INSPIRED: A Large-Scale Dataset and Simulation Framework for Exploring Interactive Learning in Knowledge-Based Question Answering

Lingbo Mo
The Ohio State University
mo.169@osu.edu

Ashley Lewis
The Ohio State University
lewis.2799@osu.edu

Huan Sun
The Ohio State University
sun.397@osu.edu

Michael White
The Ohio State University
white.1240@osu.edu

Abstract

This paper gives a condensed summary of our work (Mo et al., 2022) published in the Findings of ACL 2022 and adds a discussion section to talk about its connection with interactive learning. Existing studies on semantic parsing focus on mapping a natural-language utterance to a logical form (LF) in one turn. However, given that natural language may contain ambiguity and variability, it is challenging for a parser to obtain high enough accuracy for real use. In this work, we present INSPIRED [1], a large-scale dataset for interactive semantic parsing for knowledge-based question answering (KBQA), which involves human feedback in the loop to increase the parsing accuracy. In order to improve the transparency of the parsing process and the user experience, we investigate an interactive semantic parsing framework that explains the predicted LF *step by step* in natural language and enables the user to make corrections through *natural-language feedback* for individual steps. Moreover, we develop a simulation pipeline for automated evaluation of our framework w.r.t. a variety of KBQA models without further crowdsourcing effort. The results demonstrate that our framework equipped with the dataset is promising to be effective across such models. We further discuss its potential use for interactive learning in the end.

1 Introduction

This paper summarizes our work (Mo et al., 2022) in the Findings of ACL 2022 and further discusses its potential use for research on interactive learning in KBQA. We focus on the semantic parsing task which aims to map natural language (NL) to formal meaning representations, such as λ-DCS, API calls, SQL and SPARQL queries. As seen in previous work (Liang et al., 2013; Yih et al., 2014, 2015; Talmor and Berant, 2018b; Chen et al., 2019; Lan and Jiang, 2020a; Gu et al., 2021), parsers still face major challenges: (1) the accuracy of SOTA parsers is not high enough for real use, given that natural language questions can be ambiguous or highly variable with many possible paraphrases, and (2) it is hard for users to understand the parsing process and validate the results.

In response to the challenges above, researchers have been exploring *interactive semantic parsing*, where human users give feedback and boost system accuracy. For example, Artzi and Zettlemoyer (2011) utilize conversation logs to improve a Combinatory Categorial Grammar (CCG) parser for

[1] Our INSPIRED dataset and code are available at https://github.com/molingbo/INSPIRED.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

