A dataset and baselines for sequential open-domain question answering

Ahmed Elgohary[*], **Chen Zhao**[*], **Jordan Boyd-Graber**
Department of Computer Science, UMIACS, iSchool, Language Science Center
University of Maryland, College Park
{elgohary, chenz, jbg}@cs.umd.edu

Abstract

Previous work on question-answering systems has mainly focused on answering individual questions, assuming they are independent and devoid of context. Instead, we investigate sequential question answering, in which multiple related questions are asked sequentially. We introduce a new dataset of fully human-authored questions. We extend existing strong question answering frameworks to include information about previous asked questions to improve the overall question-answering accuracy in open-domain question answering. The dataset is publicly available at http://sequential.qanta.org.

1 Introduction

The framework of combining information retrieval and neural reading comprehension has been the basis of several systems for answering open-domain questions over unstructured text (Chen et al., 2017; Wang et al., 2018; Clark and Gardner, 2018; Htut et al., 2018). Typically, such systems take one input question at a time, retrieving and ranking multiple paragraphs that potentially contain the answer. A reading comprehension model then produces a ranked list of candidate answer spans from each paragraph. The final answer is then selected from the produced spans.

In information-seeking dialogs, e.g., personal assistants, users interact with a question answering system by asking a sequence of related questions, where questions share the same predicate, entities, or at least a topic. Answering each question in isolation is sub-optimal as information from previously asked questions and previously answers can help better answer *this* question.

We study the task of sequential open-domain question answering. We ask how a standard open-domain question answering system can incorporate connections between question-answer pairs in the same sequence. We introduce QBLink, a new dataset of about 18,000 question sequences (Figure 1); each sequence consists of three naturally occurring human-authored questions (totaling around 56,000 unique questions). The sequences themselves are also naturally occurring (i.e., we do not artificially combine individually-authored questions to form sequences), which allows us to focus more on the important connections between questions that should be incorporated to improve the end-to-end question answering accuracy.

Lead-in: Only twenty-one million units in this system will ever be created. For 10 points each:
Question 1: Name this digital payment system whose transactions are recorded on a "block chain".
Answer: Bitcoin
Question 2: Bitcoin was invented by this person, who, according to a dubious Newsweek cover story, is a 64-year-old Japanese-American man who lives in California.
Answer: Satoshi Nakamoto
Question 3: This online drugs marketplace, Chris Borglum's one-time favorite, used bitcoins to conduct all of its transactions. It was started in 2011 by Ross Ulbricht using the pseudonym Dread Pirate Roberts.
Answer: Silk Road

Figure 1: An example sequence of questions from QBLink. The lead-in and question 1 are asking about the same object/answer. The subject of question 2 is the same as the object of question 1. All questions are about a narrow topic, Bitcoin.

We compare sequence-aware models to baselines that process each question separately. For our sequence-aware models, we tweak the retrieval component by incorporating previous questions and their answers with the current question to find better paragraphs. For the reader, we use the semantic relations between entities in previous questions

[*]The first two authors contributed equally.

(or corresponding answers) and entities mentioned in the paragraph being read (candidate answers) to better choose the answer entity. Both the retrieval and reading steps can be slightly improved by incorporating sequence information.

Our contributions are two-fold: first, we present a new dataset for sequential question answering. Our dataset contains complex questions on many topics. We make the dataset publicly available to encourage future research. Second, we use our dataset to compare baselines in the open-domain question answering setup with the goal of showing that incorporating sequential connections between questions helps.

2 Sequential Question Answering Task

We define the task of open-domain sequential question answering: given a document collection D and questions grouped into disjoint sequences $\{S_i \,|\, i = 1 \dots n\}$ where each S_i is an ordered sequence of question, answer pairs, and a subset of documents $S_i = ((q_i^j,\ a_i^j,\ D_i^j) \,|\, j = 1 \dots m)$, the task is to answer questions $q_i^{\hat{j}}$ with document evidences $D_i^{\hat{j}}$ given access to previously asked questions in the same sequence and their corresponding answers $\{(q_i^j,\ a_i^j) \,|\, j < \hat{j}\}$.

