Combining Question Answering with Dialog Systems

Wentseng Chang
University of Stuttgart
st164021

Tornike Tsereteli
University of Stuttgart
st170794

Abstract

Conversational agents are used in many consumer applications. Currently, however, these agents struggle to answer complex questions and do not perform well in long conversations with inter-connected context. This report aims to give an overview of current approaches to solve these issues. Most approaches are build on massive Language Models, specifically BERT (Devlin et al., 2018), that achieve state-of-the-art performance on many Machine Comprehension tasks (Rajpurkar et al., 2016), but fail at causal inference (Devlin et al., 2019). This report also gives insights on current limitations and possible future work.

1 Introduction

Recent advances in Natural Language Processing (NLP) have brought the goal of developing human-like chatbots a step closer. Specifically, chatbots that are able to answer complicated questions (e.g. in the legal or medical domains) have massive potential. A simple modern dialog system is made up of multiple models, such as speech recognition, language understanding, state tracking, dialog policy, language generation, and synthesis. However, Conversational Question Answering (ConvQA) systems struggle to answer complex questions in long conversations where context is needed to fulfill the information needs.

1.1 Question Answering

Question Answering (QA) is a subtask of Natural Language Understanding (NLU), which tests the reasoning ability of NLU systems and is used in many applications (e.g. search engine). The goal can be defined as to answer a question based on a given resource. This is typically done end-to-end, without special information extraction methods for either the question or the resource. The resource can be structured data (e.g. Knowledge Graphs (KG)), semi-structured data (e.g. Wikipedia) or unstructured data (e.g. raw text). QA systems usually focus on answering a given question (single-turn) with no regard to previous questions or answers. Multi-turn QA systems also take QA history into consideration. The answer the system responds with is either given as a span from the resource (e.g. copy a sentence that answers the question) or generated from relevant parts of the resource. While research on single-turn QA is improving, multi-turn QA introduces some problems. Most notably, the challenge arises on how to make use of history to answer the most recent question precisely.

1.2 Dialog Modeling

Dialog systems are increasing in popularity as the underlying technologies improve. A dialog system, or conversational agent, is a program intended to converse with a human in natural language. There are three branches of systems: social bots, task-oriented bots, and QA bots (Qu et al., 2019a). The former tries to mimic human behavior in conversation and extend the duration of the dialog. The main purpose for such a system can be seen as entertainment, while the research on such systems helps to improve the flow and naturalness of machine dialog. A task-oriented system can have clearly defined goals, such as executing functions or simply information retrieval. Lastly, QA bots answer questions (e.g. e-doctor). A dialog system can not only entail a QA system, but also a Question-Generating system in order to ask necessary clarification questions.

1.3 Conversational QA

A natural combination of QA and Dialog Modeling is Conversatioanl Search (ConvSearch). There are three main ConvSearch interaction forms: the user asks questions, the system asks questions, and both ask questions. In the first form, the system

needs to be able to answer questions with cross-references over multiple turns. For the second form, the systems need to be able to understand cross-referencing and ask questions based on the conversational history. And in the last form, both of these need to function interchangibly. An example can be seen in Figure 1. The user initially asks a question, the system then responds with a clarification question. Finally, the system answers the initial question with respect to the conversation history. A simplified form of ConvSearch is Conversational QA (ConvQA) as ConvQA systems do not focus on asking (clarification questions) proactively (Qu et al., 2019a).

1.4 Research Question

This report on ConvQA systems aims to answer the following research questions:

- How is conversational history encoded in ConvQA systems?
- How is turn-tracking solved?
- What are different methods of encoding conversational history?
- How are relevant turns and history found?

2 Conversational Question Answering

This section outlines Conversational Question Answering with a focus on the above research questions. This report strictly focuses on textual systems and we do not cover the entire scope of ConvQA.

2.1 Datasets & Related Tasks

Although there are only a limited number of datasets that are specifically designed for training ConvQA systems, generating synthetic datasets from existing QA datasets is simpler than for many other NLP tasks. We describe a few datasets and benchmarks in the following subsections.

2.1.1 CoQA

The Conversational Question Answering dataset (CoQA) (Reddy et al., 2018) contains 127,000 questions with answers collected from 8000 conversations. It features conversational questions and free-form text answers, all the passages are collected from seven diverse domains.

2.1.2 QuAC

The Question Answering In Context dataset (QuAC) (Choi et al., 2018) contains 98k QA pairs, which are based on conversations between a student and a teacher. Given a paragraph, the student asks questions and the teacher gives answers in the form of spans from the text. QuAC is more challenging than CoQA because the questions cannot be answered by simply encoding the previous question. A more complex understanding of the context is required to perform well on this dataset.

2.1.3 ShARC task

The ShARC task (Saeidi et al., 2018), a benchmark of UrcaNet (Sharma et al., 2019), requires a system to answer user questions referring to related rules or policies in natural language text. The model needs to not only understand questions, but also comprehend them with applicable rules and provide a yes/no answers to the user. Figure 1 shows an example.

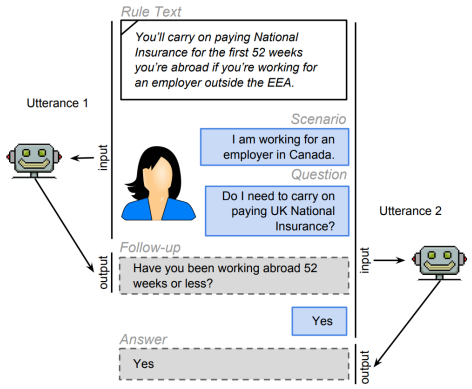

Figure 1: QA dialog with rule interpretation and generation of clarification questions. Based on the history, scenario, background knowledge (Canada is not in the EEA) leads to the answer "Yes". Source: (Saeidi et al., 2018).