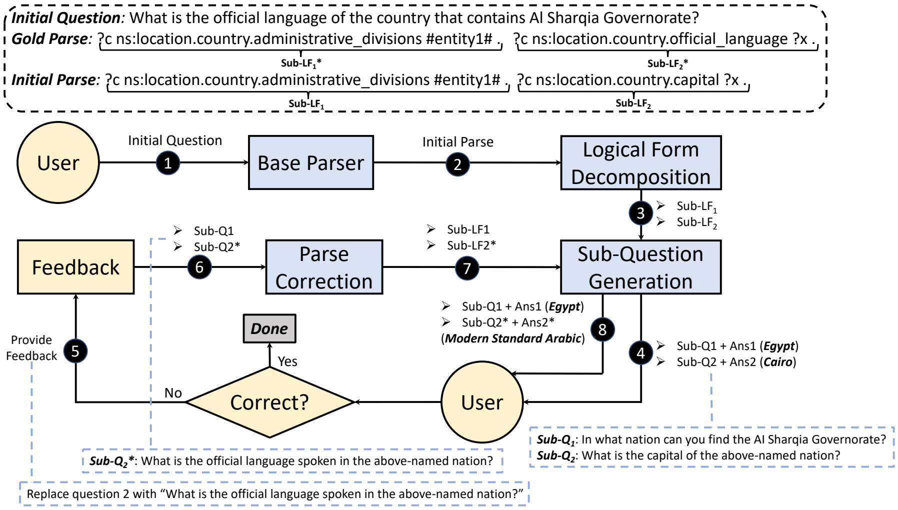

Initial Question: What is the official language of the country that contains Al Sharqia Governorate?
Gold Parse: ?c ns:location.country.administrative_divisions #entity1# . ?c ns:location.country.official_language ?x .
Sub-LF$_1$*
Sub-LF$_2$*
Initial Parse: ?c ns:location.country.administrative_divisions #entity1# . ?c ns:location.country.capital ?x .
Sub-LF$_1$
Sub-LF$_2$
User
Initial Question
1
Base Parser
Initial Parse
2
Logical Form Decomposition
3
Sub-LF$_1$
Sub-LF$_2$
Feedback
Sub-Q1
Sub-Q2*
6
Parse Correction
Sub-LF1
Sub-LF2*
7
Sub-Question Generation
Sub-Q1 + Ans1 (Egypt)
Sub-Q2* + Ans2* (Modern Standard Arabic)
8
4
Sub-Q1 + Ans1 (Egypt)
Sub-Q2 + Ans2 (Cairo)
Done
Yes
Provide Feedback
5
No
Correct?
User
Sub-Q$_1$: In what nation can you find the Al Sharqia Governorate?
Sub-Q$_2$: What is the capital of the above-named nation?
Sub-Q$_2$*: What is the official language spoken in the above-named nation?
Replace question 2 with "What is the official language spoken in the above-named nation?"

Figure 1: Illustration of our interactive semantic parsing framework for KBQA. The box on the top lists a running example. The prefix of a SPARQL query (i.e., LF used for KBQA in this paper) is omitted for brevity. The bottom figure shows the entire workflow of our framework.

understanding user utterances. Thomason et al. (2015) employ incremental learning of a parser from conversations on a mobile robot. Gur et al. (2018) ask multiple choice questions about a limited set of predefined errors. Su et al. (2018) show that fine-grained user interaction greatly improves the usability of natural language interfaces to Web APIs. Yao et al. (2019) allow their semantic parser to ask users clarification questions when generating an If-Then program. Recently, Elgohary et al. (2020) crowdsource the SPLASH dataset for correcting SQL queries using natural language feedback. Elgohary et al. (2021) convert feedback in SPLASH into a canonical form of edits that are deterministically applied. In this work, we focus on parse errors in KBQA and propose to do the step-by-step correction through decomposition. We break down the parse into a sequence of sub-components and enable the user to provide step-by-step feedback, thereby simplifying the task of parse correction and increasing the likelihood of an accurate parse.

Our main contributions are as follows: (1) We design a more transparent interactive semantic parsing framework that explains to a user how a complex question is answered step by step and enables them to make corrections in natural language and trust the final answer. Figure 1 illustrates this framework. (2) To support research on interactive semantic parsing for KBQA, we release a high-quality dialogue dataset called INSPIRED (**IN**teractive **S**emantic **P**ars**I**ng for Cor**RE**ction with **D**ecomposition) using our framework. (3) We establish baseline models for two core sub-tasks in this framework: Sub-Question Generation and Parse Correction. (4) Although INSPIRED is constructed using a selected base parser, it can be used to train models to simulate user feedback, allowing us to study the promise of our framework to correct errors made by other semantic parsers without more annotation effort.

2 Dataset

2.1 Dataset Construction

Following the design of our interactive semantic parsing framework, we design a workflow for dataset construction. Firstly, we prepare pairs of complex questions and SPARQL parses predicted by a base semantic parser. Then, we decompose the gold and predicted parses and determine *correction operations*. The sub-LFs are translated to NL questions using templates and we employ crowdworkers to paraphrase these questions to be more natural and fluent.

Preparing Questions and Logical Form Decomposition. We start with the COMPLEXWEBQUESTIONS 1.1 (CWQ) dataset (Talmor and Berant, 2018a,b), which contains complex questions paired with gold SPARQL queries for Freebase (Bollacker et al., 2008). We adopt a transformer-based seq2seq model (Vaswani et al., 2017) as the base semantic parser to prepare a predicted SPARQL query for each complex question. An important goal of creating INSPIRED is to make the process of question answering transparent to the user. Each dialogue features a decomposition process by which our framework transforms the complex question into an initial parse, breaks it into sub-LFs, retrieves answers, and presents this whole process in natural language to the user for correction.