Following Chen et al. (2017), we split the task into two steps—a retrieval step and a reading step. In the retrieval step the current question $q_i^{\hat{j}}$ and previous questions and answers $\{(q_i^j,\ a_i^j) \,|\, j < \hat{j}\}$ are used to retrieve a ranked list of paragraphs $D_i^{\hat{j}}$ from D that are likely to contain the correct answer to the current question $q_i^{\hat{j}}$. The retrieved paragraphs $D_i^{\hat{j}}$ are the input to the reading step that selects a span from $D_i^{\hat{j}}$ as the answer to $q_i^{\hat{j}}$. The reading step has access to previous questions and answers $\{(q_i^j,\ a_i^j) \,|\, j < \hat{j}\}$ as well.

3 Dataset Construction

This section describes QBLink's construction. QBLink is based on the *bonus questions* of **Q**uiz **B**owl tournaments. Unlike previous work that only uses the starter (or tossup) questions (Boyd-Graber et al., 2012), bonus questions are not interruptable (players always hear the complete question) and have greater variability in difficulty. Bonus questions start with a lead-in, which sets the stage for the rest of the question, followed by a sequence of related questions (Figure 1).

Num. Questions (Num. Sequences) $\times$ 3	
Training	45,747 (15,249)
Developme	3,630 (1,210)
Testing	6,555 (2,185)
Num. Sequences per Domain	
Current Events	240
Fine Arts	2,588
Geography	472
History	3,961
Literature	3,879
Mythology	758
Philosophy	692
Religion	746
Science	4,028
Social Science	827
Trash	453
Num. Questions Tokens	
Training	32.6 ± 9.6
Developme	33.5 ± 9.9
Testing	32.1 ± 10.24
Num. TagMe Entities	
Training	2.46 ± 1.68
Developme	2.49 ± 1.68
Testing	2.48 ± 1.77
Num. Unique Answers	43,597
Num. Unique Answer Pages	18,529

Table 1: Statistics about QBLink. Most questions are fairly long and contain 2.5 entity mentions, making the questions relatively complex.

Specifically, we collect bonus questions from http://quizdb.org for the tournaments in 2008–2018. Each question is categorized by topic as history, literature, science, geography, fine arts, philosophy, religion, mythology, social sciences, current events or current events. We filter out too short questions (fewer than ten tokens), and only keep questions with exactly three sub-questions. One advantage of working with Quizbowl data is that the community emphasizes sharing and redistribution of old questions: new students can practice and improve without paying for or licensing questions.

We map the answers to unambiguous Wikipedia pages using combination of rule based matching and fuzzy string matching, then filter out the questions whose answers are not mapped to any Wikipedia page (12.5% of the questions).

To keep our development and test set intact and and of a reasonable percentage of questions, we use the questions in 2014 tournament (the year with the largest number of questions) for development and testing, and the rest of the questions are used for training (Table 1). We use TagMe (Ferragina and Scaiella, 2010) for mention detection and linking question text to Wikipedia.

4 Baselines

We build our baselines DrQA from Chen et al. (2017) for open-domain question answering over Wikipedia.[1] The framework starts with a retrieval phase followed by a reading phase. Retrieval ranks Wikipedia articles using tf-idf (Salton and Buckley, 1987) a question query.

The reading phase is a multi-layer recurrent neural network model that extracts an answer span from the top d retrieved paragraphs. The reader model computes a contextualized representation of each token $\mathbf{t_i}$ by running the token sequence through a multi-layer bidirectional long short-term memory network (BiLSTM) (Hochreiter and Schmidhuber, 1997) and taking the corresponding hidden state to each token at the top layer. The question is encoded as vector $\mathbf{q}$, averaging a BiLSTM's hidden states over the question's tokens. An unnormalized score of t_i encodes which tokens start and end the answer span,

$$Start(i) = \exp(\mathbf{t_i}^T \mathbf{W_{start}} \mathbf{q});$$
$$End(i) = \exp(\mathbf{t_i}^T \mathbf{W_{end}} \mathbf{q}). \quad (1)$$

To find the answer in multiple paragraphs at test time, we merge all paragraphs before feeding them to the reader (Clark and Gardner, 2018).