2.1.4 Sequential Instruction Understanding

Sequential Instruction Understanding (SIU) is a task of understanding a sequence of natural language instructions. This task is conducted by FlowQA (Huang et al., 2018) and FlowDelta (Yeh and Chen, 2019). Given a sequence of instructions, where the meaning of each instruction may depend on the entire history and world state, the task is to understand the instructions and modify the world accordingly. A simplified version is shown in Figure 2.

In the *World-State N-1*, there are three men, one

with a purple shirt, another with a blue shirt and purple hat, and the last with an orange shirt and yellow hat. Given a question (intruction) "He took the blue guy's hat." and the context of *World-State N-1* as the input, the system output the answer and move the purple hat from the man with the blue shirt to the man with the purple shirt. The final world state will be updated as *World-State N* following the instruction.

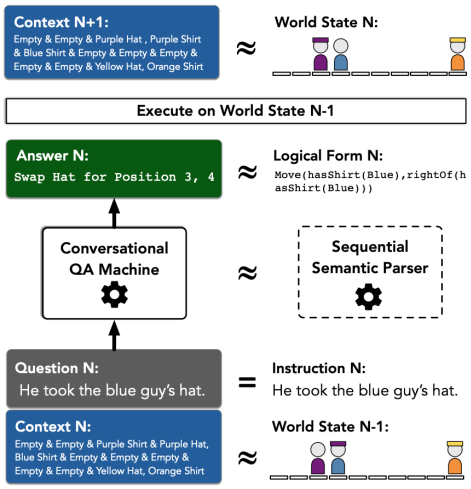

Figure 2: An Illustration of the SIU task. This simplified version is from (Long et al., 2016) and the graph is from (Huang et al., 2018)

2.2 ConvQA Architecture Overview (Input)

A ConvQA system typically requires the following elements: a resource, a question-answer pair(s), a QA system, and a prediction system (e.g. classifier).

ConvQA systems largely depend on encoding the history of the conversation to make appropriate decisions and give the right responses during interactions. Each utterance, whether from the user or from the system, can be viewed as a *turn*. The turn acts as a counter and a reference index, while the set of question-answer pairs of previous turns makes up the *conversation history*. The input of the system is made up of the resource and question-answer pairs, which get encoded into vector representations. Training procedures combine the input, turn and history information in different ways. Attention mechanisms can help the model focus on relevant information during training as well as during the prediction steps. We explain each of these elements in the following subsections.

The input for ConvQA systems is typically a question-answer pair, conversation history, a paragraph, or a knowledge base. The input is encoded into vector space representations as vector embeddings contain semantic knowledge. An underlying commonality in most recent works in NLP is the choice of pretrained embeddings, which is either GloVe (Pennington et al., 2014) or BERT (Devlin et al., 2019).

2.2.1 BERT

The common underlying context embedding used in most of the methods described in this report is the BERT model (Devlin et al., 2019). Figure 3 shows the standard input module. The representation is the sum of the token, segement, and position embeddings. BERT makes use of WordPiece (Johnson et al., 2017) for its token embeddings. There are two special tokens: the first token of a sentence [CLS] and the sentence seperator token [SEP]. The segment embeddings are either 0 or 1, signaling tokens that belong to either sentence A or sentence B. This input embedding is part of the training procedure for next sentence prediction. The position embeddings are used to mark the token positions in the input sentence.

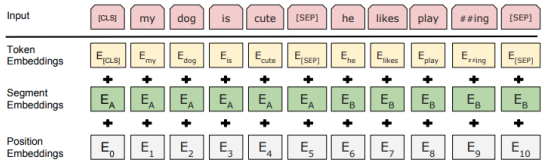

Figure 3: Standard BERT (Devlin et al., 2019) input representation made up of the sum of the token, segment, and position embeddings.

2.2.2 Turn Tracking

One of the major challenges of ConvQA systems is being able to answer the current question. Turn-tracking is one of a combination of possible solutions. Tracking the current turn, or question-response pair, has multiple benefits.

First, turn information is frequently used to mark conversational history (Huang et al., 2018; Yeh and Chen, 2019; Chen et al., 2019; Qu et al., 2019a,b; Sharma et al., 2019; Christmann et al., 2019; Gao et al., 2019). We further describe how history is encoded in Section 2.2.3.

Second, users may have varying information needs during the conversation. Thus, questions may have the dialog behaviors of *topic shift* and *topic return* (Reddy et al., 2018). A topic shift implies that the user's current information need

shifts to a new topic, while a topic return suggests that the information need reverts back to a previous question. Information on turns allows mapping different turns to topics.

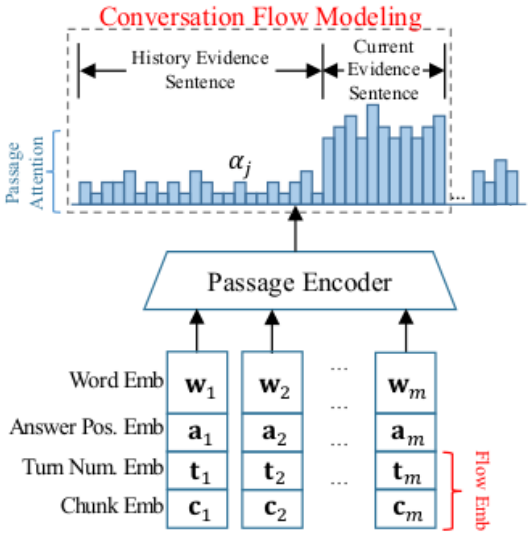

Figure 4: Flow embedding proposed by Gao et al. (2019).

