Neural Conversational QA: Learning to Reason v.s. Exploiting Patterns

Abhishek Sharma[1,2*] **Danish Contractor**[1] **Harshit Kumar**[1] **Sachindra Joshi**[1]
[1]IBM Research AI, New Delhi [2]Indian Institute of Technology (BHU), Varanasi
abhishek.sharma.mat16@itbhu.ac.in, {dcontrac,harshitk,jsachind}@in.ibm.com

Abstract

In this paper we work on the recently introduced ShARC task - a challenging form of conversational QA that requires reasoning over rules expressed in natural language. Attuned to the risk of superficial patterns in data being exploited by neural models to do well on benchmark tasks (Niven and Kao 2019), we conduct a series of probing experiments and demonstrate how current state-of-the-art models rely heavily on such patterns. To prevent models from learning based on the superficial clues, we modify the dataset by automatically generating new instances reducing the occurrences of those patterns. We also present a simple yet effective model that learns embedding representations to incorporate dialog history along with the previous answers to follow-up questions. We find that our model outperforms existing methods on all metrics, and the results show that the proposed model is more robust in dealing with spurious patterns and learns to reason meaningfully.

1 Introduction

A recently introduced class of question-answering tasks, called conversational QA, involves answering questions as part of a conversation flow. In tasks such as QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2019), a user may ask a question as part of a conversation, and the answer to that question may depend on contextual references from previously asked questions or their answers (See Figure 1 (a)). Such tasks assume that questions (with the inclusion of conversation history) are deemed to be fully specified. The answers to such questions are either spans extracted from a passage or are generated based on information present in the passage.

In contrast, the recently introduced ShARC task (Saeidi et al., 2018), requires a system to answer user questions about rules and policies specified in natural language text. This is a challenging problem - not only does the system need to understand a user's question (along with any background information that the user might provide), it also needs to comprehend them in the context of an applicable rule and then provide a yes-no answer to a user's questions. Additionally, it may choose to ask follow-up questions to determine user eligibility before returning the final yes-no answer. For example, in Figure 1 (b), a rule/policy needs to be studied in the context of a user who shares some background information (scenario) about herself and asks a question that the system needs to answer. The system in-turn asks a follow-up question to seek more details before answering the original question; a user's response to these follow-up questions is either yes or no.

Recent models such as BERT-QA (Devlin et al., 2019), E3 (Zhong and Zettlemoyer, 2019), BiSon (Lawrence et al., 2019) have been applied and demonstrated to perform reasonably well on this task. Our exploration with this data and task indicate that there are spurious patterns that exist in the data which could be exploited by neural models to obtain good performance. This is similar to recent observations made by Niven and Kao who show that BERT's peak performance on argument reasoning tasks could be attributed to the exploitation of spurious statistical clues in the dataset (Niven and Kao, 2019).

In this task, we observe that 84% of the time, the answer provided by a user to the last follow-up question, is the final answer to the user question. This is because most rules in the data-set have either only conjunctive clauses or disjunctive clauses, if there is more than one follow-up question with "no" as an answer, it is likely to be

