be free-form text (abstractive answers), while the extractive spans act as rationales for the actual answers. Therefore, the answer for \textit{Q4} is simply \textit{Three} while its rationale is spanned across multiple sentences.

The third goal of COQA is to enable building QA systems that perform robustly across domains. The current reading comprehension datasets mainly focus on a single domain which makes it hard to test the generalization ability of existing models. Hence we collect our dataset from seven different domains — children’s stories, literature, middle and high school English exams, news, Wikipedia, science articles and Reddit. The last two are used for out-of-domain evaluation.

\subsection{6.2.1 Task Definition}

We first define the task formally. Given a passage $P$, a conversation consists of $n$ turns, and each turn consists of $(Q_i, A_i, R_i)$, $i = 1, \ldots, n$, where $Q_i$ and $A_i$ denote the question and the answer in the $i$-th turn, and $R_i$ is the rationale which supports the answer $A_i$ and must be a single span of the passage. The task is defined as to answer the next question $Q_i$ given the conversation so far: $Q_1, A_1, \ldots, Q_{i-1}, A_{i-1}$. It is worth noting that we collect $R_i$ with the hope that they can help understand how answers are derived and improve training our models, while they are not provided during evaluation.

For the example in Figure 6.3, the conversation begins with question $Q_1$. We answer $Q_1$ with $A_1$ based on the evidence $R_1$ from the passage. In this example, the answerer

\textsuperscript{2}Concurrent with our work, Choi et al. (2018) also created a conversational dataset with a similar goal, but it differs in many key design decisions. We will discuss it in Section 6.5.
The Virginia governor’s race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn’t trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Figure 6.3: A conversation showing coreference chains in colors. The entity of focus changes in Q₄, Q₅, Q₆.

wrote only the Governor as the answer but selected a longer rationale The Virginia governor’s race. When we come to Q₂ Where?, we must refer back to the conversation history since otherwise its answer could be Virginia or Richmond or something else. In our task, conversation history is indispensable for answering many questions. We use conversation history Q₁ and A₁ to answer Q₂ with A₂ based on the evidence R₂. For an unanswerable
question, we give unknown as the final answer and do not highlight any rationale.

In this example, we observe that the entity of focus changes as the conversation progresses. The questioner uses his to refer to Terry in Q4 and he to Ken in Q5. If these are not resolved correctly, we end up with incorrect answers. The conversational nature of questions requires us to reason from multiple sentences (the current question and the previous questions or answers, and sentences from the passage). It is common that a single question may require a rationale spanned across multiple sentences (e.g., Q1 Q4 and Q5 in Figure 6.1). We describe additional question and answer types in 6.2.3.

### 6.2.2 Dataset Collection

We detail our dataset collection process as follows. For each conversation, we employ two annotators, a questioner and an answerer. This setup has several advantages over using a single annotator to act both as a questioner and an answerer: 1) when two annotators chat about a passage, their dialogue flow is natural compared to chatting with oneself; 2) when one annotator responds with a vague question or an incorrect answer, the other can raise a flag which we use to identify bad workers; and 3) the two annotators can discuss guidelines (through a separate chat window) when they have disagreements. These measures help to prevent spam and to obtain high agreement data.³

We use the Amazon Mechanical Turk (AMT) to pair workers on a passage for which we use the ParlAI Mturk API (Miller et al., 2017). On average, each passage costs 3.6 USD for conversation collection and another 4.5 USD for collecting three additional answers for development and test data.

**Collection interface.** We have different interfaces for a questioner and an answerer (Figure 6.4 and Figure 6.5). A questioner’s role is to ask questions, and an answerer’s role is to answer questions in addition to highlighting rationales. We want questioners to avoid using exact words in the passage in order to increase lexical diversity. When they type a word that is already present in the passage, we alert them to paraphrase the question if possible. For the answers, we want answerers to stick to the vocabulary in the passage in

---

³Due to AMT terms of service, we allowed a single worker to act both as a questioner and an answerer after a minute of waiting. This constitutes around 12% of the data.
order to limit the number of possible answers. We encourage this by automatically copying the highlighted text into the answer box and allowing them to edit copied text in order to generate a natural answer. We found 78% of the answers have at least one edit such as changing a word’s case or adding a punctuation.
### Table 6.1: Distribution of domains in CoQA.