Third, information on turns helps generate more natural conversations (*flow*) (Gao et al., 2019; Chen et al., 2019; Huang et al., 2018). Gao et al. (2019) introduce a method for coreference alignment and conversation flow modeling generating inter-connected questions based on passages. The former explicitly aligns coreferent mentions with pronominal references, while the latter starts questioning on the first sentences of a passage and shifts focus onto later parts. This is achieved via a flow embedding that encodes the correlations between the number of turns and narrative structure of passages. Figure 4 shows how the turn number is embedded into the passage encoder to generate the flow embedding.

Fourth, the number of turns can be an indicator which dialog act is most relevant for the next turn (e.g. follow-up question prediction). However, Sharma et al. (2019) argue that models depend too heavily on these cues, which may not reflect real-world conversations. The likelihood of a follow-up question decreases as the number of turns increases.

Fifth, it provides knowledge on how far the conversation has advanced. While more complex questions may require more turns, or longer interactions, a goal parameter could be to minimize the number of turns needed to answer user queries.

Finally, turn information is used in the KG-based method introduced by Christmann et al. (2019) to expand the graph as the conversation develops. Details of the method are described in Section 2.2.5.

2.2.3 Conversational History

Knowledge of conversational history is a fundamental part of ConvQA. Many different approaches have been proposed for encoding history into systems in order to generate the right response (e.g. follow-up or answer) (Sharma et al., 2019), to respond to inputs with coreferences or ellipsis (missing context) (Chen et al., 2019; Gao et al., 2019), or to better understand context (Qu et al., 2019a,b). Simply prepending previous question-answer pairs to the current input (Reddy et al., 2018; Zhu et al., 2018) or embedding answer locations into the passage (Choi et al., 2018; Yatskar, 2019) may not be sufficient in dealing with complex and long conversations (Qu et al., 2019b). Furthermore, ignoring previous *reasoning* processes performed by the model may be lost information (Huang et al., 2018). Multiple new approaches are described below.

Huang et al. (2018) introduce FlowQA, a method to implicitly model the context representations by deeply integrating the latent semantics of the conversation history. The method uses several context integration layer (BiLSTM) outputs from previous questions, which locate answer candidates, to the current question answering process. Yeh and Chen (2019) expand on FlowQA with FlowDelta by explicitly modeling information gain to allow the model to focus on more informative cues. This is shown in Figure 5. FlowDelta measures the change in flow from previous hidden representations and assumes that a large difference between previous and current hidden states signals different topics. FlowDelta can be integrated into the standard BERT architecture in two locations. First, before the final prediction layer which calculates the span start and end tokens. Second, in the last BERT layer to model dialog history, which becomes a concatenation of the hidden layer, self-attention and FlowDelta.

Li et al. (2019) first add attention calculations to the input and combine the input question with the conversational history into a context vector. Figure 6 shows the encoding process. We describe the attention mechanism in Section 2.2.4.

As some parts of the conversation history may not be relevant for the current information need, encoding the entire history may introduce noise.

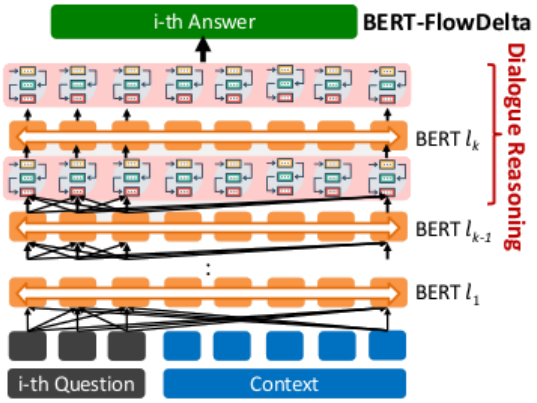

Figure 5: FlowDelta mechanism (Yeh and Chen, 2019) integration into the standard BERT architecture. FlowDelta is added before and after BERT l_k layer.

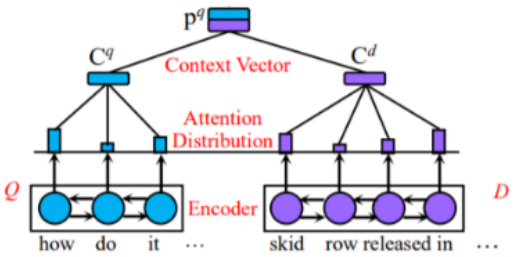

Figure 6: Input encoding process described in Li et al. (2019). C^q is the current question and C^d is its history.

Thus, encoding only informative parts can help the model focus better. Qu et al. (2019a) introduce a simple and efficient method for encoding only the last *j* turns. It is especially tailored for integrating the encoding into the BERT architecture in a natural way. More specifically, the previous *j* turns are chosen to be included in the embedding. Tokens are given extra embedding information, which are two learned *history answer embeddings* (HAE) that denote whether a token is part of history answers or not. Figure 7 shows the additional HAE layer to the standard BERT layers.

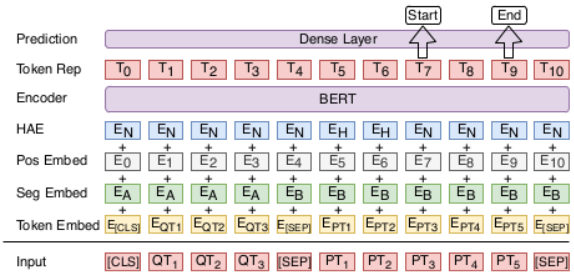

Figure 7: Architecture of the ConvQA model (Qu et al., 2019a) with the standard BERT layers and an additional history answer embedding (HAE) layer.