*This work was carried out as part of an internship at IBM Research

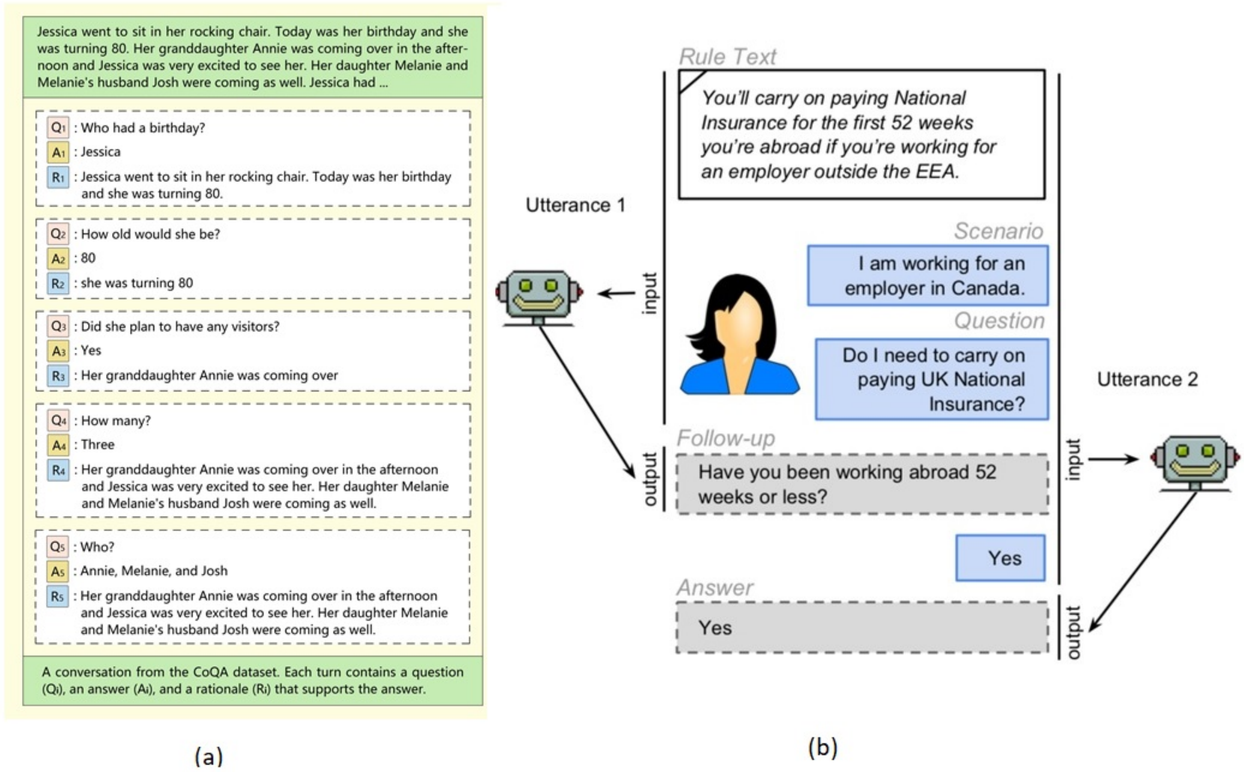

Figure 1: Conversational QA Tasks: (a) Reading Comprehension based conversation QA tasks where answers are generated from passages (Image from `https://bit.ly/2k7mFP0`). (b) Answering questions that require rule interpretation and the generation of clarification questions (Image from (Saeidi et al., 2018)).

a disjunctive rule. Similarly, for rules with conjunctive clauses, the conversation contains a series of follow-up questions with "yes" as an answer. The yes-no answer to the last follow-up question is then the final answer to the original question.

In addition, the length (number of turns) of the conversation history also induces a pattern on the likelihood of asking follow-up questions. Since there are only a finite number of clauses from which follow-up questions can be asked; As the number of turns in the history increase, the probability of asking a follow-up question decreases. While this is expected in the real-world, we demonstrate that these patterns make the model rely on these clues reducing its ability to learn other things.

This paper presents our model, *UrcaNet*, that learns embedding representations to incorporate dialog history along with the previous answers to follow-up questions. We also include embeddings that interpret user background (scenario) in the context of the rule passage. We use BERT (Devlin et al., 2019) to generate intermediate representations and apply a copy-decoder (Gu et al., 2016) to generate follow-up questions. To prevent our model from learning based on these spurious clues, we augment the dataset by automatically generating new instances; for example - we shuffle follow-up questions for a given question and include them as training samples (See Experiments for more details). While this exercise generates artificial examples unlikely to be encountered in the real world, we find that it can help our model learn meaningfully by reducing the spurious patterns available for exploitation. Our model outperforms existing systems on both the original as well as the instance augmented versions of the dataset. We also find that the current state-of-the-art model, E3, (Zhong and Zettlemoyer, 2019) suffers a 19% (relative) drop in performance[1] on the augmented dataset suggesting that it relies more heavily on the spurious clues present in the original dataset.

1.1 Contributions

1. While the existence of spurious patterns have been demonstrated in task such as reading comprehension and argument reasoning, we

[1] "Combined" metric (Zhong and Zettlemoyer, 2019)

are the first to demonstrate that conversational QA tasks also suffer from this phenomenon.