<table>
<thead>
<tr>
<th>Domain</th>
<th># Passages</th>
<th># Q/A pairs</th>
<th>Passage length</th>
<th># Turns per passage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Children’s Stories</td>
<td>750</td>
<td>10.5k</td>
<td>211</td>
<td>14.0</td>
</tr>
<tr>
<td>Literature</td>
<td>1,815</td>
<td>25.5k</td>
<td>284</td>
<td>15.6</td>
</tr>
<tr>
<td>Mid/High School Exams</td>
<td>1,911</td>
<td>28.6k</td>
<td>306</td>
<td>15.0</td>
</tr>
<tr>
<td>News</td>
<td>1,902</td>
<td>28.7k</td>
<td>268</td>
<td>15.1</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>1,821</td>
<td>28.0k</td>
<td>245</td>
<td>15.4</td>
</tr>
<tr>
<td>Science</td>
<td>100</td>
<td>1.5k</td>
<td>251</td>
<td>15.3</td>
</tr>
<tr>
<td>Reddit</td>
<td>100</td>
<td>1.7k</td>
<td>361</td>
<td>16.6</td>
</tr>
<tr>
<td>Total</td>
<td>8,399</td>
<td>127k</td>
<td>271</td>
<td>15.2</td>
</tr>
</tbody>
</table>

**Passage selection.** We select passages from seven diverse domains: children’s stories from MCTest (Richardson et al., 2013), literature from Project Gutenberg⁴, middle and high school English exams from RACE (Lai et al., 2017), news articles from CNN (Hermann et al., 2015), articles from Wikipedia, science articles from AI2 Science Questions (Welbl et al., 2017) and Reddit articles from the Writing Prompts dataset (Fan et al., 2018).

Not all passages in these domains are equally good for generating interesting conversations. A passage with just one entity often result in questions that entirely focus on that entity. We select passages with multiple entities, events and pronominal references using Stanford CORENLP (Manning et al., 2014). We truncate long articles to the first few paragraphs that result in around 200 words.

Table 6.1 shows the distribution of domains. We reserve the Science and Reddit domains for out-of-domain evaluation. For each in-domain dataset, we split the data such that there are 100 passages in the development set, 100 passages in the test set, and the rest in the training set. In contrast, for each out-of-domain dataset, we just have 100 passages in the test set without any passages in the training or the development sets.

**Collecting multiple answers.** Some questions in CoQA may have multiple valid answers. For example, another answer for Q₁ in Figure 6.3 is *A Republican candidate.*

---

⁴Project Gutenberg [https://www.gutenberg.org](https://www.gutenberg.org)
order to account for answer variations, we collect three additional answers for all questions in the development and test data. Since our data is conversational, questions influence answers which in turn influence the follow-up questions. In the previous example, if the original answer was *A Republican Candidate*, then the following question *Which party does he belong to?* would not have occurred in the first place. When we show questions from an existing conversation to new answerers, it is likely they will deviate from the original answers which makes the conversation incoherent. It is thus important to bring them to a common ground with the original answer.

We achieve this by turning the answer collection task into a game of predicting original answers. First, we show a question to a new answerer, and when she answers it, we show the original answer and ask her to verify if her answer matches the original. For the next question, we ask her to guess the original answer and verify again. We repeat this process until the conversation is complete. In our pilot experiment, the human F1 score increased by 5.4% when we use this verification setup.

### 6.2.3 Dataset Analysis

What makes the CoQA dataset conversational compared to existing reading comprehension datasets like SQUAD? How does the conversation flow from one turn to the other? What linguistic phenomena do the questions in CoQA exhibit? We answer these questions below.

**Comparison with SQUAD 2.0.** In the following, we perform an in-depth comparison of CoQA and SQUAD 2.0 (Rajpurkar et al., 2018). Figure 6.6 shows the distribution of frequent trigram prefixes. While coreferences are non-existent in SQUAD 2.0, almost every sector of CoQA contains coreferences (*he, him, she, it, they*) indicating CoQA is highly conversational. Because of the free-form nature of answers, we expect a richer variety of questions in CoQA than SQUAD 2.0. While nearly half of SQUAD 2.0 questions are dominated by *what* questions, the distribution of CoQA is spread across multiple question types. Several sectors indicated by prefixes *did, was, is, does, and* are frequent in CoQA but are completely absent in SQUAD 2.0.
FIGURE 6.6: Distribution of trigram prefixes of questions in SQuAD 2.0 and CoQA.

