Interpretation of Natural Language Rules in Conversational Machine Reading

Marzieh Saeidi[1*], Max Bartolo[1*], Patrick Lewis[1*], Sameer Singh[1,2], Tim Rocktäschel[3], Mike Sheldon[1], Guillaume Bouchard[1], and Sebastian Riedel[1,3]

[1]Bloomsbury AI
[2]University of California, Irvine
[3]University College London

{marzieh.saeidi,maxbartolo,patrick.s.h.lewis}@gmail.com

Abstract

Most work in machine reading focuses on question answering problems where the answer is directly expressed in the text to read. However, many real-world question answering problems require the reading of text not because it contains the literal answer, but because it contains a recipe to derive an answer together with the reader's background knowledge. One example is the task of interpreting regulations to answer "Can I...?" or "Do I have to...?" questions such as "I am working in Canada. Do I have to carry on paying UK National Insurance?" after reading a UK government website about this topic. This task requires both the interpretation of rules and the application of background knowledge. It is further complicated due to the fact that, in practice, most questions are underspecified, and a human assistant will regularly have to ask clarification questions such as "How long have you been working abroad?" when the answer cannot be directly derived from the question and text. In this paper, we formalise this task and develop a crowd-sourcing strategy to collect 32k task instances based on real-world rules and crowd-generated questions and scenarios. We analyse the challenges of this task and assess its difficulty by evaluating the performance of rule-based and machine-learning baselines. We observe promising results when no background knowledge is necessary, and substantial room for improvement whenever background knowledge is needed.

1 Introduction

There has been significant progress in teaching machines to read text and answer questions when the answer is directly expressed in the text (Rajpurkar et al., 2016; Joshi et al., 2017; Welbl et al., 2018; Hermann et al., 2015). However, in many settings, the text contains rules expressed in natural language that can be used to infer the answer when combined with background knowledge, rather than the literal answer. For example, to answer someone's question "I am working for an employer in Canada. Do I need to carry on paying National Insurance?" with "Yes", one needs to read that "You'll carry on paying National Insurance if you're working for an employer outside the EEA" and understand how the rule and question determine the answer.

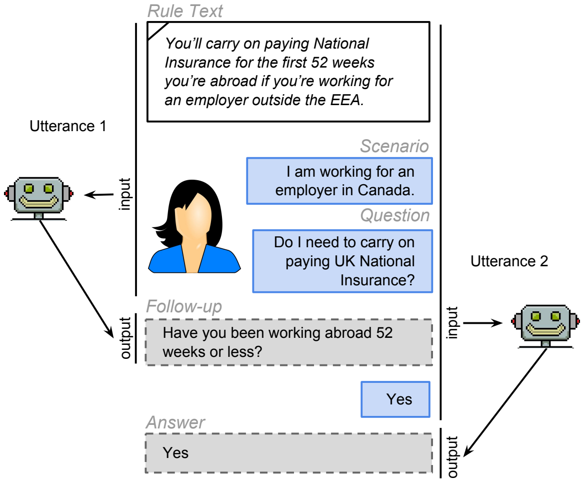
Rule Text
You'll carry on paying National Insurance for the first 52 weeks you're abroad if you're working for an employer outside the EEA.
Utterance 1
Scenario
I am working for an employer in Canada.
Question
Do I need to carry on paying UK National Insurance?
input
output
Follow-up
Have you been working abroad 52 weeks or less?
Utterance 2
input
Yes
Answer
Yes
output

Figure 1: An example of two utterances for rule interpretation. In the first utterance, a follow-up question is generated. In the second, the scenario, history and background knowledge (Canada is not in the EEA) is used to arrive at the answer "Yes".

Answering questions that require rule interpretation is often further complicated due to missing information in the question. For example, as illustrated in Figure 1 (Utterance 1), the actual rule also mentions that National Insurance only needs to be paid for the first 52 weeks when abroad. This means that we cannot answer the original question without knowing how long the user has already

[*]These three authors contributed equally

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2087–2097
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

been working abroad. Hence, the correct response in this conversational context is to issue another query such as "Have you been working abroad 52 weeks or less?"

To capture the fact that question answering in the above scenario requires a dialog, we hence consider the following *conversational machine reading* (CMR) problem as displayed in Figure 1: Given an input question, a context scenario of the question, a snippet of supporting rule text containing a rule, and a history of previous follow-up questions and answers, predict the answer to the question ("Yes"or "No") or, if needed, generate a follow-up question whose answer is necessary to answer the original question. Our goal in this paper is to create a corpus for this task, understand its challenges, and develop initial models that can address it.