Crowdsourcing. To make queries understandable for an average user, as in the Sub-Qs in Figure 1, we translate the decomposed LFs into English questions using templates. To obtain natural sounding questions, we conduct crowdsourcing on Amazon Mechanical Turk (AMT), in which crowdworkers are employed to rephrase sub-questions from the clunky, templated form into more concise and natural English in the context of a dialogue. The task is conducted using ParlAI (Miller et al., 2017), which allows us to set up a versatile dialogue interface. We specify ethical considerations during our crowdsourcing process in Appendix C.

2.2 Dataset Statistics and Analysis

Dataset Statistics. We create 10,374 dialogues in total, based on 3,492 questions from the training set, 3,441 from the validation set, and 3,441 from the test set of CWQ. We omit a small set of questions from the original validation and test sets that are consistently confusing to crowdworkers. Table 2 in Appendix A shows a breakdown of the CWQ question types in the INSPIRED dataset, along with the average number of corrections and sub-questions.

Data Quality. We meticulously design the data collection process to make sure of a high-quality dataset. During the data collection process, the crowdworkers read a detailed tutorial, pass two qualification tasks, and have their work spot-checked at each stage. We keep our pool of workers small and are thus able to maintain frequent communication with them throughout the process, giving feedback in an ongoing fashion. We also use a semi-automatic data cleaning method to identify inaccurate paraphrases for manual repair, resulting in edits to 325 sub-questions in total. Check our previous work (Mo et al., 2022) for more detailed analysis of the characteristics of paraphrases, the diversity of the questions, and comparisons of the templated and rephrased questions.

3 Experiments

In this section, we conduct extensive experiments on two core sub-tasks (i.e., **sub-question generation** and **parse correction**) in our framework. We treat both of them as seq2seq tasks and incorporate different contexts in the dialogue for exploration. Experimental results indicate that the BART-large (Lewis et al., 2020) model with inputs that leverage both the history of sub-questions and sub-LFs achieves the best performance for the parse correction task. Meanwhile, by adding the complex question and the history of templated sub-questions to the input, BART-large performs the best for the sub-question generation task (see details in Appendix B). These trained models for sub-question generation and parse correction will be used in our simulation pipeline described below.

3.1 Simulation

Furthermore, in order to study the promise of our framework for other KBQA parsers (beyond the one used to construct INSPIRED) without introducing extra crowdsourcing effort, we design a simulation pipeline which simulates dialogues based on our sub-question generation and parse correction models for automated evaluation. To simulate a dialogue, the pipeline consists of the following steps: (1) Automatically translate a parser's predicted LFs into natural questions using the trained sub-question generation model (described above). (2) Use oracle error detection and train a generator to simulate a human user's corrections for these dialogues. This generator is a BART-large model that leverages the complex question and templated sub-questions as input to generate human feedback. (3) Correct erroneous parses using the previously trained parse correction model (described above).

We conduct simulation experiments on BART-large (Lewis et al., 2020) and QGG (Lan and Jiang, 2020b) which are representatives from two mainstream methodologies for KBQA. We report both F1 and EM on the test set for BART-large before and after the correction process using the simulation pipeline. For QGG, since its generated query graphs do not take exactly the same format as SPARQL queries, we report F1 score of the predicted answers only. As shown in the left part of Table 1, the performance gains on both models after the correction show that INSPIRED can help train effective sub-question generation and parse correction models, which makes our interactive framework applicable

	BART-large	QGG
EM	60.9	-
EM*	**75.1**	-
F1	65.8	49.0
F1*	**75.7**	**56.5**

Attempt	EM	F1
BART-large		
1	75.1	75.7
2	78.7	79.9
3	**79.0**	**80.1**

Table 1: The left table shows the performance of two types of semantic parsers (BART-large and QGG) after parse correction through our simulation process. * denotes results after correction. The right table shows BART-large's performance after multiple attempts of correction.

to other KBQA parsers. Simulating user feedback makes it easy and far less costly to understand the potential of our interactive framework for any base parser (as long as it outputs LFs).