Since a conversation is spread over multiple turns, we expect conversational questions and answers to be shorter than in a standalone interaction. In fact, questions in CoQA can be made up of just one or two words (who?, when?, why?). As seen in Table 6.2, on average, a question in CoQA is only 5.5 words long while it is 10.1 for SQuAD. The answers are also usually shorter in CoQA than SQuAD 2.0.

Table 6.3 provides insights into the type of answers in SQUAD 2.0 and CoQA. While the original version of SQUAD 2.0 (Rajpurkar et al., 2016) does not have any unanswerable questions, SQUAD 2.0 (Rajpurkar et al., 2018) focuses solely on obtaining them resulting in higher frequency than in CoQA. SQUAD 2.0 has 100% extractive answers by design, whereas in CoQA, 66.8% answers can be classified as extractive after ignoring punctuation and case mismatches.\(^5\) This is higher than we anticipated. Our conjecture is that human factors such as wage may have influenced workers to ask questions that elicit faster responses by selecting text. It is worth noting that CoQA has 11.1% and 8.7% questions with yes or no as answers whereas SQUAD 2.0 has 0%. Both datasets have a high number of named entities and noun phrases as answers.

\(^5\)If punctuation and case are not ignored, only 37% of the answers are extractive.
Table 6.2: Average number of words in passage, question and answer in SQUAD 2.0 and CoQA.

<table>
<thead>
<tr>
<th></th>
<th>SQUAD 2.0</th>
<th>CoQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passage Length</td>
<td>117</td>
<td>271</td>
</tr>
<tr>
<td>Question Length</td>
<td>10.1</td>
<td>5.5</td>
</tr>
<tr>
<td>Answer Length</td>
<td>3.2</td>
<td>2.7</td>
</tr>
</tbody>
</table>

Table 6.3: Distribution of answer types in SQUAD 2.0 and CoQA.

<table>
<thead>
<tr>
<th></th>
<th>SQUAD 2.0</th>
<th>CoQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answerable</td>
<td>66.7%</td>
<td>98.7%</td>
</tr>
<tr>
<td>Unanswerable</td>
<td>33.3%</td>
<td>1.3%</td>
</tr>
<tr>
<td>Extractive</td>
<td>100.0%</td>
<td>66.8%</td>
</tr>
<tr>
<td>Abstractive</td>
<td>0.0%</td>
<td>33.2%</td>
</tr>
<tr>
<td>Named Entity</td>
<td>35.9%</td>
<td>28.7%</td>
</tr>
<tr>
<td>Noun Phrase</td>
<td>25.0%</td>
<td>19.6%</td>
</tr>
<tr>
<td>Yes</td>
<td>0.0%</td>
<td>11.1%</td>
</tr>
<tr>
<td>No</td>
<td>0.1%</td>
<td>8.7%</td>
</tr>
<tr>
<td>Number</td>
<td>16.5%</td>
<td>9.8%</td>
</tr>
<tr>
<td>Date/Time</td>
<td>7.1%</td>
<td>3.9%</td>
</tr>
<tr>
<td>Other</td>
<td>15.5%</td>
<td>18.1%</td>
</tr>
</tbody>
</table>

**Conversation flow.** A coherent conversation must have smooth transitions between turns. We expect the narrative structure of the passage to influence our conversation flow. We split the passage into 10 uniform chunks, and identify chunks of interest of a given turn and its transition based on rationale spans.

Figure 6.7 portrays the conversation flow of the top 10 turns. The starting turns tend to focus on the first few chunks and as the conversation advances, the focus shifts to the later chunks. Moreover, the turn transitions are smooth, with the focus often remaining in the same chunk or moving to a neighbouring chunk. Most frequent transitions happen to the first and the last chunks, and likewise these chunks have diverse outward transitions.

**Linguistic phenomena.** We further analyze the questions for their relationship with the passages and the conversation history. We sample 150 questions in the development set
Figure 6.7: Chunks of interests as a conversation progresses. The x-axis indicates the turn number and the y-axis indicates the passage chunk containing the rationale. The height of a chunk indicates the concentration of conversation in that chunk. The width of the bands is proportional to the frequency of transition between chunks from one turn to the other.

and annotate various phenomena as shown in Table 6.4.