4.1 Answering Question in Isolation

We experiment with three models that ignore the sequential connections between questions and answer each question in isolation. Our first model is a simple information retrieval (IR) baseline that only uses the retrieval component: the title of the top-1 Wikipedia article is predicted as the answer.

Our second baseline is the full DrQA whose reader is trained/tuned on the training/development questions. To assign paragraphs to each of the training questions, we follow a similar distant-supervision approach to Chen et al. (2017). We retrieve the top twenty Wikipedia articles for each question, exclude the paragraphs that do not contain the gold answer, and then rank the remaining paragraphs using tf-idf. Each of the top ten paragraphs is paired with the question to form a data instance for training the reader.

Finally, we tweak the DrQA reader to limit the candidate answer spans to entity mentions that are linked to Wikipedia. We set the pre-normalization start and end scores of spans that are not detected mentions to zero.

[1] We use the Wikipedia dump of 2017–09–20.

4.2 Incorporating Context in Retrieval

To incorporate the sequential connections between questions in the retrieval phase, we append the previously asked questions to the current question. We also compare appending the predicted answers (top-1 span) to each of the previous questions as well as the gold answers to the current question.

4.3 Incorporating Context in Reader

In addition to encoding which entities have appeared in previous questions, we also want to provide our models with *relationship* information. However, pre-defined relationships from knowledge bases tend to be brittle. Instead, we use a continuous representation of relationships (Iyyer et al., 2016). For example, suppose we want to encode the relationships for an entity (answer candidate) that starts at i and ends at j. We summarize that entities relationships from each of possible k relation-spans. A relation-span is a sequence of tokens from Wikipedia that contains both the answer candidate and an answer to a previous question (For example, the correct answer in Figure 2 has a relation-span *"He is best known for defending President Ronald Reagan during the assassination attempt by John Hinckley Jr."* with the previous answer *"Ronald Reagan"*). This is summarized in a vector $\mathbf{r}_{ij}$ by merging all k relation-spans in a single span that is then fed through a BiLSTM whose hidden states are combined as a weighted sum with self-attention (Lin et al., 2017).

The stronger the similarity between the relation that the question is asking about and the relation-spans, the higher the score of the candidate answer should be. We estimate the similarity r by concatenating the elementwise absolute difference and Hadamard product between $\mathbf{r}_{ij}$ and the question embedding $\mathbf{q}$. We then use a trainable weight vector $\mathbf{w_{rel}}$ to combine the components of the concatenation and produce a single similarity score

$$r = \mathbf{w_{rel}}^T [|\mathbf{q} - \mathbf{r_{ij}}|; \mathbf{q} \circ \mathbf{r_{ij}}].$$

This influences the final selection of the answer span by adding the relation similarity score r to the start and end scores of the candidate answer (Equation 1),

$$Start(i) = \exp(\mathbf{t_i}^T W_{start} \mathbf{q} + r)$$
$$End(j) = \exp(\mathbf{t_j}^T W_{end} \mathbf{q} + r). \quad (2)$$

The relation embedding module is trained jointly with the reader.

Method	EM
Baselines: Questions in Isolation	
IR	15.6
DrQA	39.3
DrQA + limiting to entities	39.7
DrQA + Retrieval with context	
Previous questions	36.4
Previous predicted answers	39.8
Previous gold answers	40.1
DrQA + Reading with context	
Append relation descriptions w/ predicted answers	40.2
Append relation descriptions w/ gold answers	40.7
Explicit relation embedding w/ predicted answers	38.3
Explicit relation embedding w/ gold answers	39.5
IR w/ Previous gold answers + **Reading** w/ Append relation descriptions w/ gold answers	40.7

Table 2: Incorporating sequence information in the retrieval and the reading step slightly improves overall accuracy compared to answering questions in isolation.