Moreover, we expand the simulation experiment to include multiple attempts of correction to simulate situations in which the model does not repair the parse correctly on the first attempt. We use the same human feedback generator to decode several of the highest scoring sequences as candidates for different attempts at correction. We evaluate this strategy with a maximum of three attempts. As shown in the right part of Table 1, F1 scores are up to 80.1 after three attempts of correction.

4 Discussion

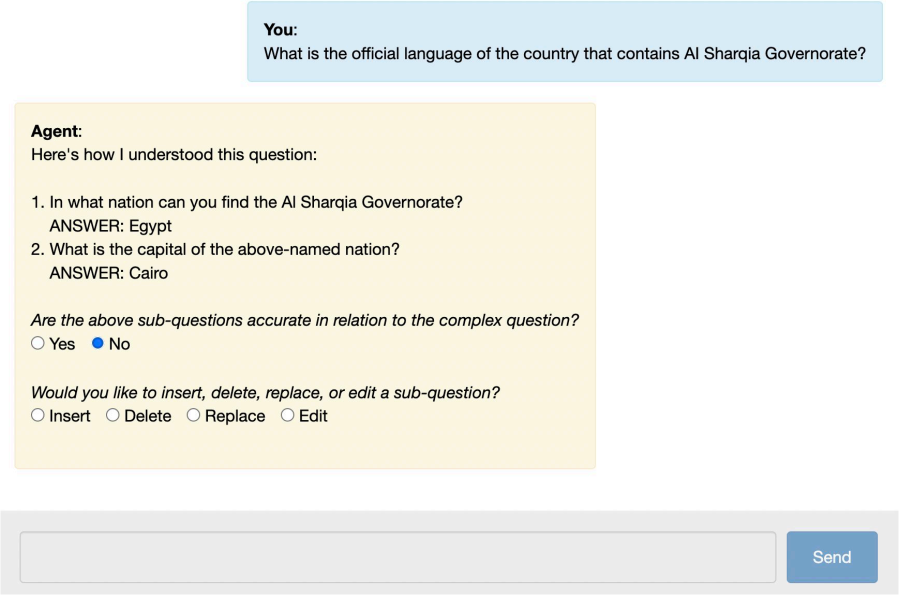
Our framework incorporates the sub-questions paraphrased by the crowdworkers in the INSPIRED dataset to train the parse correction model for interactive parsing. It could be easily adapted to interactive learning with simulated users in the simulation pipeline or real users in our upcoming user study (see details below).

Interactive Learning via Simulation Pipeline. Our simulation pipeline can be applied to other KBQA parsers and include multiple attempts of correction to a single step. Beyond that:

- The simulation pipeline provides an efficient means of automatically evaluating different model variants and settings in an interactive scenario without requiring further human effort. Though currently it relies on the gold logical forms to determine whether the parsing has been corrected, one could train an additional classifier using the annotation of the natural language questions to make this judgement automatically, removing the dependency on gold data. Also, in real use, we assume that the user stands a good chance of identifying if there are still parsing errors or not.
- The simulation pipeline can produce more simulated interactions and correction turns in a dialogue when handling new questions. If we incrementally add simulated data produced by the pipeline to the training process of the parse correction model, the learning curves could be obtained to explore the relationship between the simulated data volume and the parsing performance. We can also conduct the similar study w.r.t. the training of base parsers over time with either the simulation framework or actual human feedback. Exploring this would allow us to examine the balance of efficiency, both in human annotation and model resources, and performance gains. Also, our simulation pipeline targets the human-in-the-loop scenario described below.

Human in the Loop. Beyond the simulation pipeline, our framework is designed for a human-in-the-loop setup and could learn through interaction with users. We are conducting a user study to test the viability of this framework for real use and provide insights about its scalability.