If a question contains at least one content word that appears in the passage, we classify it as lexical match. These comprise around 29.8% of the questions. If it has no lexical match but is a paraphrase of the rationale, we classify it as paraphrasing. These questions contain phenomena such as synonymy, antonymy, hypernymy, hyponymy and negation. These constitute a large portion of questions, around 43.0%. The rest, 27.2%, have no lexical cues, and we classify them under pragmatics. These include phenomena like common sense and presupposition. For example, the question Was he loud and boisterous? is not a direct paraphrase of the rationale he dropped his feet with the lithe softness of a cat but the rationale combined with world knowledge can answer this question.
<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Example</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relationship between a question and its passage</td>
<td></td>
</tr>
<tr>
<td>Lexical match</td>
<td>Q: Who had to rescue her?</td>
<td>29.8%</td>
</tr>
<tr>
<td></td>
<td>A: the coast guard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R: Outen was rescued by the coast guard</td>
<td></td>
</tr>
<tr>
<td>Paraphrasing</td>
<td>Q: Did the wild dog approach?</td>
<td>43.0%</td>
</tr>
<tr>
<td></td>
<td>A: Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R: he drew cautiously closer</td>
<td></td>
</tr>
<tr>
<td>Pragmatics</td>
<td>Q: Is Joey a male or female?</td>
<td>27.2%</td>
</tr>
<tr>
<td></td>
<td>A: Male</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R: it looked like a stick man so she kept him.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>She named her new noodle friend Joey</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relationship between a question and its conversation history</td>
<td></td>
</tr>
<tr>
<td>No coreference</td>
<td>Q: What is IFL?</td>
<td>30.5%</td>
</tr>
<tr>
<td>Explicit coreference</td>
<td>Q: Who had Bashti forgotten?</td>
<td>49.7%</td>
</tr>
<tr>
<td></td>
<td>A: the puppy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q: What was his name?</td>
<td></td>
</tr>
<tr>
<td>Implicit coreference</td>
<td>Q: When will Sirisena be sworn in?</td>
<td>19.8%</td>
</tr>
<tr>
<td></td>
<td>A: 6 p.m local time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q: Where?</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.4: Linguistic phenomena in CoQA questions.

For the relationship between a question and its conversation history, we classify questions into whether they are dependent or independent on the conversation history. If dependent, whether the questions contain an explicit marker or not.

As a result, around 30.5% questions do not rely on coreference with the conversational history and are answerable on their own. Almost half of the questions (49.7%) contain explicit coreference markers such as *he, she, it*. These either refer to an entity or an event introduced in the conversation. The remaining 19.8% do not have explicit coreference markers but refer to an entity or event implicitly.
6.3 Models

Given a passage \( p \), the conversation history \( \{q_1, a_1, \ldots q_{i-1}, a_{i-1}\} \) and a question \( q_i \), the task is to predict the answer \( a_i \). Our task can be modeled as either a conversational response generation problem or a reading comprehension problem. We evaluate strong baselines from each class of models and a combination of the two on COQA.

6.3.1 Conversational Models

The basic goal of conversational models is to predict the next utterance based on its conversation history. Sequence-to-sequence (seq2seq) models (Sutskever et al., 2014) have shown promising results for generating conversational responses (Vinyals and Le, 2015; Li...
et al., 2016; Zhang et al., 2018). Motivated by their success, we use a standard sequence-to-sequence model with an attention mechanism for generating answers. We append the passage, the conversation history (the question/answer pairs in the last \( n \) turns) and the current question as, \( \langle \text{q}\rangle q_{i-n} \langle \text{a}\rangle a_{i-n} \ldots \langle \text{q}\rangle q_{i-1} \langle \text{a}\rangle a_{i-1} \langle \text{q}\rangle q_{i} \), and feed it into a bidirectional LSTM encoder, where \( \langle \text{q}\rangle \) and \( \langle \text{a}\rangle \) are special tokens used as delimiters. We then generate the answer using a LSTM decoder which attends to the encoder states.

Moreover, as the answer words are likely to appear in the original passage, we adopt a copy mechanism in the decoder proposed for summarization tasks (Gu et al., 2016; See et al., 2017), which allows to (optionally) copy a word from the passage and the conversation history. We call this model the Pointer-Generator network (See et al., 2017), PGN. Figure 6.8 illustrates a full model of PGN. Formally, we denote the encoder hidden vectors by \( \{\tilde{h}_i\} \), the decoder state at timestep \( t \) by \( h_t \) and the input vector by \( x_t \), an attention function is computed based on \( \{\tilde{h}_i\} \) and \( h_t \) as \( \alpha_i \) (Equation 3.13) and the context vector is computed as \( c = \sum_i \alpha_i \tilde{h}_i \) (Equation 3.14).