5 Baseline Results

We compare the baselines' question answering accuracy: incorporating previous questions and answers slightly improves accuracy (Table 2).

We set the maximum number of retrieved documents to ten, and each document is divided into paragraphs each of 400 tokens. At test time, we merge the top ten ranked such paragraphs and feed them to the reader. We use the reader network of Chen et al. (2017). We limit the number of relation description spans for each entity pair to five. We used an LSTM of one hidden layer and 128 hidden units for the paragraph, question, and relation description encoders. Each reader was trained for twenty epochs.

Table 2 summarizes the results of the baselines (Section 4). Question-answering accuracy is exact-match accuracy since we limit the answer spans to entity mentions whose boundaries are fixed for all models.

Incorporating the previous answer in the retrieval and the reading components slightly improves the overall question answering accuracy (Table 2). The accuracy drops by more than 3% when using the entire text of previous questions in the retrieval phase. Modeling relations reduces the accuracy slightly compared to augmenting paragraphs with relation spans. One possible explanation is that our relation embedding model is under-trained because many questions lack relevant relation-spans. Replacing Wikipedia with a larger corpus (e.g., ClueWeb) or improving reference detection might improve relation embedding model. Unsurprisingly, gold answers to previous questions are more useful than the predicted answers, which highlights a need for models that take into account the uncertainty about previous answers when gold previous answers are not available. However, providing answers to previous questions is consistent for most Quizbowl tournament play.

Question: This man attempted to impress Jodie Foster by shooting Ronald Reagan, but he failed to kill the President. At trial, he was found not guilty by reason of insanity.
Gold answer to previous question: Ronald Reagan
Predict without relation span: George H. W. Bush
Correct answer: John Hinckley Jr.
Relation span: He is best known for defending President Ronald Reagan during the assassination attempt by John Hinckley Jr.

Figure 2: Modeling the relation between *President Ronald Reagan* and *John Hinckley Jr.* expressed by relation span helps the reader select the correct answer entity.

Figure 2 gives an example of how explicit relation embedding helps reader get a correct prediction. Without the relation span, the model predicts George H. W. Bush (vice president at that time) as correct answer. Including the direct relation span between Reagan and John Hinckley Jr., the model gets the correct answer.

6 Related Work and Discussion

We adopt the open-domain question answering framework (Wang et al., 2018; Chen et al., 2017). Previous work considers improving that base framework itself (Clark and Gardner, 2018; Swayamdipta et al., 2018, inter alia) but retains the assumption of answering individual questions.

Aside from the open-domain setup, much of the recent work on question answering focuses on reading-comprehension, where the gold answer to each question is assumed to exist in a given single paragraph for the model to read (Hermann et al., 2015; Rajpurkar et al., 2016; Seo et al., 2017). Another line of work on question answering is question answering over structured knowledge-bases (Berant et al., 2013; Berant and Liang, 2014; Yao and Van Durme, 2014; Gardner and Krish-

namurthy, 2017). Although we focus on general open-domain, QBLink can evaluate both reading-comprehension and knowledge-bases.

Several question answering datasets have been proposed (Berant et al., 2013; Joshi et al., 2017; Trischler et al., 2017; Rajpurkar et al., 2018, inter alia). However, all of them are limited to answering individual questions.

Saha et al. (2018) study the problem of sequential question answering, and introduce a dataset for the task. However, we differ in two aspects: 1) They consider question-answering over structured knowledge-bases. 2) Their dataset construction is synthetic: human annotators collect templates given knowledge-base predicates. Further, sequences are constructed synthetically by grouping individual questions by predicate or subjects.

Both Iyyer et al. (2017) and Talmor and Berant (2018) answer complex questions by decomposing each into a sequence of simple questions. Iyyer et al. (2017) adopt a semantic parsing approach to answer questions over semi-structured tables. They construct a dataset of around 6,000 question sequences by asking humans to rewrite a set of 2,000 complex questions into simple sequences. Talmor and Berant (2018) consider the setup of open-domain question answering over unstructured text, but their dataset is constructed synthetically (with human paraphrasing) by combining simple questions with a few rules.