- First, in the user study we are doing, users are employed on the Amazon Mechanical Turk platform. Before doing the real task, participants need to read through a tutorial and complete a qualification task in the form of a multiple-choice quiz. In the real task, they utilize the framework to correct parsing errors and query a knowledge base for answers in real time. As shown in Figure 2, they can specify edit operations using button-clicks, then delete a sub-question or use a response box to insert, replace or edit a sub-question. This structured method of providing feedback allows the human user to easily make edits and the system to easily understand those edits. In terms of evaluation, we will use parse accuracy after the interaction to verify the usefulness of our framework. We will also ask survey questions to measure the subjective quality of the machine-generated sub-questions, intermediate and final answers, accessibility of the overall system, etc.



You:
What is the official language of the country that contains Al Sharqia Governorate?

Agent:
Here's how I understood this question:

1. In what nation can you find the Al Sharqia Governorate?
   ANSWER: Egypt
2. What is the capital of the above-named nation?
   ANSWER: Cairo

Are the above sub-questions accurate in relation to the complex question?
○ Yes ● No

Would you like to insert, delete, replace, or edit a sub-question?
○ Insert ○ Delete ○ Replace ○ Edit

Figure 2: User study interface, which is implemented using ParlAI (Miller et al., 2017). In addition to inserting/deleting/replacing sub-questions, we provide a new operation 'edit' to support minor changes, where the original sub-question is auto-filled into the response box after the user makes the selection.

- Second, we hope to generalize the framework to handle other unlabeled questions for parsing. By interacting with the framework, human users can provide feedback to directly correct parse errors without requiring extra annotations and validate the answer in a timely manner. Meanwhile, the interactions in the dialogue can be collected to provide data for re-training the parse correction model and the base parser continuously. On top of that, the mispredicted parses recognized by the user can be treated as negative instances to enhance the training process via contrastive learning, which would help further improve the parsing performance.

5 Conclusion

We have given a summary of our work in the Findings of ACL 2022. In the ACL paper, we proposed an interactive semantic parsing framework and instantiated it with KBQA. Using this framework, we crowdsourced a novel dataset, dubbed INSPIRED, and experimentally showed that it could greatly increase the parse accuracy of a base parser. In addition, we designed a simulation pipeline to explore the potential of our framework for a variety of semantic parsers, without further annotation effort. The performance improvement showed interactive semantic parsing could be promising for further improving KBQA. In this work, we added a discussion section to further talk about its connection with and potential use for interactive learning.

Acknowledgments

We thank the Clippers and Pragmatics groups at OSU for helpful discussion. We also thank Yu Su, Xiang Deng, Luke Song and Vardaan Pahuja for their valuable feedback. This research was partially supported by a collaborative open science research agreement between Facebook and The Ohio State University. MW has been a paid consultant for Facebook during the period of research. HS was partly supported by NSF IIS-1815674 and NSF CAREER #1942980.

References

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 421–432.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering. 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 345–356.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ahmed Elgohary, Ahmed Hassan Awadallah, et al. 2020. Speak to your parser: Interactive text-to-sql with natural language feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2065–2077.

Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. NL-EDIT: Correcting semantic parse errors through natural language interaction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5599–5610, Online. Association for Computational Linguistics.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488.

Izzeddin Gur, Semih Yavuz, Yu Su, and Xifeng Yan. 2018. Dialsql: Dialogue based structured query generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1339–1349.

Yunshi Lan and Jing Jiang. 2020a. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974.

Yunshi Lan and Jing Jiang. 2020b. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Percy Liang, Michael I Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84.

Lingbo Mo, Ashley Lewis, Huan Sun, and Michael White. 2022. Towards transparent interactive semantic parsing via step-by-step correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 322–342.

Yu Su, Ahmed Hassan Awadallah, Miaosen Wang, and Ryen W White. 2018. Natural language interfaces with fine-grained user interaction: A case study on web apis. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 855–864.

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

Alon Talmor and Jonathan Berant. 2018a. Repartitioning of the complexwebquestions dataset. *arXiv preprint arXiv:1807.09623*.

Alon Talmor and Jonathan Berant. 2018b. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Jesse Thomason, Shiqi Zhang, Raymond J Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ziyu Yao, Xiujun Li, Jianfeng Gao, Brian Sadler, and Huan Sun. 2019. Interactive semantic parsing for if-then recipes via hierarchical reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 643–648.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. 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 1298–1308.