For a copy mechanism, it first computes the generation probability \( p_{\text{gen}} \in [0, 1] \) which controls the probability that it generates a word from the full vocabulary \( V \) (rather than copying a word) as:

\[
p_{\text{gen}} = \sigma \left( w^{(c)} c + w^{(x)} x_t + w^{(h)} h_t + b \right).
\] (6.1)

The final probability distribution of generating word \( w \) is computed as:

\[
P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i: w_i = w} \alpha_i,
\] (6.2)

where \( P_{\text{vocab}}(w) \) is the original probability distribution (computed based on \( c \) and \( h_t \)) and \( \{w_i\} \) refers to all the words in the passage and the dialogue history. For more details, we refer readers to (See et al., 2017).

### 6.3.2 Reading Comprehension Models

The second class of models we evaluate is the neural reading comprehension models. In particular, the models for the span prediction problems can’t be applied directly, as a large
portion of the COQA questions don’t have a single span in the passage as their answer, e.g.,
$Q_3$, $Q_4$ and $Q_5$ in Figure 6.1. Therefore, we modified the STANFORD ATTENTIVE READER
model we described in Section 3.2 for this problem. Since the model requires text spans
as answers during training, we select the span which has the highest lexical overlap (F1
score) with the original answer as the gold answer. If the answer appears multiple times in
the story we use the rationale to find the correct one. If any answer word does not appear
in the passage, we fall back to an additional unknown token as the answer (about 17%).
We prepend each question with its past questions and answers to account for conversation
history, similar to the conversational models.

### 6.3.3 A Hybrid Model

The last model we build is a hybrid model, which combines the advantages of the afore-
mentioned two models. The reading comprehension models can predict a text span as an
answer, while they can’t produce answers that do not overlap with the passage. There-
fore, we combine STANFORD ATTENTIVE READER with PGNET to address this problem
since PGNET can generate free-form answers effectively. In this hybrid model, we use the
reading comprehension model to first point to the answer evidence in text, and PGNET nat-
uralizes the evidence into the final answer. For example, for $Q_5$ in Figure 6.1, we expect that
the reading comprehension model first predicts the rationale $R_5$ *Her granddaughter Annie
was coming over in the afternoon and Jessica was very excited to see her. Her daughter
Melanie and Melanie’s husband Josh were coming as well.*, and then PGNET generates $A_5$
*Annie, Melanie and Josh* from $R_5$.

We make a few changes to both models based on empirical performance. For the STAN-
FORD ATTENTIVE READER model, we only use rationales as answers for the questions
with an non-extractive answer. For PGNET, we only provide current question and span
predictions from the the STANFORD ATTENTIVE READER model as input to the encoder.
During training, we feed the oracle spans into PGNET.
6.4 Experiments

6.4.1 Setup

For the seq2seq and PGNet experiments, we use the OpenNMT toolkit (Klein et al., 2017). For the reading comprehension experiments, we use the same implementation that we used for SQUAD (Chen et al., 2017). We tune the hyperparameters on the development data: the number of turns to use from the conversation history, the number of layers, number of each hidden units per layer and dropout rate. We initialize the word projection matrix with GloVe (Pennington et al., 2014) for conversational models and FastText (Bojanowski et al., 2017) for reading comprehension models, based on empirical performance. We update the projection matrix during training in order to learn embeddings for delimiters such as \(<q>\).

For all the experiments of seq2seq and PGNet, we use the default settings of OpenNMT: 2-layers of LSTMs with 500 hidden units for both the encoder and the decoder. The models are optimized using SGD, with an initial learning rate of 1.0 and a decay rate of 0.5. A dropout rate of 0.3 is applied to all layers.

For all the reading comprehension experiments, the best configuration we find is 3 layers of LSTMs with 300 hidden units for each layer. A dropout rate of 0.4 is applied to all LSTM layers and a dropout rate of 0.5 is applied to word embeddings.

6.4.2 Experimental Results

Table 6.5 presents the results of the models on the development and the test data. Considering the results on the test set, the seq2seq model performs the worst, generating frequently occurring answers irrespective of whether these answers appear in the passage or not, a well known behavior of conversational models (Li et al., 2016). PGNet alleviates the frequent response problem by focusing on the vocabulary in the passage and it outperforms seq2seq by 17.8 points. However, it still lags behind STANFORD ATTENTIVE READER by 8.5 points. A reason could be that PGNet has to memorize the whole passage before answering a question, a huge overhead which STANFORD ATTENTIVE READER avoids. But STANFORD ATTENTIVE READER fails miserably in answering questions with
### Table 6.5: Models and human performance (F1 score) on the development and the test data.