Sharma et al. (2019) use a similar approach by adding informative embedding layers to the original BERT model. They additionally add turn, history and scenario embeddings. Figure 8 shows the three additional layers. In their setup, the input also contains a set of rules, a given scenario, expressed in natural language (e.g. "Eligible if RULE."). The turn is encoded as either a number for the turn in which a token occurs or empty. The history is also encoded as a simple "yes" or "no" marker if the user responded with either of those answers, or it also left empty. The scenario additionally either marks whether a token is part of the rule or not or is empty.

2.2.4 Attention

The attention mechanism (Bahdanau et al., 2014; Luong et al., 2015), an ubiquitous method now in NLP, is the heart of the BERT model. It allows the model to learn to "focus" on imporant words in the input through attention weights. In the ConvQA setting, attention is used to focus on relevant context and history turns (Li et al., 2019; Pan et al., 2019; Chen et al., 2019; Huang et al., 2018; Yeh and Chen, 2019; Qu et al., 2019a,b), to generate answers (Gao et al., 2019; Sharma et al., 2019), and to predict dialog acts (Qu et al., 2019b).

FlowQA (Huang et al., 2018) uses the entire hidden representations by encoding the previous question-answer pairs, which may entail clues on conversational context. It makes use of the attention mechansim, first, on the question (to focus on context words) and then on the context to enhance context word embeddings.

As the previous *j* turns may not be informative enough or may entail noisy information if *j* is too large, choosing the right history turns from the complete history is important. (Qu et al., 2019b) extend on HAE embeddings (Qu et al., 2019a) by also incorporating a history attention module (Vaswani et al., 2017) that helps select the right history turns. Furthermore, the history attention module allows a more fine-grained prediction of which dialog act should follow as well as a prediction for the span of the answer. Figure 9 shows how the history attention module is used for both of these tasks. The architecture combines the history attention module on the sequence-level with the contextualized representation on the token-level. Section 2.3 describes the different prediction tasks in more detail.

Gao et al. (2019) use attention on the conversation-level to align non-pronominal coref-

Figure 8: Architecture of the UrcaNet model (Sharma et al., 2019) with the standard BERT layers and additional turn, history, and scenario embedding layers. In the top left corner is a dialog act prediction mechanism and in the top right a copy mechanism that generates questions.

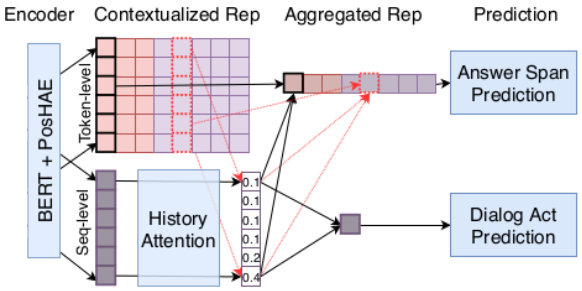

Figure 9: Histoy attention module proposed by Qu et al. (2019b) in combination with history answer embeddings (HAE) (Qu et al., 2019a).

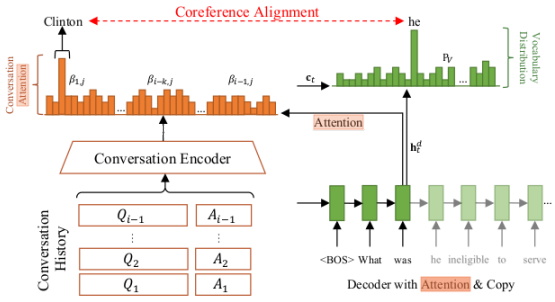

Figure 10: Coreference alignment using attention proposed by Gao et al. (2019).

erent mentions with pronominal reference words. Figure 10 shows how the attention mechanism over the input and conversation is used to align the most likely (personal) pronoun.

2.2.5 Knowledge Graph-based

An alternative to encoding conversational history is when the resource is a Knowledge Base. A KG can be thought of as a context span, with relevant nodes representing the context. Christmann et al. (2019) propse a graph-traversal method in which the first question narrows down the KG into a subgraph and follow-up questions expand the subgraph with relevant nodes. Figure 11 shows an example of the graph-traversal. The top subgraph at turn 0 is generated upon an initial question. The subgraphs at turns 1-6 are generated through the context expansion.

Follow-up questions can often be incomplete and ungrammatical or have missing entities and unspecified contexts. Context subgraphs can help infer missing or wrong pieces by looking at 1-hop or 2-hop relevant nodes. By scoring node likelihoods, a potential best candidate node can effectively be picked.

2.3 Prediction

The answer is the output of the QA system. The formats of the answer can be categorized as following: copy answer, generated answer, yes/no answer or unknown answer.

A copy answer means that the answer is directly copied from the context. This means the question is rather straightforward. However, some answers need to be deducted from the context. The machine should be able to extract the possible candidate information and generate it into understandable phrases or sentences. The yes/no answer means the answer is yes or no. If the answer cannot be found from the paragraph, it is unanswerable.