To collect a dataset for this task, we could give a textual rule to an annotator and ask them to provide an input question, scenario, and dialog in one go. This poses two problems. First, this setup would give us very little control. For example, users would decide which follow-up questions become part of the scenario and which are answered with "Yes" or "No". Ultimately, this can lead to bias because annotators might tend to answer "Yes", or focus on the first condition. Second, the more complex the task, the more likely crowd annotators are to make mistakes. To mitigate these effects, we aim to break up the utterance annotation as much as possible.

We hence develop an annotation protocol in which annotators collaborate with virtual users—agents that give system-produced answers to follow-up questions—to incrementally construct a dialog based on a snippet of rule text and a simple underspecified initial question (e.g., "Do I need to ...?"), and then produce a more elaborate question based on this dialog (e.g., "I am ... Do I need to...?"). By controlling the answers of the virtual user, we control the ratio of "Yes" and "No" answers. And by showing only subsets of the dialog to the annotator that produces the scenario, we can control what the scenario is capturing. The question, rule text and dialogs are then used to produce utterances of the kind we see in Figure 1. Annotators show substantial agreement when constructing dialogs with a three-way annotator agreement at a Fleiss' Kappa level of 0.71.[1] Likewise, we find that our crowd-annotators produce questions that are coherent with the given dialogs with high accuracy.

In theory, the task could be addressed by an end-to-end neural network that encodes the question, history and previous dialog, and then decodes a Yes/No answer or question. In practice, we test this hypothesis using a seq2seq model (Sutskever et al., 2014; Cho et al., 2014), with and without copy mechanisms (Gu et al., 2016) to reflect how follow-up questions often use lexical content from the rule text. We find that despite a training set size of 21,890 training utterances, successful models for this task need a stronger inductive bias due to the inherent challenges of the task: interpreting natural language rules, generating questions, and reasoning with background knowledge. We develop heuristics that can work better in terms of identifying what questions to ask, but they still fail to interpret scenarios correctly. To further motivate the task, we also show in oracle experiments that a CMR system can help humans to answer questions faster and more accurately.

This paper makes the following contributions:

1. We introduce the task of conversational machine reading and provide evaluation metrics.
2. We develop an annotation protocol to collect annotations for conversational machine reading, suitable for use in crowd-sourcing platforms such as Amazon Mechanical Turk.
3. We provide a corpus of over 32k conversational machine reading utterances, from domains such as grant descriptions, traffic laws and benefit programs, and include an analysis of the challenges the corpus poses.
4. We develop and compare several baseline models for the task and subtasks.

2 Task Definition

Figure 1 shows an example of a conversational machine reading problem. A user has a question that relates to a specific rule or part of a regulation, such as "Do I need to carry on paying National Insurance?". In addition, a natural language description of the context or *scenario*, such as "I am working for an employer in Canada", is provided. The question will need to be answered using a small snippet of supporting *rule text*. Akin to machine reading problems in previous work (Rajpurkar et al., 2016; Hermann et al., 2015), we assume that this snippet is pre-identified. We generally assume that the question is *underspecified*, in the sense that the

[1] This is well within the range of what is considered as substantial agreement (Artstein and Poesio, 2008).

question often does not provide enough information to be answered directly. However, an agent can use the supporting rule text to infer what needs to be asked in order to determine the final answer. In Figure 1, for example, a reasonable follow-up question is “Have you been working abroad 52 weeks or less?”.

We formalise the above task on a per-utterance basis. A given dialog corresponds to a sequence of prediction problems, one for each utterance the system needs to produce. Let W be a vocabulary. Let $q = w_1 \dots w_{n_q}$ be an *input question* and $r = w_1 \dots w_{n_r}$ an input *support* rule text, where $w_i \in W$ is a word from a vocabulary. Furthermore, let $h = (f_1, a_1) \dots (f_{n_h}, a_{n_h})$ be a dialog *history* where each $f_i \in W^*$ is a *follow-up question*, and each $a_i \in \{\text{YES}, \text{NO}\}$ is a *follow-up answer*. Let s be a scenario describing the context of the question. We will refer to $x = (q, r, h, s)$ as the *input*. Given an input x, our task is to predict an *answer* $y \in \{\text{YES}, \text{NO}, \text{IRRELEVANT}\} \cup W^*$ that specifies whether the answer to the input question, in the context of the rule text and the previous follow-up question dialog, is either YES, NO, IRRELEVANT or another follow-up question in W^*. Here IRRELEVANT is the target answer whenever a rule text is not related to the question q.