<table>
<thead>
<tr>
<th></th>
<th>In-domain</th>
<th>Out-of-domain</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Children</td>
<td>Literature</td>
<td>Exam</td>
</tr>
<tr>
<td><strong>Development data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ2SEQ</td>
<td>30.6</td>
<td>26.7</td>
<td>28.3</td>
</tr>
<tr>
<td>PGNET</td>
<td>49.7</td>
<td>42.4</td>
<td>44.8</td>
</tr>
<tr>
<td>SAR</td>
<td>52.4</td>
<td>52.6</td>
<td>51.4</td>
</tr>
<tr>
<td>HYBRID</td>
<td><strong>64.5</strong></td>
<td><strong>62.0</strong></td>
<td><strong>63.8</strong></td>
</tr>
<tr>
<td>HUMAN</td>
<td>90.7</td>
<td>88.3</td>
<td>89.1</td>
</tr>
<tr>
<td></td>
<td><strong>Test data</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ2SEQ</td>
<td>32.8</td>
<td>25.6</td>
<td>28.0</td>
</tr>
<tr>
<td>PGNET</td>
<td>49.0</td>
<td>43.3</td>
<td>47.5</td>
</tr>
<tr>
<td>SAR</td>
<td>46.7</td>
<td>53.9</td>
<td>54.1</td>
</tr>
<tr>
<td>HYBRID</td>
<td><strong>64.2</strong></td>
<td><strong>63.7</strong></td>
<td><strong>67.1</strong></td>
</tr>
<tr>
<td>HUMAN</td>
<td>90.2</td>
<td>88.4</td>
<td>89.8</td>
</tr>
</tbody>
</table>

SAR: Stanford Attentive Reader.

free-form answers (see row Abstractive in Table 6.6). When the Stanford Attentive Reader is fed into PGNet, we empower both Stanford Attentive Reader and PGNet — Stanford Attentive Reader in producing free-form answers; PGNet in focusing on the rationale instead of the passage. This combination outperforms the PGNet and the Stanford Attentive Reader models by 21.0 and 12.5 points respectively.

**Models vs. Humans.** The human performance on the test data is 88.8 F1, a strong agreement indicating that the COQA’s questions have concrete answers. Our best model is 23.7 points behind humans, suggesting that the task is difficult to accomplish with current models. We anticipate that using a state-of-the-art reading comprehension model (Devlin et al., 2018) may improve the results by a few points.

**In-domain vs. Out-of-domain.** All models perform worse on out-of-domain datasets compared to in-domain datasets. The best model drops by 6.6 points. For in-domain results, both the best model and humans find the literature domain harder than the others since literature’s vocabulary requires proficiency in English. For out-of-domain results, the Reddit domain is apparently harder. This could be because Reddit requires reasoning on longer passages (see Table 6.1).
Table 6.6: Fine-grained results of different question and answer types in the development set. For the question type results, we only analyze 150 questions as described in Section 6.2.3.

While humans achieve high performance on children’s stories, models perform poorly, probably due to the fewer training examples in this domain compared to others. Both humans and models find Wikipedia easy.
6.4.3 Error Analysis

Table 6.6 presents fine-grained results of models and humans on the development set. We observe that humans have the highest disagreement on the unanswerable questions. Sometimes, people guess an answer even when it is not present in the passage, e.g., one can guess the age of Annie in Figure 6.1 based on her grandmother’s age. The human agreement on abstractive answers is lower than on extractive answers. This is expected because our evaluation metric is based on word overlap rather than on the meaning of words. For the question *did Jenny like her new room?*, human answers *she loved it* and *yes* are both accepted.

Finding the perfect evaluation metric for abstractive responses is still a challenging problem (Liu et al., 2016) and beyond the scope of our work. For our models’ performance, SEQ2SEQ and PGN-ET perform well on the questions with abstractive answers, and Stanford Attentive Reader performs well on the questions with extractive answers, due to their respective designs. The combined model improves on both categories.

Among the lexical question types, humans find the questions with lexical matches the easiest followed by paraphrasing, and the questions with pragmatics the hardest — this is expected since questions with lexical matches and paraphrasing share some similarity with the passage, thus making them relatively easier to answer than pragmatic questions. The best model also follows the same trend. While humans find the questions without coreferences easier than those with coreferences (explicit or implicit), the models behave sporadically. It is not clear why humans find implicit coreferences easier than explicit coreferences. A conjecture is that implicit coreferences depend directly on the previous turn whereas explicit coreferences may have long distance dependency on the conversation.