2. We share an augmented version of the original dataset that reduces the risk of learning from spurious clues.

3. We present a simple yet effective model that beats existing state-of-the-art on the ShARC conversation QA task.

2 Related Work

In the last few years, a series of NLP benchmarking tasks, that serve as datasets for standardized comparison, have been introduced by the research community; these include Question Answering tasks (Nguyen et al., 2016; Joshi et al., 2017; Mihaylov et al., 2018; Yang et al., 2018), and specifically conversational QA tasks (Choi et al., 2018; Reddy et al., 2019). Recent studies on neural models that perform well on such benchmark tasks have revealed several weaknesses; they are prone to easy adversarial attacks suggesting limited reasoning capabilities (Wallace et al., 2019), they often exploit spurious patterns in the dataset (Niven and Kao, 2019) or can even end up modeling annotator bias to do well on tasks (Geva et al., 2019). Entailment-driven Extract and Edit (E3) network (Zhong and Zettlemoyer, 2019) solves the ShARC task by learning to extract implicit rules in the rule, selecting which rules are entailed by the conversation history, and edit rules that are not entailed to create follow-up questions to the user. BertQA (Zhong and Zettlemoyer, 2019) is an adaption of the BERT model on the ShARC task where additional tokens (yes/no/irrelevant) are added to the rule passage.

In this paper, we report our findings on the possible exploits used by neural models on the ShARC task (Saeidi et al., 2018). To the best of our knowledge we are the first to present a detailed study highlighting these aspects in a conversational QA task; our intuitive approach of automatically generating additional instances reduces the effect of spurious patters. Our model architecture is based on recent state of the art QA models (Devlin et al., 2019) and similar to recent work on the QuAC task (Qu et al., 2019), we incorporate marker embeddings to model dialog history. We however, extend marker embeddings to also model facts entailed in unstructured text[2]. The current state-of-the-art model E3(Zhong and Zettlemoyer, 2019) is also based on BERT but and uses methods based on semantic overlap to relate rule clauses with user input.

3 Model

Given a *rule* passage r, *question* q, *scenario* s – background information provided by a user, and dialog *history* – a series of past follow-up question and answer pairs $(hq_1, ha_1), \dots (hq_{N_h}, ha_{N_h})$, where each follow-up answer is either `Yes` or `No`; the system is expected to respond in one of following ways: (i) Answer the user question q with a `Yes` or a `No`. (ii) Identify the question as being `irrelevant` with respect to the clauses in the *rule* passage. (iii) Generate a follow-up clarification question based on clauses in the *rule* passage.

Rule	## Sanctionable benefits The following benefits can be reduced or stopped if you commit benefit fraud: * CarerâĂŹs Allowance * Employment and Support Allowance * Housing Benefits * Incapacity Benefit
Scenario	I am a 40 year old man working as an engineer.
Question	Can my benefit be reduced or stopped?
History	
System	Do you get Carer's Allowance?
User	No
Ground truth	Do you get housing benefits?

Table 1: A sample instance from the ShARC dataset (Saeidi et al., 2018)

The proposed model, *UrcaNet*, consists of the following elements:

1. **BERT:** to encode text of the *question*, *rule*, *history* and *scenario*. (Devlin et al., 2019).

2. **Dialog History Embedding:** Marker embeddings indicative of yes-no answers and the turn number for each follow-up question in the *history*.

3. **Scenario Embeddeings:** Marker embeddings generated by interpreting the *rule* and the user background (*scenario*) indicative of information shared by the user with respect to the clauses in the *rule*.

[2] *Scenario* in the ShARC task.