There are several mechanisms used to predict or generate the answer spans. Most common and essential methods are answer span prediction and generation. The answer span selection method is proposed by (Huang et al., 2017) which is used in

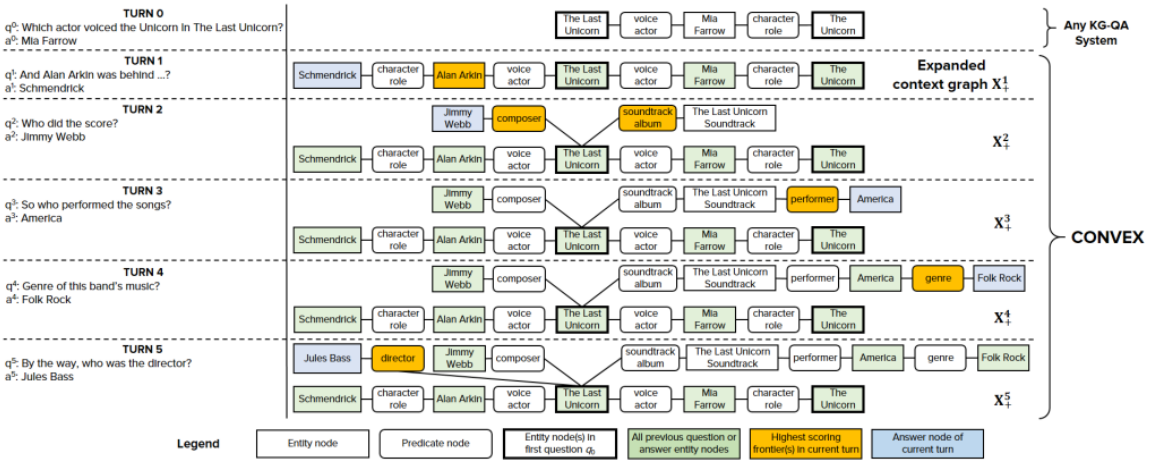

Figure 11: A typical conversation example using the CONVEX algorithm (Christmann et al., 2019). Context expansion occurs at each turn after turn 0. The answers are predicted using the most likely nodes in the current turn's context subgraph.

FlowQA (Huang et al., 2018) and related works. (Cite answer generation methods) Besides, there are some other optimal ways to predict the answer more accurately, such as classification or dialog prediction, which are used in Attentive BERT model (Qu et al., 2019b).

2.3.1 Answer Span Prediction

Answer span prediction computes the maximum likelihood of begin and the end tokens. The computation is slightly different for each model.

In FlowQA (Huang et al., 2018), the answer span selection method estimates the start and the end probabilities $P^S_{i,j}$, $P^E_{i,j}$ from the both j-th context token and referring i-th question token.

$$P^S_{i,j} \propto exp([c^h_{i,j}]^T W_S p_i) \tag{1}$$

where p_i history-aware question vector from the final layer and W_S is a trainable square matrix.

In order to apply the information of the span start into the span end, the final vector p_i and the sum of span start and last layer of context token vector are combined through GRU (Cho et al., 2014).

$$\widetilde{p}_i = GRU(p_i, \sum_{i,j} P^S_{i,j} c^h_{i,j}) \tag{2}$$

where h in $c^h_{i,j}$ is the last layer from the training step. The probability of the end span is computed as:

$$P^E_{i,j} \propto exp([c^h_{i,j}]^T W_E \widetilde{p}_i) \tag{3}$$

where W_E is another trainable square matrix. The final output is considered as candidate answer spans with highest start and end probability.

ConvQA (Qu et al., 2019b) takes the contextualized token representation produced from the history attentive network. The model learns two sets of parameters, a begin vector and an end vector, to map a token representation to a logit. In the end, they use the softmax function to compute the probabilities across all token sequences and predict the answer span by computing the probability of the begin and end tokens. The probabilities are evaluated by the cross entropy loss. It's worth noting that for unanswerable questions, the token "CANNOTANSWER" is appended into each passage in QuAC dataset. The model learns to predict this answer span if it believes the question is not answerable.

2.3.2 Optimization: Classifier

In the FlowQA model, a classifier is suggested as a threshold to decide whether a question is answerable or not and set before the answer span prediction process. To address unanswerable questions, the probability of "no answer" is computed as:

$$P^\phi_i \propto exp([\sum_{j=1}^{m} c^h_{i,j}; \max_j c^h_{i,j}]^T W p_i) \tag{4}$$

Only an answerable question can be processed to predict its answer span.

Furthermore, GraphFlow (Chen et al., 2019) suggests training different classifiers based on different types of QA pairs in the benchmarks CoQA and QuAC. For CoQA, the authors use a multiclass classifier with four categories: "unknown",

”yes”, ”no” and ”other”. Only the question type ”other” processes text span prediction. For QuAC, they train three separate classifiers: a binary classification task (answerable/unanswerable) and two multi-class classifiers (yes/no and follow-up). Only answerable questions process text span prediction.

Another classifier is used for dialog act prediction suggested by Qu et al. (2019b). Dialog act prediction uses multi-task learning to improve the prediction accuracy by revealing users intents. They first aggregate sequence representations for a training instance. Two sets of parameters are learned to predict the dialog act such as affirmation (yes/no), continuation (follow up, maybe follow up, don't follow up) or unanswerable flag annotated in QuAC. The dialog act prediction is independent from the information of each single training instance (current question, passage, conversation history) without history encoding.

2.3.3 Answer Generation

There are several purposes for answer generation. One reason is to build a natural language answer for conversational dialog, such as using language model, natural language generation or copy mechanism. Another reason is the question cannot directly be answered without rules or conditions such as ShARC task.

The model UrcaNet (Sharma et al., 2019) processes ShARC task questions by using the following steps: classification, span extraction, training, and decoding for follow-up question generation. In the classification step, the question is predicted from four categories: ”yes”, ”no”, ”irrelevant” or ”follow-up”. If the question is answerable, the system outputs a yes/no answer. If the question is considered irrelevant, the system outputs ”unanswerable”.

If the question is categorized as ”follow-up”, the question has to be generated. The system then applies the next steps: first, it extracts a contiguous span from the rules. Afterwards, the system generates a question from the extracted span to collect information in the next dialog turn by using a sequence-to-sequence model CopyNet (Gu et al., 2016). The input of the CopyNet is a concatenation of the extracted span from the rule tokens, question, and the follow-up question in the history. The output is the next follow-up question.