A Dataset Statistics

Table 2 presents the statistics of our INSPIRED dataset. It shows a breakdown of the CWQ question types in the INSPIRED dataset, along with the average number of corrections and sub-questions.

B More Experimental Results

In this section, we explore two sub-tasks under our framework (i.e., parse correction and sub-question generation). We treat both of them as seq2seq tasks, then present and evaluate several baseline models including Seq2Seq (Sutskever et al., 2014), Transformer (Vaswani et al., 2017), BART-base and BART-large (Lewis et al., 2020) for each task, in which we use INSPIRED for training and testing.

B.1 Parse Correction with NL Feedback

Given a sub-question q, the parse correction task is to convert it into a new sub-LF p. By parsing the templates used by correction operations, we extract the operation (i.e., replace, delete, or insert a sub-question) and apply it to the appropriate step. Then, sub-LFs are compiled accordingly to form a correction parse P for the entire question. We predict the sub-LF based on q without considering contexts, and present the results of several baselines. We report both the turn-level accuracy—the

Number of	**Train**	**Dev**	**Test**	**Overall**
Complex Questions	3,492	3,441	3,441	10,374
- Composition	1,196	1,532	1,490	4,218
- Conjunction	1,796	1,503	1,553	4,852
- Comparative	253	217	207	677
- Superlative	247	189	191	627
Predicted Sub-Questions	1.7	2.0	1.9	1.9
Gold Sub-Questions	2.2	2.1	2.1	2.1
Range of the number of predicted sub-questions				0 - 5
Range of the number of gold sub-questions				2 - 4
Average number of edits				1.4
Dialogues with 0 edits				5,016

Table 2: Statistics for our INSPIRED dataset: the number of complex questions for each reasoning type, the average number of sub-questions and edit operations in a dialogue (excluding those that do not have edits).

Correction Models	**Turn-level EM**	**Dialog-level EM**
w/o Correction	-	52.3
2nd-Beam	-	55.8
Seq2Seq(LSTM)	78.9	65.0
Transformer	81.2	68.0
BART-base	82.3	70.3
BART-large	**82.9**	**71.3**

Table 3: Turn-level and Dialogue-level accuracy of different models after incorporating feedback (where applicable).

accuracy of sub-LFs in correction turns—and the dialog-level accuracy—the end-to-end accuracy of the entire LFs after correction—on our test set.

Since models like BART adopt a subword tokenization scheme, the validness of predicates generated by concatenating subwords can not always be guaranteed. We use beam search of size 10 to generate LFs as candidates, filtering those with invalid predicates and excluding erroneous predictions previously made by the parser. We additionally compare with a baseline named 2nd-Beam, which applies beam search on the base parser to obtain two initial parses and uses the second for parse correction. It has some performance gains over the setting without correction, but is much lower than those settings with human feedback. Results in Table 3 further suggest: (1) incorporating human feedback can substantially improve the parse accuracy and (2) using BART-large with pretraining as the correction model achieves the best performance, achieving 19.0 points higher than the base parser without correction in terms of the dialog-level EM score.

Then, using BART-large as the correction model, we further study the correction process by concatenating different contexts to the input, including the history of sub-questions h_q and sub-LFs h_{lf}. We report both the accuracy for each turn of correction and the end-to-end accuracy. As shown in Table 4, we find that: (1) Adding contexts into the input can further improve the correction accuracy. (2) As the number of turns goes up, context contributes more to the correction process, which indicates that including the full dialogue history in the input leads to the best results. (3) The BART-large model with inputs that leverage h_q and h_{lf} achieves the best performance, with a 21.2 increase under dialog-level EM compared to the base parser without correction.