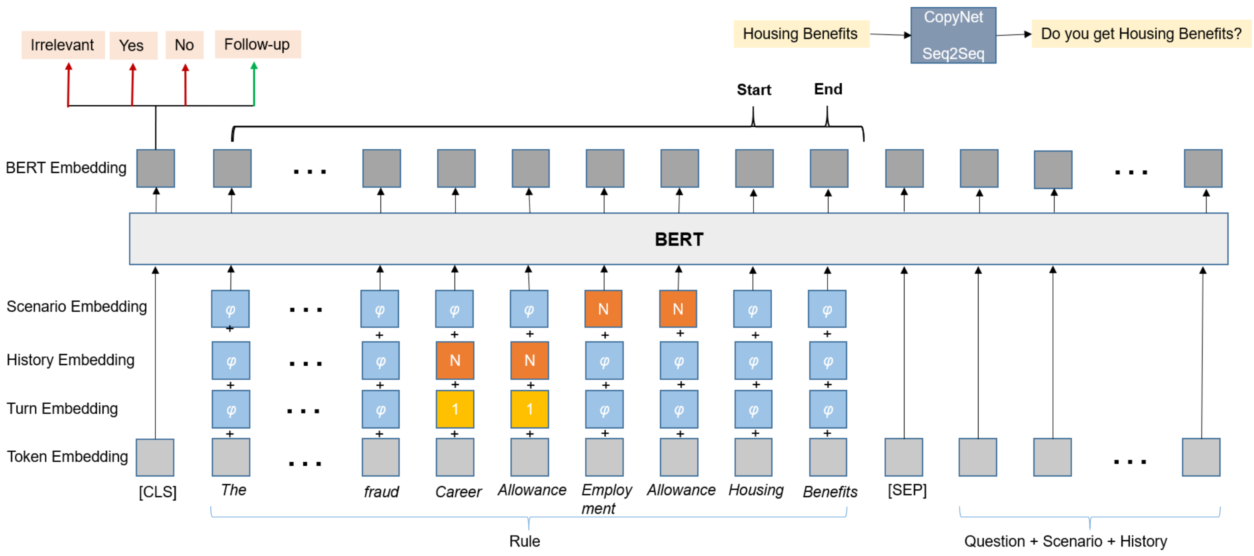

Figure 2: The *UrcaNet* Model - using the example in Table 1 as input.

4. **Follow-up question Generator:** A sequence to sequence model with copy mechanism (Gu et al., 2016) to generate the follow-up question.

3.1 Input Embeddings and BERT

We define context c as the concatenation of embedded rule r, question q, history h and scenario s. We add `[CLS]` token in the beginning and use `[SEP]` token as a separator during concatenation to demarcate boundaries. The context c is fed as input to a BERT encoder that outputs a combined representation of the context, $|c| \times \text{BERT}_{dim}$, where $|c|$ is the number of tokens in the context c, and BERT_{dim} is the embedding size of the BERT pre-trained model.

For each token in the context c, BERT generates an embedding by adding the the token embedding, segment embedding and position embedding (Devlin et al., 2019). Our model uses additional embeddings to encode information from the dialog history and user scenario, that serve as markers on the tokens present in the rule. Simpler variants of our marker embeddings have been found to be helpful in other conversation QA tasks (Qu et al., 2019).

Let the marker embeddings generated by our model for token i of the rule be r_m^i; the overall input embedding r_e^i is then given by:

$$r_e^i = r_t^i + r_s^i + r_p^i + r_m^i \quad (1)$$

where r_t^i, r_s^i and r_p^i correspond to the token, segement and position embeddings generated by BERT. The sequence of embeddings, r_e^1, r_e^2, $\ldots$, r_e^n, represent the embedded rule. We describe the construction of marker embeddings r_m in the next section.

3.2 Marker Embeddings

The marker embeddings are generated from two sources (i) Dialog *history* (ii) User *scenario*.

3.2.1 Dialog History Embedding:

When the model needs to generate a new follow-up question, knowing what has already been asked previously in the context of the *rule* clauses is important. Thus, indicating whether a token in a *rule* is also a part of a follow-up question in the *history* can be a rich source of information for the model. Further, encoding information about *how* the follow-up question pertaining to the clause was answered (yes or no) can also provide useful clues when the model reasons over the *rule*.

In order to determine the clause that a follow-up question relates to, we find the longest common subsequence between the sequence of tokens in the *rule* and the follow-up questions in the *history*. For each token in the *rule* that matched with a follow-up question, we include a trainable embedding corresponding to the answer of that follow-up question. Recall, that follow-up questions can contain only yes-no answers. Thus, each token of the rule that matches a follow-up question is assigned the corresponding *yes* or *no* marker embedding indicative of the answer to that question. Tokens from the rule that do not match any follow-up question are encoded with a special marker embedding ϕ. Additionally, embeddings that encode the sequence number of the follow-up question in dialog *history* are also included. We refer to these as *turn embeddings.*. The dialog history marker embedding r_h^i for token i in a rule is given by:

$$\begin{cases} r_h^i = r_{he}^i + r_{te}^i, & \text{if token } i \text{ is part of dialog history.} \\ r_h^i = \phi, & \text{otherwise} \end{cases} \tag{2}$$

where r_{he}^i and r_{te}^i correspond to the yes-no answer embedding and the turn embedding markers described above.