3 Annotation Protocol

Our annotation protocol is depicted in Figure 2 and has four high-level stages: Rule Text Extraction, Question Generation, Dialog Generation and Scenario Annotation. We present these stages below, together with discussion of our quality-assurance mechanisms and method to generate negative data. For more details, such as annotation interfaces, we refer the reader to Appendix A.

3.1 Rule Text Extraction Stage

First, we identify the source documents that contain the rules we would like to annotate. Source documents can be found in Appendix C. We then convert each document to a set of *rule texts* using a heuristic which identifies and groups paragraphs and bulleted lists. To preserve readability during the annotation, we also split by a maximum rule text length and a maximum number of bullets.

3.2 Question Generation Stage

For each rule text we ask annotators to come up with an input question. Annotators are instructed to ask questions that cannot be answered directly but instead require follow-up questions. This means that the question should a) match the topic of the support rule text, and b) be underspecified. At present, this part of the annotation is done by expert annotators, but in future work we plan to crowdsource this step as well.

3.3 Dialog Generation Stage

In this stage, we view human annotators as assistants that help users reach the answer to the input question. Because the question was designed to be broad and to omit important information, human annotators will have to ask for this information using the rule text to figure out which question to ask. The follow-up question is then sent to a *virtual user*, *i.e.*, a program that simply generates a random YES or NO answer. If the input question can be answered with this new information, the annotator should enter the respective answer. If not, the annotator should provide the next follow-up question and the process is repeated.

When the virtual user is providing random YES and NO answers in the dialog generation stage, we are traversing a specific branch of a decision tree. We want the corpus to reflect all possible dialogs for each question and rule text. Hence, we ask annotators to label additional branches. For example, if the first annotator received a YES as the answer to the second follow-up question in Figure 3, the second annotator (orange) receives a NO.

3.4 Scenario Annotation Stage

In the final stage, we choose parts of the dialogs created in the previous stage and present this to an annotator. For example, the annotator sees “Are you working or preparing for work?” and NO. They are then asked to write a scenario that is consistent with this dialog such as “I am currently out of work after being laid off from my last job, but am not able to look for any yet.”. The number of questions and answers that the annotator is presented with for generating a scenario can vary from one to the full length of a dialog. Users are encouraged to paraphrase the questions and not to use many words from the dialog.

In an attempt to make these scenarios closer to the real-world situations where a user may provide a lot of unnecessary information to an operator, not only do we present users with one or more questions and answers from a specific dialog but

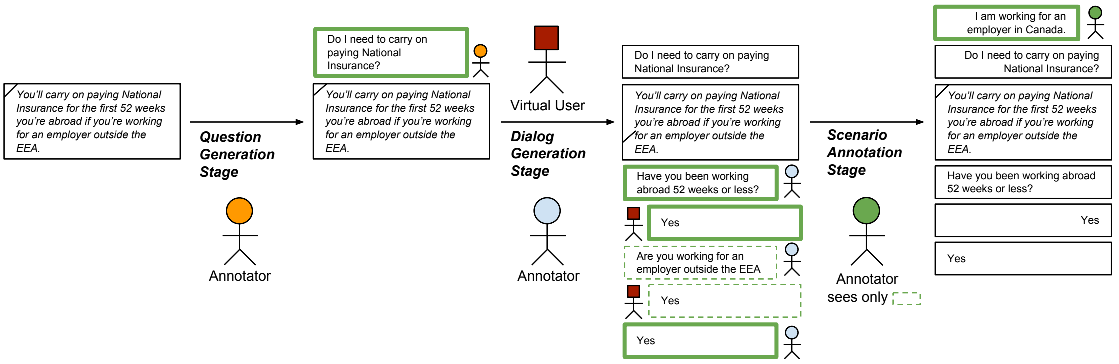
You'll carry on paying National Insurance for the first 52 weeks you're abroad if you're working for an employer outside the EEA.
Question Generation Stage
Annotator
Do I need to carry on paying National Insurance?
You'll carry on paying National Insurance for the first 52 weeks you're abroad if you're working for an employer outside the EEA.
Virtual User
Dialog Generation Stage
Annotator
Do I need to carry on paying National Insurance?
You'll carry on paying National Insurance for the first 52 weeks you're abroad if you're working for an employer outside the EEA.
Have you been working abroad 52 weeks or less?
Yes
Are you working for an employer outside the EEA
Yes
Yes
Scenario Annotation Stage
Annotator sees only
I am working for an employer in Canada.
Do I need to carry on paying National Insurance?
You'll carry on paying National Insurance for the first 52 weeks you're abroad if you're working for an employer outside the EEA.
Have you been working abroad 52 weeks or less?
Yes
Yes

Figure 2: The different stages of the annotation process (excluding the rule text extraction stage). First a human annotator generates an underspecified input question (question generation). Then, a virtual user and a human annotator collaborate to produce a dialog of follow-up questions and answers (dialog generation). Finally, a scenario is generated from parts of the dialog, and these parts are omitted in the final result.