**Importance of conversation history.** Finally, we examine how important the conversation history is for the dataset. Table 6.7 presents the results with a varied number of previous turns used as conversation history. All models succeed at leveraging history but only up to a history of one previous turn (except PGN-ET). It is surprising that using more turns could decrease the performance.

---

We collect children’s stories from MCTest which contains only 660 passages in total, of which we use 200 stories for development and test.
Table 6.7: Results on the development set with different history sizes. History size indicates the number of previous turns prepended to the current question. Each turn contains a question and its answer. SAR: STANFORD ATTENTIVE READER.

We also perform an experiment on humans to measure the trade-off between their performance and the number of previous turns shown. Based on the heuristic that short questions likely depend on the conversation history, we sample 300 one or two word questions, and collect answers to these varying the number of previous turns shown.

When we do not show any history, human performance drops to 19.9 F1 as opposed to 86.4 F1 when full history is shown. When the previous question and answer is shown, their performance boosts to 79.8 F1, suggesting that the previous turn plays an important role in making sense of the current question. If the last two questions and answers are shown, they reach up to 85.3 F1, almost close to the performance when the full history is shown. This suggests that most questions in a conversation have a limited dependency within a bound of two turns.

6.5 Discussion

So far, we have discussed the CoQA dataset and several competitive baselines based on conversational models and reading comprehension models. We hope that our efforts can enable the first step to building conversational QA agents.

On the one hand, we think there is ample room for further improving performance on CoQA: our hybrid system obtains an F1 score of 65.1%, which is still 23.7 points behind the human performance (88.8%). We encourage our research community to work on this dataset and push the limits of conversational question answering models. We think there are several directions for further improvement:
• All the baseline models we built only use the conversation history by simply concatenating the previous questions and answers with the current question. We think that there should be better ways to connect the history and the current question. For the questions in Table 6.4, we should build models to actually understand that *his* in the question *What was his name?* refers to the puppy, and the question *Where?* means *Where will Sirisena be sworn in?*. Indeed, a recent model FLOWQA (Huang et al., 2018a) proposed a solution to effectively stack single-turn models along the conversational flow and demonstrated a state-of-the-art performance on CoQA.

• Our hybrid model aims to combine the advantages from the span prediction reading comprehension models and the pointer-generator network model to address the nature of abstractive answers. However, we implemented it as a pipeline model so the performance of the second component depends on whether the reading comprehension model can extract the right piece of evidence from the passage. We think that it is desirable to build an end-to-end model which can extract rationales while also rewriting the rationale into the final answer.

• We think the rationales that we collected can be better leveraged into training models.

On the other hand, CoQA certainly has its limitations and we should explore more challenging and more useful datasets in the future. One clear limitation is that the conversations in CoQA are only turns of question and answer pairs. That means the answerer is only responsible for answering questions while she can’t ask any clarification questions or communicate with the questioner through conversations. Another problem is that CoQA has very few (1.3%) unanswerable questions, which we think are crucial in practical conversational QA systems.

In parallel to our work, Choi et al. (2018) also created a dataset of conversations in the form of questions and answers on text passages. In our interface, we show a passage to both the questioner and the answerer, whereas their interface only shows a title to the questioner and the full passage to the answerer. Since their setup encourages the answerer to reveal more information for the following questions, their answers are as long as 15.1 words on average (ours is 2.7). While the human performance on our test set is 88.8 F1, theirs is 74.6 F1. Moreover, while CoQA’s answers can be abstractive, their answers are restricted
to only extractive text spans. Our dataset contains passages from seven diverse domains, whereas their dataset is built only from Wikipedia articles about people. Also, concurrently, Saeidi et al. (2018) created a conversational QA dataset for regulatory text such as tax and visa regulations. Their answers are limited to yes or no along with a positive characteristic of permitting to ask clarification questions when a given question cannot be answered.
Chapter 7

Conclusions

In this dissertation, we gave readers a thorough overview of neural reading comprehension: the foundations (PART I) and the applications (PART II), as well as how we contributed to the development of this field since it emerged in late 2015.