3.2.2 Scenario Embedding:

Recall that a *scenario* consists of sentences describing the user background or any information that a user chooses to provide (in addition to the question). Scenarios in the ShARC task were created by crowd-sourced workers who wrote them based on guiding text in the form of follow-up questions, but these questions were not included in dialog history when the dataset was released (Saeidi et al., 2018). However, these follow-up questions and their answers – termed as *evidence* are available as meta-data in the dataset. We utilize this *evidence* as a source of supervision to learn how to interpret scenarios in the context of the rules.

Scenario Interpretation: We use a BERT based classifier that takes as inputs a *scenario* and *rule*. For each token in the *rule*, the classifier predicts whether: (a) the token was part of an *evidence* follow-up question with an affirmative response, (b) the token was part of a evidence follow-up question with a negative response, (c) the token was not part of evidence follow-up questions. We trained this classifier on ShARC training data and get a macro F1 score of 0.47 on the development dataset. Note that the three classes predicted by the classifier correspond to the same setting under which marker embeddings from dialog history are generated. We thus, use the same yes-no answers embeddings that were used to represent dialog history, as the scenario embeddings.

Formally, the scenario embedding for a token i in the *rule* is given by:

$$\begin{cases} r_s^i = r_{he}^i, & \text{if token } i \text{ is predicted to be in evidence.} \\ r_s^i = \phi, & \text{otherwise} \end{cases} \tag{3}$$

where r_{he}^i is the embedding corresponding to predicted yes-no answer . The marker embeddings used in Equation 1 is thus given by:

$$r_m^i = r_h^i + r_s^i \tag{4}$$

The scenario embedding thus represents information from the *scenario* in the context of the *rule*.

Finally, the input context c augmented with marker embeddings for the tokens in the *rule* is fed to a pre-trained BERT model to generate the output embeddings (See Figure 2).

3.3 Answering and Follow-up question Generation

Classification: We use the BERT output embedding of the `[CLS]` token and feed it to a linear layer for 4-way classification (yes/no/irrelevant/follow-up). In case the classifier output is a *yes* or a *no*, these are returned as answers to the user question; if the predicted class is irrelevant, the user question is unanswerable given the *rule*, and if the classifier predicts the last class, i.e. follow-up, a follow-up question needs to be generated.

We generate the follow-up question in a two step process: the first step extracts a contiguous span from the rule, after which we generate a question from the extracted span.

Span Extraction: Similar to other BERT based QA models (Devlin et al., 2019), we pass the BERT embeddings of the tokens in the rule through two different linear layers to obtain probability distributions over start and end positions of the span. We choose the span for which the sum of these two probabilities is maximised.

Training: We use multi task learning to fine tune BERT and train the classification layers. The loss for training the model is the weighted sum of the classification loss and the span extraction loss. The span extraction loss is sum of cross entropy loss for the start and end positions. We obtain the ground truth span for calculating this loss by finding the longest common subsequence between the rule and the ground truth next utterance.

Decoding for Follow-up Question Generation: Once the span is extracted, we use the CopyNet sequence to sequence model (Gu et al., 2016) to generate a question from this span. The input to this model is concatenation of the *rule* r with

Dataset	#Train Instances	Class Distribution			
		Irr.	Yes	No	More
ShARC	21890	5.74	30.94	32.24	31.08
ShARC-Augmented	31506	22.41	27.09	28.11	22.39

Table 2: Dataset statistics for the original and augmented ShARC datasets. The ShARC-augmented dataset also has a more even split of class distributions.

special tokens around the extracted span, question q and the follow-up questions in the history h. We provide the previous follow-up questions as input so that the model can copy/learn how they were framed. The output of the model is the next follow-up question. While training this model, we use the ground truth spans as the input, and while testing we use the predicted span.