B.2 Sub-Question Generation

Sub-question generation aims to translate a sub-LF p into a natural sub-question q. We explore an off-the-shelf paraphrasing model,[2] which takes corresponding templated sub-question q^t as the input and outputs q. It is fine-tuned on BART-large using three paraphrasing datasets including

[2] https://huggingface.co/eugenesiow/bart-paraphrase

Context	Dialog-level EM	Turn-1 (3441)	Turn-2 (3441)	Turn-3 (345)	Turn-4 (56)
w/o Correction	52.3	-	-	-	-
BART-large					
w/o Context	71.3	84.6	81.5	85.5	53.6
+ h_q	72.2	84.7	82.2	89.3	**100.0**
+ h_{lf}	72.0	84.3	82.1	89.3	**100.0**
+ h_q & h_{lf}	**73.5**	**86.4**	**83.2**	**91.0**	**100.0**

Table 4: Parse correction performance when considering different contexts. h_{lf} and h_q denote the dialogue history of sub-LFs and sub-questions respectively.

Generation Models	BLEU-2	BLEU-4	BERTScore
BART-paraphrase	10.6	2.7	88.0
Seq2Seq(LSTM)	17.8	6.4	90.8
Seq2Seq(LSTM)^t	18.7	6.7	91.3
Transformer	21.1	8.4	91.7
Transformer^t	23.4	9.1	92.6
BART-base	30.7	15.0	93.8
BART-base^t	32.0	15.9	94.1
BART-large	31.5	15.4	94.0
BART-large^t	**32.4**	**16.2**	**94.2**

Table 5: Question generation performance of different models. t denotes that the input incorporates templated sub-question, as well as the current sub-logical form.

Context	BLEU-2	BLEU-4	BERTScore
BART-large^t			
w/o Context	32.4	16.2	94.2
+ h_{q^t}	33.3	16.5	94.6
+ Q	33.4	16.6	94.6
+ Q & h_{q^t}	**34.1**	**17.1**	**94.8**

Table 6: Comparison of question generation performance when considering different contexts in the input.

Quora,[3] PAWS (Zhang et al., 2019) and MSR paraphrase corpus (Dolan and Brockett, 2005). The low scores demonstrate that sub-question generation is more challenging than a simple paraphrasing task. For the other models, we explore two scenarios with different inputs: (1) sub-LF p only and (2) a concatenation of p and q^t. We report BLEU scores based on n-grams overlap and BERTScores measuring semantic similarity. The results in Table 5 suggest that: (1) Using BART-large as the generation model achieves the best performance and (2) incorporating the templated sub-questions into the model input can improve performance on all baselines, which makes sense because some tokens in q^t can be directly copied into the output question.

Furthermore, we use the best-performing model (i.e. BART-large with both p and q^t as the input) in Table 5 as the basic setting to explore the modeling of different contexts including the complex question Q and the history of templated sub-questions h_{q^t}. As shown in Table 6, we find that (1) adding context into the model's input can obtain higher metric scores, which suggests that context can help in a dialogue. (2) Those settings that incorporate the original complex question Q generally perform better than the others, since the complex question contains the semantics of the sub-question to be generated. (3) BART-large with the input containing both Q and the history of templated sub-questions h_{q^t} achieves the best performance. We also tried incorporating the history of sub-LFs h_{lf}, but it does not help further improve the performance.

[3] https://www.kaggle.com/c/quora-question-pairs

C Ethical Considerations

IRB Approval. Prior to collection of the INSPIRED dataset, we obtain IRB (Institutional Review Board) approval at our institution. This data collection is considered Exempt Research, meaning that our human subjects are presented with no greater than minimal risk by their participation. Participants' personal information is not collected, aside from minimal demographic information including their native language, which is used to ensure native-speaker level proficiency in the dataset. No identifying information is included. Further, all participants are required to read and agree to an *informed consent* form before proceeding with the task. AMT automatically anonymizes crowdworkers' identities as well.

Compensation to Crowdworkers. In order to ensure both quality data collection and fair treatment of our crowdworkers, we carefully review our payment plan for the AMT task. After a pilot study we gauge the average amount of time we expect a task to require and adjust the payment amount per task according to the minimum wage amount in our state, resulting in a 70 cent payment per task. Further, we ensure compensation for the time spent on the tutorial and qualification task by awarding $10 bonuses after completion of their first 10 tasks. They also receive $10 bonuses upon every 100 tasks they complete. In total, the cost of creating the INSPIRED dataset is approximately $13,300.