2.4 Conversational Question Generation

Conversational Question Generation (CQG) aims to develop an intelligent agent drive both current passage and conversation history, understand what the question has been asked so far and generate another meaningful question in next turn to make a coherent conversation. This task is proposed to use a reinforcement learning mechanism.

The Reinforced Dynamic Reasoning (ReDR) network (Pan et al., 2019) uses a general encoder-decoder framework, but incorporates a reasoning procedure in a dynamic manner to better understand what has been asked (previous questions) and what to ask next about the passage (i.e. generate new question). It encodes the knowledge of the passage and the conversation history based on a co-attention mechanism. It then updates the encoding representation dynamically based on a soft decision maker to generate a coherent question. The answer quality is evaluated by another QA system DrQA (Chen et al., 2017), which can be seen as a proxy for real human feedback, as rewards not only fine-tune the ReDR model, but also encourage it to produce meaningful and interesting questions.

Answer-Supervised Question Reformulation (ASQR) (Li et al., 2019) is another CQG system focusing on enhancing conversational machine comprehension with reinforcement learning. This paper points out that the existing question formulation models are trained only using supervised question labels marked by annotators without considering any feedback information from answers, which is also called the ”teacher forcing” mechanism. There are several drawbacks of using annotated labels-supervised training: (1) Minority: Human annotated data is rather small because of the limitation of resources and funds. (2) Errors: Fatal errors may happen in annotated data. (3) Unmet requirements: Because current systems do not consider feedback information from subsequent functions, the reformulate questions depend on question labels and not on gold answers. The ASQR model proposes to utilize a pointer-copy-based question reformulation model as an agent, takes an action to predict the next word, and observes a reward for the whole sentence state after generating the end-of-sequence token.

In detail, the state for the whole sentence is composed of continuous actions and ends with the end-of-sequence (EOS) signal. The agent only observes a reward for the whole sentence state after generat-

ing the EOS token, which is quite different from the teacher forcing models. The reward is the similarity score between the gold answer and the predicted answer obtained by feeding the whole sentence state to a single-turn machine comprehension model.

3 Discussion

This subsection compares each model in different tasks, including the discussion of experiment results.

3.1 Experiments

Most of the models test on the QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2018) datasets. Both of which follow the conversational setting, however, QuAC asked crowd annotators to highlight answer spans from the context, while CoQA asked for free text answers to encourage natural dialog. The main evaluation metric is F_1, the harmonic mean of precision and recall at the word level. In QuAC, there are two more evaluation metrics: Human equivalence score (HEQ) in question accuracy (HEQ-Q) and dialog accuracy (HEQ-D). HEQ-Q is the accuracy of each question, where the answer is considered correct when the model's F_1 score is higher than the average human F_1 score. HEQ-D is accuracy of each dialog, it is considered correct if all the questions in the dialog satisfy HEQ. (Choi et al., 2018)

3.2 Results & Comparison

The results of conversational question answering models in CoQA and QuAC experiments in test set are shown in Table 1.

3.2.1 Conversational Question Answering

Existing approaches use single-turn machine comprehension models and augment the current question and context with the previous questions and answers. However, this strategy only provides a partial solution and ignores previous reasoning processes performed by the model.

As we see in the Table 1. FlowQA (Huang et al., 2018) propose a "FLOW mechanism", which can be viewed as stacking single-turn QA models along the dialog progression (i.e., the question turns) and building information flow along the dialog. Compared to the previous baseline model such as BiDAF++ (Choi et al., 2018) or vanilla BERT (Devlin et al., 2018), the results of FlowQA show significant improvement.

After FlowQA, there are several works that follow the flow mechanism. Except for the performance, the biggest problem of the FlowQA is the long training time (56.8 hours) even though they have integrated an Integration-Flow (IF) mechanism to process context and question layer in parallel. GraphFlow (Chen et al., 2019) point out that IF mechanism is inefficient because the results of the previous reasoning processes are not incorporated into the current reasoning process. They propose to use a Graph Neural Network (GNN) to not only shorten the training time but also raise the performance. Another approach uses BERT instead of the Flow mechanism, which is proposed by (Qu et al., 2019a). According to their data record, although the vanilla BERT only needs 6.8 hours for training, the performance is not good enough. They introduce a history answer embedding (HAE) layer into BERT which learns two unique history answer embeddings and denotes whether a token is part of history answer or not. Their results boost the performance of BERT and are comparable to Flow while requiring a much shorter training time. Moreover, they replace HAE layer with PosHAE (HAE with relative position information of a history turn) to fix the failure of considering the position of a history utterance in the dialog. They propose a history attention mechanism ConvQA(HAM) that learns to attend to all available history turns with different weights (i.e. soft-selection) to select useful history information and use dialog act prediction to raise the accuracy in answer prediction.

Flowdelta (Yeh and Chen, 2019) is a mechanism to explicitly model information gain in flow-based reasoning for multi-turn dialog, which can also be incorporated in other machine comprehension models. The authors point out that the Flow operation is expected to incorporate salient information in an implicit manner because the learned representation captured by Flow would change during multi-turn questions. It is unsure whether such change correlates well with the current answer or not. By implementing FlowDelta, which focuses on modeling the difference between the learned context representations in multi-turn dialog, it improves the performance both in FlowDeltaQA (77.6% in CoQA and 64.8% in QuAC development set) and BERT-FlowDelta model (77.7% in CoQA and 67.8% in QuAC test set). The FlowDeltaQA only test on the development set.