4 Experiments

Recent works (Jia and Liang, 2017; Niven and Kao, 2019) demonstrate how neural models often exploit spurious statistical clues in benchmark tasks to improve their performance. We train a baseline system using BERT (Devlin et al., 2019) with our copy decoder and assess model performance. We find that (i) when the model predicts that a question is *Irrelevant* , in all instances, the *history* and *scenario* were empty. (ii) In case the model answers the user question with a *Yes* or *No*, the prediction was same as the last follow-up answer 84% of the times (given that the last follow-up answer existed). In addition, as mentioned previously, as the number of turns in the history increase, the likelihood of the follow-up answer becoming the final answer understandably increases.

To reduce the effect of spurious patterns, we augment the dataset with additional examples as described in the next section.

4.1 Augmented ShARC dataset

We augment the original dataset by automatically generating training samples directed at addressing each of the spurious patterns identified:

1. In order to create new examples for the "irrelevent" class, we take an instance from the dataset with a non empty scenario, and create a new instance by replacing the associated rule with a randomly sampled ruled. Due to the nature of irrelevant questions, in the real world, it is unlikely any follow-up questions would have been asked – however, this fact induces a statistical correlation that neural models easily exploit. The additional samples we generate force the neural models to learn to infer a question's irrelevance by reasoning instead of relying on easy patterns.

2. A casual study of the dataset reveals that most rules either consist of conjunctive clauses or disjunctive clauses. As a result, in case of conjunctive clauses, if the answer to the last follow-up question was a negative, the answer to a user question will always be "no". Similarly, in case of disjunctive rule clauses, follow-up questions only to be generated as long as there's no affirmative response in the dialog history. As before, though these are characteristics expected in the real-world, these clues distract the model from learning how to reason. Thus, we create new training instances, by shuffling dialog history. We note that, this results in instances where a conjunctive clause may be negated in the first follow-up question, and yet the example contains contain subsequent follow-up questions which would appear unnecessary in the real-world. By balancing the answer class-distributions (yes/no/irrelevant/follow-up) in the augmented dataset we ensure the data is not biased towards preferring to generate follow-ups (Table 2).

We present our experiments using this data-set in the remaining part of this paper and refer to it as the ShARC-Augmented dataset.

4.2 Evaluation Metrics

We evaluate *UrcaNet* using the standard metrics of micro/macro classification accuracy and BLEU (Papineni et al., 2002) for follow-up question generation, as proposed in original task (Saeidi et al., 2018). We also report the combined metric (Comb.), defined as the product of macro-accuracy and BLEU-4 as introduced by (Zhong and Zettlemoyer, 2019).

4.3 Baselines

We compare *UrcaNet* against three other models:

- **Rule Based**: A rule based model that is designed to exploit the spurious clues on the original ShARC dataset.
- **Base Model**: A BERT based model that is augmented with our CopyNet decoder for generating follow-up questions.
- **E3**: A recent state of the art model (Zhong and Zettlemoyer, 2019) which first extracts

Model	ShARC-Augmented Dataset					Original Dataset				
	Micro-Accuracy	**Macro-Accuracy**	**BLEU 1**	**BLEU 4**	**Comb.**	**Micro-Accuracy**	**Macro-Accuracy**	**BLEU 1**	**BLEU 4**	**Comb.**
Rule Based	45.87	44.09	42.51	21.24	9.36	63.74	71.25	**63.97**	**47.78**	**34.04**
E3	67.31	68.16	56.93	36.64	24.97	66.08	72.65	57.23	43.42	31.54
Base Model	69.08	70.79	57.5	38.77	27.44	**68.37**	**73.68**	59.32	43.1	31.99
Our Model	**69.98**	**71.49**	**58.6**	**39.63**	**28.33**	65.90	71.72	61.22	45.76	32.81

Table 3: Performance of baselines and our model on both the original and the ShARC-augmented datasets. Results suggest that the current state-of-the art system E3 (Zhong and Zettlemoyer, 2019) relies more heavily on spurious clues in the dataset and suffers a steeper drop in performance on the ShARC-augmented dataset.

clauses from the *rule*. It then identifies which clauses are entailed by the conversation history using text overlap, and finally edits the clauses that have not been considered to create follow-up questions to the user.