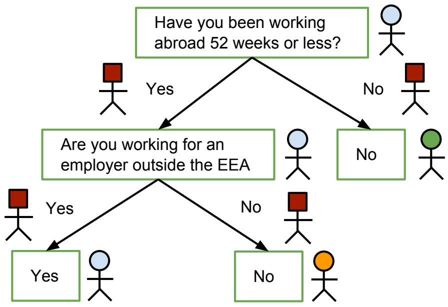
Have you been working abroad 52 weeks or less?
Yes
No
Are you working for an employer outside the EEA
No
Yes
No
Yes
No

Figure 3: We use different annotators (indicated by different colors) to create the complete dialog tree.

also with one question from a random dialog. The annotators are asked to come up with a scenario that fits all the questions and answers.

Finally, a dialog is produced by combining the scenario with the input question and rule text from the previous stages. In addition, all dialog utterances that were *not* shown to the final annotator are included as well as they complement the information in the scenario. Given a dialog of this form, we can create utterances that are described in Section 2.

As a result of this stage of annotation, we create a corpus of scenarios and questions where the correct answers (YES, NO or IRRELEVANT) to questions can be derived from the related scenarios. This corpus and its challenges will be discussed in Section 4.2.2.

3.5 Negative Examples

To facilitate the future application of the models to large-scale rule-based documents instead of rule text, we deem it to be imperative for the data to contain negative examples of both questions and scenarios.

We define a *negative question* as a question that is not relevant to the rule text. In this case, we expect models to produce the answer IRRELEVANT. For a given rule text and question pair, a negative example is generated by sampling a random question from the set of all possible questions, excluding the question itself and questions sourced from the same document using a methodology similar to the work of Levy et al. (2017).

The data created so far is biased in the sense that when a scenario is given, at least one of the follow-up questions in a dialog can be answered. In practice, we expect users to also provide background scenarios that are completely irrelevant to the input question. Therefore, we sample a *negative scenario* for each input question and rule text pair, (q, r) in our data. We uniformly sample from the scenarios created in Section 3.4 for all question and rule text pairs (q', r') unequal to (q, r). For more details, we point the reader to Appendix D.

3.6 Quality Control

We employ a range of quality control measures throughout the process. In particular, we:

1. Re-annotate pre-terminal nodes in the dialog trees if they have identical YES and NO branches.
2. Ask annotators to validate the previous dialog in case previous utterances where created by different annotators.
3. Assess a sample of annotations for each an-

notator and keep only those annotators with quality scores higher than a certain threshold.

4. We require annotators to pass a qualification test before selecting them for our tasks. We also require high approval rates and restrict location to the UK, US, or Canada.

Further details are provided in Appendix B.

3.7 Cost, Duration and Scalability

The cost of different stages of annotation is as follows. An annotator was paid $0.15 for an initial question (948 questions), $0.11 for a dialog part (3000 dialog parts) and $0.20 for a scenario (6,600 scenarios). It takes in total 2 weeks to complete the annotation process. Considering that all the annotation stages can be done through crowdsourcing and in a relatively short time period and at a reasonable cost using established validation procedures, the dataset can be scaled up without major bottlenecks or an impact on the quality.

4 ShARC

In this section, we present the *SHaping Answers with Rules and Conversation (ShARC)* dataset.[2]

4.1 Dataset Size and Quality

The dataset is built up from of 948 distinct snippets of rule text. Each has an input question and a "dialog tree". At each step in the dialog, there is a followup question posed and the tree branches depending on the answer to the followup question (yes/no). The ShARC dataset is comprised of all individual "utterances" from every tree, i.e. every possible point/node in any dialog tree. There are 6058 of these utterances. In addition, there are 6637 scenarios that provide more information, allowing some questions in the dialog tree to be "skipped" as the answers can be inferred from the scenario. Scenarios therefore modify the dialog trees, which creates new trees. When combined with scenarios and negative sampled scenarios, the total number of distinct utterances became 37087. As a final step, utterances were removed where the scenario referred to a portion of the dialog tree that was unreachable for that utterance, leaving a final dataset size of 32436 utterances.[3]

We break these into train, development and test sets such that each dataset contains approximately the same proportion of sources from each domain, targeting a 70%/10%/20% split.