The ASQR (Li et al., 2019) model shows rather

	CoQA	QuAC	HEQ-Q	HEQ-D	Train Time(h)
BiDAF++	67.8	60.1	54.8	4.0	-
BERT	-	54.4	48.9	2.9	6.8
FLOW	75.0	64.1	59.6	5.8	56.8
GraphFlow	77.3	64.9	60.3	5.1	-
BERT+HAE	-	62.4	57.8	5.1	10.1
ConvQA(HAM)	-	64.4	60.2	6.1	-
BERT-FlowDelta	77.7	67.8	63.6	12.1	-
ASQR	-	53.7	48.1	2.9	-
Human	88.8	80.8	100	100	-

Table 1: Experiment results of CoQA and QuAC in test set. Column CoQA and QuAC are F_1 scores(%). HEQ-Q is the accuracy of each question. HEQ-D is the accuracy of each dialog. The unit of train time is hour.

poor performance. While the authors aim at proving the effectiveness of answer-supervised question reformulation model, they point out that only question reformulation cannot reach the best performance. Question turns, scenario transformation, and answer lapse are all important factors for single-turn MC tasks. Besides, since ASQR uses a reinforcement learning technique. The feedback from a single-turn MC model seems to not be good enough for the task. It does not give appropriate answers occasionally trained by original QuAC dataset and limit the performance improvement of the model. Overall, to get correct and appropriate feedback and combine question reformulation with implicit conversational models in order to have better integration of conversational information is a main future work of this model.

3.2.2 Other Tasks

Sequential Instruction Understanding Task
Sequential Instruction Understanding (SIU) is a task of understanding a sequence of natural language instructions. The introduction of the task has been mentioned in Section 2.1.4 and it is conducted by FlowQA and FlowDelta with SCONE (Long et al., 2016) in three domain: Scene, Tangrams, and Alchemy.

	Scene	Tangrams	Alchemy
Long et al. (2016)	14.7	27.6	52.3
FlowQA	74.5	72.3	76.4
- Flow	58.2	67.9	74.1
FlowDelta	75.1	72.5	76.1

Table 2: Test accuracy for SCONE test (in %). -Flow is an ablation study shows the effectiveness of mechanism.

Each domain has different environment settinsg. Models are compared with previous semantic parsing approaches (Long et al., 2016) which map each instruction into a logical form or into actions, then execute the logical form to update the world state. The performance is evaluated by the correctness of the final world state after five instructions.

SIU has been studied in the knowledge base setting and framed as a semantic parsing problem. Recent datasets enabled the study in text settings. In conversational MC, the task focuses on reasoning about the context based on conversation history. In Table 2, FlowQA outperforms by the baseline and the ablation study shows the effectiveness of the Flow mechanism. The result of FlowDelta outperforms in Scene and Tangrams, but has small drop in Alchemy. The authors claim that the previous dialog history is less important in this domain so that the FlowDelta does not add much improvement.

Augmented ShARC Task: Spurious Pattern
UrcaNet (Sharma et al., 2019) is the model applied on the ShARC task. The experiment of UrcaNet not only outperforms the benchmark, but also conducts a spurious pattern study. According to the paper, their recent work demonstrates how neural models often exploit spurious statistical clues in the benchmark tasks to improve their performance. To reduce the effect of spurious patterns, they augment the dataset with additional examples by automatically generating training samples directed at addressing each identified spurious patterns.

The augmented ShARC task prevents spurious patterns of questions, which rely on clue-based in turn-length and pick the last follow-up answer as the predicted answer. In Table 3, the accuracy of Base Model and E3 decrease more in miss-

	Dataset	Irrelevent	More	Yes	No
Base Model	ShARC	95.65	63.7	65.92	70.63
	Aug-ShARC	100.0	52.15	63.25	73.86
E3	ShARC	96.38	**60.50**	65.92	69.45
	Aug-ShARC	98.83	**43.35**	67.50	67.50
UrcaNet	ShARC	95.65	**58.90**	63.30	68.40
	Aug-ShARC	98.85	**65.85**	62.10	65.15

Table 3: Accuracy for ShARC and Augmented ShARC task (in %) in four question categories. The performance of the UrcaNet is relatively stable between original and augmented tasks especially in the "More" category. E3 model is the state-of-the-art model in the paper.

classification rates especially in generating follow-up questions, while UrcaNet remain consistent.

3.2.3 Conversational Question Generation

Question generation works use BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) to measure the relevance between generated and the ground truth questions. Furthermore, ReDR (Pan et al., 2019) also evaluate the diversity of the generated questions, which are Dist-n (n=1,2) (Li et al., 2015) and Ent-n (n=4) (Zhang et al., 2018) metrics. Dist-n is the proportion of the unique n-grams over the total number of n-grams in the generated questions for all passages; while Ent-n reflects how evenly the n-gram distribution is over all generated questions. The larger the score the more relevant or diverse the generated questions are.

Table 4 shows the CQG experiment results. Each paper has a slightly different experiment setting and baseline models.

In ReDR and FlowNet experiments, both use questions in CoQA dataset as gold targets to compare with their model. In ReDR experiment, they adopt the 4th smoothing technique of BLEU for short text generation. The baseline of ReDR experiment models are shown below:

- **Seq2Seq Model** (Sutskever et al., 2014) A basic encoder-decoder sequence learning system which is widely used in dialog generation. They concatenate the rationale and the conversation history as the input sequence for the setting.
- **NQG** (Du et al., 2017) A strong attention-based neural network approach for question generation task. The input setting is the same as Seq2Seq Model.