In parallel to our work, Choi et al. (2018) and Reddy et al. (2018) introduce sequential question answering datasets (QuAC and CoQA) that focus on reading comprehension (i.e., a single text snippet is pre-specified for answering the given questions). QBLink is entirely naturally occurring (all questions and answers were authored independently from any knowledge sources) and is primarily designed to challenge human players.

Our baseline, which improves reading by incorporating additional relation description spans, is similar to Weissenborn et al. (2017) and Mihaylov and Frank (2018), who integrate background commonsense knowledge into reading-comprehension systems. Both rely on structured knowledge bases to extract information about semantic relations that hold between entities. Instead, we extract text spans that mention each pair of entities and encoded them into vector representations of the relations between entities.

7 Conclusions and Future Work

We introduce QBLink, a dataset of 56,000 naturally occurring sequential question, answer pairs. The questions are designed primarily to challenge human players in Quiz Bowl tournaments. We use QBLink to evaluate baselines for sequential open-domain question answering. We show that incorporating sequential information helps slightly improve question answering accuracy.

Because our questions come from the Quizbowl domain, another extension would be to explore how answering linked questions could improve situated gameplay. He et al. (2016) use opponent answers on questions to better estimate what players know; in a complete game with both tossups and bonuses, a complete opponent model would use both to improve strategy.

In the future, we would like to invest in building better sequential question answering models that push the accuracy beyond the presented baselines. Specifically, we will look at how to better model the interaction between the reader and the relation embedding model and how to improve the relation embedding model itself by adopting ideas from the relation extraction (Miwa and Bansal, 2016; Peng et al., 2017; Ammar et al., 2017).

Acknowledgments

We thank the anonymous reviewers and members of the UMD CLIP lab for their comments and suggestions. Jordan Boyd-Graber is supported by NSF Grant IIS-1652666. Ahmed Elgohary is supported by an IBM PhD fellowship. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.

References

Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. 2017. The AI2 system at SemEval-2017 task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *The International Workshop on Semantic Evaluation*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Association for Computational Linguistics*.

Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. 2012. Besting the quiz master: Crowdsourcing incremental classification games. In *Empirical Methods in Natural Language Processing*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics*.

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 *Empirical Methods in Natural Language Processing*.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Association for Computational Linguistics*.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *ACM International Conference on Information and Knowledge Management*.

Matt Gardner and Jayant Krishnamurthy. 2017. Open-vocabulary semantic parsing with both distributional statistics and formal knowledge. In *AAAI Conference on Artificial Intelligence*.

He He, Kevin Kwok, Jordan Boyd-Graber, and Hal Daumé III. 2016. Opponent modeling in deep reinforcement learning.

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*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Phu Mon Htut, Samuel R Bowman, and Kyunghyun Cho. 2018. Training a ranking function for open-domain question answering. In *North American Chapter of the Association for Computational Linguistics: Student Research Workshop*.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *North American Association for Computational Linguistics*.

Mohit Iyyer, Wen tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Association for Computational Linguistics*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Association for Computational Linguistics*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Association for Computational Linguistics*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics* 5.

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*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2018. CoQA: A conversational question answering challenge. In *Empirical Methods in Natural Language Processing*.

Amrita Saha, Vardaan Pahuja, Mitesh M Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *AAAI Conference on Artificial Intelligence*.

Gerard Salton and Chris Buckley. 1987. Term weighting approaches in automatic text retrieval. Technical report, Cornell University.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Swabha Swayamdipta, Ankur P Parikh, and Tom Kwiatkowski. 2018. Multi-mention learning for reading comprehension with neural cascades. In *International Conference on Learning Representations*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *North American Association for Computational Linguistics*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *The 2nd Workshop on Representation Learning for NLP*.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence aggregation for answer re-ranking in open-domain question answering. In *International Conference on Learning Representations*.

Dirk Weissenborn, Tom Koisk, and Chris Dyer. 2017. Dynamic integration of background knowledge in neural NLU systems. *arXiv preprint arXiv:1706.02596* .

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Association for Computational Linguistics*.