To evaluate the quality of dialog generation HITs, we sample a subset of 200 rule texts and questions and allow each HIT to be annotated by three distinct workers. In terms of deciding whether the answer is a YES, NO or some follow-up question, the three annotators reach an answer agreement of 72.3%. We also calculate Cohen's Kappa, a measure designed for situations with two annotators. We randomly select two out of the three annotations and compute the unweighted kappa values, repeated for 100 times and averaged to give a value of 0.82.

The above metrics measure whether annotators agree in terms of deciding between YES, NO or some follow-up question, but not whether the follow-up questions are equivalent. To approximate this, we calculate BLEU scores between pairs of annotators when they both predict follow-up questions Generally, we find high agreement: Annotators reach average BLEU scores of 0.71, 0.63, 0.58 and 0.58 for maximum orders of 1, 2, 3 and 4 respectively.

To get an indication of human performance on the sub-task of classifying whether a response should be a YES, NO or FOLLOW-UP QUESTION, we use a similar methodology to (Rajpurkar et al., 2016) by considering the second answer to each question as the human prediction and taking the majority vote as ground truth. The resulting human accuracy is 93.9%.

To evaluate the quality of the scenarios, we sample 100 scenarios randomly and ask two expert annotators to validate them. We perform validation for two cases: 1) scenarios generated by turkers who did not attempt the qualification test and were not filtered by our validation process, 2) scenarios that are generated by turkers who have passed the qualification test and validation process. In the second case, annotators approved an average of 89 of the scenarios whereas in the first case, they only approved an average of 38. This shows that the qualification test and the validation process im-

[2] The dataset and its Codalab challenge can be found at https://sharc-data.github.io.

[3] One may argue that the the size of the dataset is not sufficient for training end-to-end neural models. While we believe that the availability of large datasets such as SNLI or SQuAD has helped drive the state-of-the-art forward on related tasks, relying solely on large datasets to push the boundaries of AI cannot be as practical as developing better models for incorporating common sense and external knowledge which we believe ShARC is a good test-bed for. Furthermore, the proposed annotation protocol and evaluation procedure can be used to reliably extend the dataset or create datasets for new domains.

proved the quality of the generated scenarios by more than double. In both cases, the annotators agreed on the validity of 91-92 of the scenarios. For further details on dataset quality, the reader if referred to Appendix B.

4.2 Challenges

We analyse the challenges involved in solving conversational machine reading in ShARC. We divide these into two parts: challenges that arise when interpreting rules, and challenges that arise when interpreting scenarios.

4.2.1 Interpreting Rules

When no scenarios are available, the task reduces to a) identifying the follow-up questions within the rule text, b) understanding whether a follow-up question has already been answered in the history, and c) determining the logical structure of the rule (*e.g.*disjunction vs. conjunction vs. conjunction of disjunctions) .

To illustrate the challenges that these sub-tasks involve, we manually categorise a random sample of 100 (q_i, r_i) pairs. We identify 9 phenomena of interest, and estimate their frequency within the corpus. Here we briefly highlight some categories of interest, but full details, including examples, can be found in Appendix G.

A large fraction of problems involve the identification of at least two conditions, and approximately 41% and 27% of the cases involve logical disjunctions and conjunctions respectively. These can appear in linguistic coordination structures as well as bullet points. Often, differentiating between conjunctions and disjunctions is easy when considering bullets—key phrases such as “if all of the following hold” can give this away. However, in 13% of the cases, no such cues are given and we have to rely on language understanding to differentiate. For example:

Q: Do I qualify for Statutory Maternity Leave?

R: You qualify for Statutory Maternity Leave if
- you’re an employee not a “worker”
- you give your employer the correct notice

4.2.2 Interpreting Scenarios

Scenario interpretation can be considered as a multi-sentence entailment task. Given a scenario (premise) of (usually) several sentences, and a question (hypothesis), a system should output Yes (Entailment), No (Contradiction) or Irrelevant (Neutral). In this context, Irrelevant indicates that the answer to the question cannot be inferred from the scenario.

Different types of reasoning are required to interpret the scenarios. Examples include numerical reasoning, temporal reasoning and implication (common sense and external knowledge). We manually label 100 scenarios with the type of reasoning required to answer their questions. Table 1 shows examples of different types of reasoning and their percentages. Note that these percentages do not add up to 100% as interpreting a scenario may require more than one type of reasoning.