In FlowNet experiment, they first extract the CoQA dataset as a quadruple of passage, question, answer, and conversation history (previous n-turns of QA pairs). Then filter out the QA pairs with "yes/no" or "unknown" as answers (about 28.7%), because there is little information to generate the information. Finally, the selected datasets are split into train, validations and test sets. Besides, they locate the extractive spans from the passage with maximum F1 score answers as answer position embedding, choose long turn (n=3) conversation history, label the evidence sentences, and employ the teacher-forcing training. The baseline of FlowNet experiment models are shown below:

- **NQG** (Du and Cardie, 2018) A pointer-generator network that takes current answer features concatenated with the word embeddings during encoding.
- **CorefNet** A coreference alignment model which is proposed by the authors using coreference to refer back. This alignment model tells the docoder to look at the correct non-pronominal coreferent mention in the conversation attention distribution to produce the pronominal reference word.
- **CorefNet** A coreference alignment model which is proposed by the authors using coreference to refer back. This alignment model tells the docoder to look at the correct non-pronominal coreferent mention in the conversation attention distribution to produce the pronominal reference word.
- **FlowNet** A conversation flow model which learns smooth transitions across turns of conversation.
- **CFNet** A model with both coreference alignment and the conversation flow modeling.

The performance of NQG shows that answer position embedding is helpful for asking questions to the point. Both CorefNet and FlowNet perform better than NQG. They both use hierarchical encoding and hierarchical attention for conversation history to model the dependency across different turns in conversations. The reason that FlowNet performs better than CorefNet is because the conversation flow modeling improves all test samples while the coreference alignment can only contribute those questions containing pronominal references. The performance of the combination model CFNet also proves this point of view.

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	RG-L	Dist-1	Dist-2	Ent-4
ref. Reinforced Dynamic Reasoning (ReDR) (Pan et al., 2019)								
Vanilla Seq2Seq Model	-	-	-	7.64	26.68	0.010	0.034	3.370
NQG (Du et al., 2017)	-	-	-	13.97	31.75	0.017	0.068	6.518
ReDR	-	-	-	19.69	34.05	0.069	0.225	8.367
ref. FlowNet (Gao et al., 2019)								
NQG (Du and Cardie, 2018)	35.56	21.14	14.84	-	45.58	-	-	-
CorefNet	36.89	22.28	15.77	-	46.53	-	-	-
FlowNet	36.87	22.49	15.98	-	46.64	-	-	-
CFNet	37.38	22.81	16.25	-	46.90	-	-	-

Table 4: Experiment results of BLEU-1, 2,3,4 and ROUGE-L scores, which evaluate the relevance of generated questions. Dist-1,2 and Ent-4 evaluate the diversity.

3.3 Limitations

Many QA systems depend on massive Language Models (LMs) to serve as the language representations and entail reasoning abilities. Recent work shows that state-of-the-art LMs (e.g. BERT, GPT-2, etc.) fail at causal inference[1]. When tasked with completing the sentence in EXAMPLEFIG, non-logical yet grammatical sentences are generated. Furthermore, training times increase as more history has to be encoded into the reasoning steps. While some ConvQA systems handle follow-up questions, none are tested on erratic or continuous user input. Most datasets assume the user only answers in a single message. Ideally, these frameworks should also be able handle multiple inputs with distributed information.

4 Conslusion

In conclusion, encoding conversational history plays a large role in how well ConvQA systems reason over questions and answers. Excplicitly embedding history into the used architectures allows the models to more naturally understand context. Simple approaches, like prepending history, are outperformed by modern history embedding approaches, such as HAE and PosHAE. Turn-tracking is a valuable aid to being able to encode conversational history. Incorporating dialog act prediction into models as multi-task training did not show significant improvements. However, further research is needed on the topic.

Future work could look at creating high-quality, complex ConvQA datasets to allow models to learn to reason through conversational history. Further research on combining KG-based and LM-based approaches could help bridge the gap between structured information, massive LMs and their inability to do causal inference. Further research directions could also look at more active or natural conversations in which users may provide information without formal request. Being able to process such input plays a valuable role in the naturalness of chatbots.

[1] https://bit.ly/2PK2aol

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051.

Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2019. Graphflow: Exploiting conversation flow with graph neural networks for conversational machine comprehension.

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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac : Question answering in context. *CoRR*, abs/1808.07036.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. *Proceedings of the 28th*

ACM International Conference on Information and Knowledge Management - CIKM '19.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *CoRR*, abs/1705.00106.

Yifan Gao, Piji Li, Irwin King, and Michael R. Lyu. 2019. Interconnected question generation with coreference alignment and conversation flow modeling. *CoRR*, abs/1906.06893.

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.

Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. *CoRR*, abs/1810.06683.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2017. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *CoRR*, abs/1711.07341.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *CoRR*, abs/1510.03055.

Qian Li, Hui Su, Cheng Niu, Daling Wang, Zekang Li, Shi Feng, and Yifei Zhang. 2019. Answer-supervised question reformulation for enhancing conversational machine comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 38–47, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. *CoRR*, abs/1606.05378.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced dynamic reasoning for conversational question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2124, Florence, Italy. Association for Computational Linguistics.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. BERT with history answer embedding for conversational question answering. *CoRR*, abs/1905.05412.

Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. 2019b. Attentive history selection for conversational question answering. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management - CIKM '19*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. Coqa: A conversational question answering challenge. *CoRR*, abs/1808.07042.

Marzieh Saeidi, Max Bartolo, Patrick S. H. Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. *CoRR*, abs/1809.01494.

Abhishek Sharma, Danish Contractor, Harshit Kumar, and Sachindra Joshi. 2019. Neural conversational qa: Learning to reason v.s. exploiting patterns.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Mark Yatskar. 2019. A qualitative comparison of CoQA, SQuAD 2.0 and QuAC. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2318–2323, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi-Ting Yeh and Yun-Nung Chen. 2019. Flowdelta: Modeling flow information gain in reasoning for conversational machine comprehension.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. *CoRR*, abs/1809.05972.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *CoRR*, abs/1812.03593.