4.4 Evaluation

Our experiments[3] aim to answer the following research questions:

1. How does the performance of models vary on the original and ShARC-augmented dataset?

2. Are the spurious patterns in the original dataset indeed problematic? Do they affect how well a model learns?

3. How meaningful is our modeling choice of using marker embeddings? How does each constituent marker embedding contribute to the overall model performance?

Model	Dataset	Irr.	More	Yes	No
Base Model	ShARC	95.65	63.7	65.92	70.63
	ShARC-Aug	100.0	52.15	63.25	73.86
E3	ShARC	96.38	60.50	65.92	69.45
	ShARC-Aug	98.83	43.35	67.50	67.50
Base Model (+) TE	ShARC	93.48	66.55	67.54	71.02
	ShARC-Aug	98.95	64.04	71.35	59.95
UrcaNet	ShARC	95.65	58.90	63.30	68.40
	ShARC-Aug	98.85	65.85	62.10	65.15

Table 4: Class-wise accuracy of models on the ShARC and ShARC-Augmented datasets. The performance of *UrcaNet* remains relatively stable across both datasets while the performances of the Base Model and E3 fluctuate, especially while predicting follow-ups (More).

Model Comparison: As can be seen in Table 3 we find that *UrcaNet* outperforms existing methods on both the ShARC-augmented dataset. It is interesting to note that the performance of the current state-of-the-art model (Zhong and Zettlemoyer, 2019), degrades considerably on the ShARC-augmented dataset indicating its possible reliance on spurious clues in the original dataset. Our rule based model performs very well on the original ShARC dataset.

Effect of Spurious Patterns: To investigate the effect of spurious pattern on a model's performance, consider the results shown in Table 4. We would expect that the models, that rely on clues based on turn-length and pick the last follow-up answer as the answer to the question, suffer degradation on the ShARC-augmented dataset where these patterns are reduced. As expected, we find that both the Base Model and E3 suffer from higher mis-classification rates especially while generating follow-up questions (More) while the performance of UrcaNet on both datasets remains consistent. Lastly, to further demonstrate that turn-lengths are indeed strong indicators that neural models can easily exploit, we update the Base Model to incorporate Turn Embeddings. Unsurprisingly, we find its performance on the original ShARC dataset is good; in fact reporting the highest scores amongst all models for three out of four classes.

4.4.1 Ablation Study

Model	Micro-Acc.	Macro-Acc.	BLEU 1	BLEU 4
Base Model	69.08	70.79	57.50	38.77
(+) History Emb.	**70.78**	72.08	56.18	37.92
(+) History Emb. (+) Turn Emb.	70.74	**72.33**	57.77	38.49
(+) History Emb. (+) Turn Emb. (+) Scenario Emb.	69.98	71.49	**58.6**	**39.63**

Table 5: Ablation study: Including history, turn and scenario embeddings to our BERT based baseline model incrementally improve performance.

To study the contributions made by each of our

[3] All tables refer to the "follow-up" class as "More" for ease of presentation.

marker embeddings, we conduct an ablation study. Table 5 shows the relative contributions made by each constituent marker embedding presented on the ShARC augmented dataset. As can be seen, including history, turn and scenario embeddings, each contribute towards incrementally improving overall model performance.

4.4.2 Error Analysis

To qualitatively assess the performance of *UrcaNet*, we conduct a study of 100 instances on the development set from the original ShARC dataset. We find that 20% errors were due to errors in the dataset (e.g. incorrect ground truth, inconsistent clauses in rules).

22% of errors made by our system were due to our model's incorrect interpretation of the user scenarios. Another 25% of the errors could be traced to incorrect reasoning over the rule (e.g. missing negations, detecting whether clauses were in conjunction or disjunction). 18% of the errors were due to poor framing of the follow-up question; these include grammatical errors such as well as posing questions for the wrong entity ("*Are you born early*" instead of "*Was your baby born early?*".

Most of the remaining 15% errors were because the model didn't ask a follow-up question at later turns and instead directly answered the question.