5 Experiments

To assess the difficulty of ShARC as a machine learning problem, we investigate a set of baseline approaches on the end-to-end task as well as the important sub-tasks we identified. The baselines are chosen to assess and demonstrate both feasibility and difficulty of the tasks.

Metrics For all following classification tasks, we use micro- and macro- averaged accuracies. For the follow-up generation task, we compute the BLEU scores at orders 1, 2, 3 and 4 computed between the gold follow-up questions, y_i and follow-up question $\hat{y}_i = w_{\hat{y}_i,1}, w_{\hat{y}_i,2} \ldots w_{\hat{y}_i,n}$ for all utterances i in the evaluation dataset.

5.1 Classification (excluding Scenarios)

On each turn, a CMR system needs to decide, either explicitly or implicitly, whether the answer is Yes or No, whether the question is not relevant to the rule text (Irrelevant), or whether a follow-up question is necessary—an outcome we label as More. In the following experiments, we will test whether one can learn to make this decision using the ShARC training data.

When a non-empty scenario is given, this task also requires an understanding of how scenarios answer follow-up questions. In order to focus on the challenges of rule interpretation, here we only consider empty scenarios.

Formally, for an utterance $x = (q, r, h, s)$, we require models to predict an answer y where $y \in \{\textsc{Yes}, \textsc{No}, \textsc{Irrelevant}, \textsc{More}\}$. Since we consider only the classification task without scenario influence, we consider the subset of utterances such that $s = NULL$. This data subset consists of 4026 train, 431 dev and 1601 test utterances.

Category	Questions	Scenario	%
Explicit	Has your wife reached state pension age? Yes	My wife just recently reached the age for state pension	25%
Temporal	Did you own it before April 1982? Yes	I purchased the property on June 5, 1980.	10%
Geographic	Do you normally live in the UK? No	I'm a resident of Germany.	7%
Numeric	Do you work less than 24 hours a week between you? No	My wife and I work long hours and get between 90 - 110 hours per week between the two of us.	12%
Paraphrase	Are you working or preparing for work? No	I am currently out of work after being laid off from my last job, but am not able to look for any yet.	19%
Implication	Are you the baby's father? No	My girlfriend is having a baby by her ex.	51%

Table 1: Types of reasoning and their proportions in the dataset based on 100 samples. Implication includes reasoning beyond what is explicitly stated in the text, including common sense reasoning and external knowledge.

Model	Micro Acc.	Macro Acc.
Random	0.254	0.250
Surface LR	0.555	0.511
Heuristic	0.791	0.779
Random Forest	**0.808**	**0.797**
CNN	0.677	0.681

Table 2: Selected Results of the baseline models on the classification sub-task.

Baselines We evaluate various baselines including random, a surface logistic regression applied to a TFIDF representation of the rule text, question and history, a rule-based heuristic which makes predictions depending on the number of overlapping words between the rule text and question, detecting conjunctive or disjunctive rules, detecting negative mismatch between the rule text and the question and what the answer to the last follow-up history was, a feature-engineered Random Forest and a Convolutional Neural Network applied to the tokenised inputs of the concatenated rule text, question and history.

Results We find that, for this classification sub-task, Random Forest slightly outperforms the heuristic. All learnt models considerably outperform the random and majority baselines.

5.2 Follow-up Question Generation without Scenarios

When the target utterance is a follow-up question, we still have to determine what that follow-up question is. For an utterance $x = (q, r, h, s)$, we require models to predict an answer y where y is the next follow-up question, $y = w_{y,1}, w_{y,2} \dots w_{y,n} = f_{m+1}$ if x has history of length m. We therefore consider the subset of utterances such that $s = NULL$ and $y \notin \{\textsc{Yes}, \textsc{No}, \textsc{Irrelevant}\}$. This data subset consists of 1071 train, 112 dev and 424 test utterances.

Baselines We first consider several simple baselines to explore the relationship between our evaluation metric and the task. As annotators are encouraged to re-use the words from rule text when generating follow-up questions, a baseline that simply returns the final sentence of the rule text performs surprisingly well. We also implement a rule-based model that uses several heuristics.

If framed as a seq2seq task, a modified CopyNet is most promising (Gu et al., 2016). We also experiment with span extraction/sequence-tagging approaches to identify relevant spans from the rule text that correspond to the next follow-up questions. We find that Bidirectional Attention Flow (Seo et al., 2017) performed well.[4] Further implementation details can be found in Appendix H.