In Chapter 2, we walked through the history of reading comprehension, which dates back to the 1970s. At the time, researchers already recognized its importance as a proper way of testing the language understanding abilities of computer programs. However, it was not until the 2010s that, reading comprehension started to be formulated as a supervised learning problem by collecting human-labeled training examples in the form of (passage, question, answer) triples. Since 2015, the field has been completed reshaped, by the creation of large-scale supervised datasets, and the development of neural reading comprehension models. Although it has been only 3 years so far, the field has been moving strikingly fast. Innovations in building better datasets and more effective models have occurred alternately, and both contributed to the development of the field. We also formally defined the task of reading comprehension, and described the four most common types of problems: cloze style, multiple choice, span prediction and free-form answers and their evaluation metrics.

In Chapter 3, we covered all the elements of modern neural reading comprehension models. We introduced the Stanford Attentive Reader, which we first proposed for the CNN/Daily Mail cloze style task, and is one of the earliest neural reading comprehension models in this field. Our model has been studied extensively on other cloze style
and multiple choice tasks. We later adapted it to the SQuAD dataset and achieved what was then state-of-the-art performance. Compared to conventional feature-based models, this model doesn’t rely on any downstream linguistic features and all the parameters are jointly optimized together. Through empirical experiments and a careful hand-analysis, we concluded that neural models are more powerful at recognizing lexical matches and paraphrases. We also discussed recent advances in developing neural reading comprehension models, including better word representations, attention mechanisms, alternatives to LSTMs, and other advances such as training objectives and data augmentation.

In Chapter 4, we discussed future work and open questions in this field. We examined error cases on SQuAD (for both our model and the state-of-the-art model which surpasses the human performance). We concluded that these models have been doing very sophisticated matching of text but they still have difficulty understanding the inherent structure between entities and the events expressed in the text. We later discussed future work in both models and datasets. For models, we argued that besides accuracy, there are other important aspects which have been overlooked that we will need to work on in the future, including speed and scalability, robustness, and interpretability. We also believe that future models will need more structures and modules to solve more difficult reading comprehension problems. For datasets, we discussed more recent datasets developed after SQuAD — these datasets either require more complex reasoning across sentences or documents, or need to handle longer documents, or need to generate free-form answers instead of extracting a single span, or predict when there is no answer in the passage. Lastly, we examined several questions we think are important to the future of neural reading comprehension.

In PART II, the key questions we wanted to answer are: Is reading comprehension only a task of measuring language understanding? If we can build high-performing reading comprehension systems which can answer comprehension questions over a short passage of text, can it enable useful applications?

In Chapter 5, we showed that we can combine information retrieval techniques and neural reading comprehension models to build an open-domain question-answering system: answering general questions over a large encyclopedia or the Web. In particular, we implemented this idea in the DRQA project, a large-scale, factoid question answering system over English Wikipedia. We demonstrated the feasibility of doing this by evaluating
the system on multiple question answering benchmarks. We also proposed a procedure to automatically create additional distantly-supervised training examples from other question answering resources and demonstrated the effectiveness of this approach. We hope that our work takes the first step in this research direction and this new paradigm of combining information retrieval and neural reading comprehension will eventually lead to a new generation of open-domain question answering systems.

In Chapter 6, we addressed the conversational question answering problem, where a computer system needs to understand a text passage and answer a series of questions that appear in a conversation. To approach this, we built CoQA: a Conversational Question Answering challenge for measuring the ability of machines to participate in a question-answering style conversation. Our dataset contains 127k questions with answers, obtained from 8k conversations about text passages from seven diverse domains. We also built several competitive baselines for this new task, based on conversational and reading comprehension models. We believe that building such systems will play a crucial role in our future conversational AI systems.

All together, we are really excited about the progress that has been made in this field for the past 3 years and have been glad to be able to contribute to this field. At the same time, we also deeply believe there is still a long way to go towards genuine human-level reading comprehension, and we are still facing enormous challenges and a lot of open questions that we will need to address in the future. One key challenge is that we still don’t have good ways to approach deeper levels of reading comprehension — those questions which require understanding the reasoning and implications of the text. Often this occurs with how or why questions, such as In the story, why is Cynthia upset with her mother?, How does John attempt to make up for his original mistake? In the future, we will have to address the underlying science of what is being discussed, rather than just answering from text matching, to achieve this level of reading comprehension.

We also hope to encourage more researchers to work on the applications, or apply neural reading comprehension to new domains or tasks. We believe that it will lead us towards building better question answering and conversational agents and hope to see these ideas implemented and developed in industry applications.
Bibliography


Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.