Model	Micro-Acc.	Macro-Acc.	BLEU 1	BLEU 4	Comb.
E3	**67.6**	**73.3**	54.1	38.7	28.37
UrcaNet *(Orig.)*	66.6	72.5	56.4	40.6	29.44
BaseModel (+) T.E. *(Orig.)*	65.1	71.2	**60.5**	**46.1**	**32.82**
UrcaNet *(Augm.)*	65.3	71.3	60.2	44.9	32.01

Table 6: Official Leaderboard scores: "*Orig.*" refers to models trained on the original ShARC dataset, while "*Augm.*" refers to models trained using the ShARC-augmented dataset. T.E. refers to the use of only turn embeddings.

4.4.3 Probing Study

To further illustrate how *UrcaNet* better resists spurious clues in the dataset, we show an example from the ShARC augmented development dataset (Table 7). E3 model relying on spurious patterns in the training data incorrectly predicts the last follow-up answer as the answer, while *UrcaNet* is able to reason and gives the correct answer. Even if we shuffle the order of follow-up questions in the dialogue history, this pattern repeats. E3 predicts the last follow-up answer as the answer, while UrcaNet gives the correct answer for all the possible permutations of the dialogue history in this instance

Rule	## Items that qualify for the zero rate You may be able to apply zero VAT when you sell the following to an eligible charity: * equipment for making ‘talking" books and newspapers * lifeboats and associated equipment, including fuel * medicine or ingredients for medicine * resuscitation training models
Scenario	
Question	Can I apply zero VAT to this item?
History	
System	Is it equipment for making âĂŸtalkingâĂŹ books and newspapers?
User	No
System	Are you selling medicine or ingredients for medicine?
User	Yes
System	Are you selling lifeboats and associated equipment, including fuel?
User	No
E3	No
UrcaNet	Yes

Table 7: Comparing the responses generated by E3 and UrcaNet on an example in ShARC-augmented development dataset.

4.4.4 Leader-Board Submission

The creators of the ShARC dataset retain a held-out set which is used to rank models on the public leaderboard.We submit three models (i) Our full model trained on the original ShARC dataset (ii) Our model without history and scenario embeddings but uses only turn embeddings. As demonstrated earlier, we expect this model could rely on the weaknesses identified in the held-out set by utilizing turn spurious clues from follow-up question in `history`. (iii) Our full model trained on the ShARC-augmented dataset. We expect that if our model is indeed reasoning and solving the task without relying on spurious clues in the dataset, this version should not suffer a significant drop in performance when tested on a held-out set that is likely to have those clues.

Table 6 shows the performance of these models on the unseen held-out set. As can be seen each of our models report a significantly higher BLEU score, with our best system outperforming official state of the art submission (E3) by approximately 16%. Further, it is interesting to note that our model trained on the ShARC-augmented dataset does better than one trained on the original dataset which suggests that our full model is solving the task meaningfully and not relying on spurious clues. Unsurprisingly, the use of turn embeddings with the base model, appears to perform the best by being able to make use of spurious correlations present in the original dataset.

5 Conclusion

In this paper we show how the existing neural models exploit spurious patterns that exist in the data for the ShARC task - a conversation QA that requires reasoning over rules expressed in natural language. We demonstrate how existing models can exploit spurious patterns in such conversational QA datasets and introduce an augmented version of the ShARC dataset that discourages a model from exploiting such spurious clues. We also present a simple yet effective model, *UrcaNet*, that learns embedding representation from the dialog history, dialog turns, and the history of past follow-up question and answer pairs. The network generate intermediate representations which is input to a copy decoder to generate a follow-up question. *UrcaNet* outperforms existing systems on both the original ShARC corpus and the augmented ShARC corpus.

6 Acknowledgements

We would like to thank Nikhil Verma for his help in running BERT-QA based models on inputs with large contexts.

References

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *CoRR*, abs/1908.07898.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2021–2031.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611.

Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. Attending to future tokens for bidirectional sequence generation. *arXiv preprint arXiv:1908.05915*.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4658–4664.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. Bert with history answer embedding for conversational question answering. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 1133–1136, New York, NY, USA. ACM.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *TACL*, 7:249–266.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2087–2097.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal trigger sequences for attacking and analyzing nlp. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

Victor Zhong and Luke Zettlemoyer. 2019. E3: entailment-driven extracting and editing for conversational machine reading. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2310–2320.