Results Our results, shown in Table 3 indicate that systems that return contiguous spans from the rule text perform better according to our BLEU metric. We speculate that the logical forms in the data are challenging for existing models to extract and manipulate, which may suggest why the explicit rule-based system performed best. We further note that only the rule-based and NMT-Copy models are capable of generating genuine questions rather than spans or sentences.

5.3 Scenario Interpretation

Many utterances require the interpretation of the scenario associated with a question. If the scenario

[4] We use AllenNLP implementations of BiDAF & DAM

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
First Sent.	0.221	0.144	0.119	0.106
NMT-Copy	0.339	0.206	0.139	0.102
BiDAF	0.450	0.375	0.338	0.312
Rule-based	**0.533**	**0.437**	**0.379**	**0.344**

Table 3: Selected Results of the baseline models on follow-up question generation.

Model	Micro Acc.	Macro Acc.
Random	0.330	0.326
Surface LR	**0.682**	0.333
DAM (SNLI)	0.479	**0.362**
DAM (ShARC)	0.492	0.322

Table 4: Results of entailment models on ShARC.

is understood, certain follow-up questions can be skipped because they are answered within the scenario. In this section, we investigate how difficult scenario interpretation is by training models to answer follow-up questions based on scenarios.

Baselines We use a random baseline and also implement a surface logistic regression applied to a TFIDF representation of the combined scenario and the question. For neural models, we use Decomposed Attention Model (DAM) (Parikh et al., 2016) trained on each the SNLI and ShARC corpora using ELMO embeddings (Peters et al., 2018).[4]

Results Table 4 shows the result of our baseline models on the entailment corpus of ShARC test set. Results show poor performance especially for the macro accuracy metric of both simple baselines and neural state-of-the-art entailment models. This performance highlights the challenges that the scenario interpretation task of ShARC presents, many of which are discussed in Section 4.2.2.

5.4 Conversational Machine Reading

The CMR task requires all of the above abilities. To understand its core challenges, we compare baselines that are trained end-to-end vs. baselines that reuse solutions for the above subtasks.

Baselines We present a Combined Model (CM) which is a pipeline of the best performing Random Forest classification model, rule-based follow-up question generation model and Surface LR entailment model. We first run the classification model to predict YES, NO, MORE or IRRELEVANT. If MORE is predicted, the Follow-up Question Generation model is used to produce a follow-up question, f_1. The rule text and produced follow-up question are then passed as inputs to the Scenario Interpretation model. If the output of this is IRRELEVANT, then the CM predicts f_1, otherwise, these steps are repeated recursively until the classification model no longer predicts MORE or the entailment model predicts IRRELEVANT, in which case the model produces a final answer. We also investigate an extension of the NMT-copy model on the end-to-end task. Input sequences are encoded as a concatenation of the rule text, question, scenario and history. The model consists of a shared encoder LSTM, a 4-class classification head with attention, and a decoder GRU to generate followup questions. The model was trained by alternating training the classifier via standard softmax-cross entropy loss and the followup generator via seq2seq. At test time, the input is first classified, and if the predicted class is MORE, the follow-up generator is used to generate a followup question, f_1. A simpler model without the separate classification head failed to produce predictive results.

Model	Micro Acc	Macro Acc	BLEU-1	BLEU-4
CM	0.619	0.689	0.544	0.344
NMT	0.448	0.428	0.340	0.078

Table 5: Results of the models on the CMR task.

Results We find that the combined model outperforms the neural end-to-end model on the CMR task, however, the fact that the neural model has learned to classify better than random and also predict follow-up questions is encouraging for designing more sophisticated neural models for this task.

User Study In order to evaluate the utility of conversational machine reading, we run a user study that compares CMR to when such an agent is not available, i.e. the user has to read the rule text and determine themselves the answer to the question. On the other hand, with the agent, the user does not read the rule text, instead only responds to follow-up questions. Our results show that users using the conversational agent reach conclusions > 2 times faster than ones that are not, but more importantly, they are also much more accurate (93% as compared to 68%). Details of the experiments and the results are included in Appendix I.

6 Related Work

This work relates to several areas of active research.

Machine Reading In our task, systems answer questions about units of texts. In this sense, it is most related to work in Machine Reading (Rajpurkar et al., 2016; Seo et al., 2017; Weissenborn et al., 2017). The core difference lies in the conversational nature of our task: in traditional Machine Reading the questions can be answered right away; in our setting, clarification questions are often needed. The domain of text we consider is also different (regulatory vs Wikipedia, books, newswire).

Dialog The task we propose is, at its heart, about conducting a dialog (Weizenbaum, 1966; Serban et al., 2018; Bordes and Weston, 2016). Within this scope, our work is closest to work in dialog-based QA where complex information needs are addressed using a series of questions. In this space, previous approaches have been looking primarily at QA dialogs about images (Das et al., 2017) and knowledge graphs (Saha et al., 2018; Iyyer et al., 2017). In parallel to our work, both Choi et al. (2018) and Reddy et al. (2018) have to began to investigate QA dialogs with background text. Our work not only differs in the domain covered (regulatory text vs wikipedia), but also in the fact that our task requires the interpretation of complex rules, application of background knowledge, and the formulation of free-form clarification questions. Rao and Daume III (2018) does investigate how to generate clarification questions but this does not require the understanding of explicit natural language rules.

Rule Extraction From Text There is a long line of work in the automatic extraction of rules from text (Silvestro, 1988; Moulin and Rousseau, 1992; Delisle et al., 1994; Hassanpour et al., 2011; Moulin and Rousseau, 1992). The work tackles a similar problem—interpretation of rules and regulatory text—but frames it as a text-to-structure task as opposed to end-to-end question-answering. For example, Delisle et al. (1994) maps text to horn clauses. This can be very effective, and good results are reported, but suffers from the general problem of such approaches: they require careful ontology building, layers of error-prone linguistic preprocessing, and are difficult for non-experts to create annotations for.

Question Generation Our task involves the automatic generation of natural language questions. Previous work in question generation has focussed on producing questions for a given text, such that the questions can be answered using this text (Vanderwende, 2008; M. Olney et al., 2012; Rus et al., 2011). In our case, the questions to generate are *derived* from the background text but cannot be answered by them. Mostafazadeh et al. (2016) investigate how to generate natural follow-up questions based on the content of an image. Besides not working in a visual context, our task is also different because we see question generation as a sub-task of question answering.

7 Conclusion

In this paper we present a new task as well as an annotation protocol, a dataset, and a set of baselines. The task is challenging and requires models to generate language, copy tokens, and make logical inferences. Through the use of an interactive and dialog-based annotation interface, we achieve good agreement rates at a low cost. Initial baseline results suggest that substantial improvements are possible and require sophisticated integration of entailment-like reasoning and question generation.

Acknowledgements

This work was supported by an Allen Distinguished Investigator Award.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Antoine Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *CoRR*, abs/1605.07683.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC : Question Answering in Context. In *EMNLP*. ArXiv: 1808.07036.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jos MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2.

Sylvain Delisle, Ken Barker, Jean franois Delannoy, Stan Matwin, and Stan Szpakowicz. 1994. From text to horn clauses: Combining linguistic analysis and machine learning. In *In 10th Canadian AI Conf*, pages 9–16.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR*, abs/1603.06393.

Saeed Hassanpour, Martin O'Connor, and Amar Das. 2011. A framework for the automatic extraction of rules from online text.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based Neural Structured Learning for Sequential Question Answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:1821–1831.

Mandar Joshi, Eunsol Choi, Daniel S. 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*, Vancouver, Canada. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Andrew M. Olney, Arthur Graesser, and Natalie Person. 2012. Question generation from concept maps. 3.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. *CoRR*, abs/1603.06059.

B. Moulin and D. Rousseau. 1992. Automated knowledge acquisition from regulatory texts. *IEEE Expert*, 7(5):27–35.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Sudha Rao and Hal Daume III. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. CoQA: A Conversational Question Answering Challenge. *arXiv:1808.07042 [cs]*. ArXiv: 1808.07042 Citation Key: reddyCoQAConversationalQuestion2018.

Vasile Rus, Paul Piwek, Svetlana Stoyanchev, Brendan Wyse, Mihai Lintean, and Cristian Moldovan. 2011. Question generation shared task and evaluation challenge: Status report. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, pages 318–320, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *The International Conference on Learning Representations (ICLR)*.

Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2018. A Survey of Available Corpora For Building Data-Driven Dialogue Systems: The Journal Version. *Dialogue & Discourse*, 9(1):1–49.

Kenneth Silvestro. 1988. Using explanations for knowledge-base acquisition. *International Journal of Man-Machine Studies*, 29(2):159 – 169.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Lucy Vanderwende. 2008. The importance of being important: Question generation. In *In Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Fastqa: A simple and efficient neural architecture for question answering. *CoRR*, abs/1703.04816.

Joseph Weizenbaum. 1966. ELIZAa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of ACL*, abs/1710.06481.