Insights of a Usability Study for KBQA Interactive Semantic Parsing: Generation Yields Benefits over Templates but External Validity Remains Challenging

Ashley Lewis,[1] Lingbo Mo,[1] Marie-Catherine de Marneffe,[2]
Huan Sun,[1] Michael White[1]
[1]The Ohio State University [2]FNRS - UCLouvain
[1]{lewis.2799, mo.169, sun.397, white.1240}@osu.edu [2]marie-catherine.demarneffe@uclouvain.be

Abstract

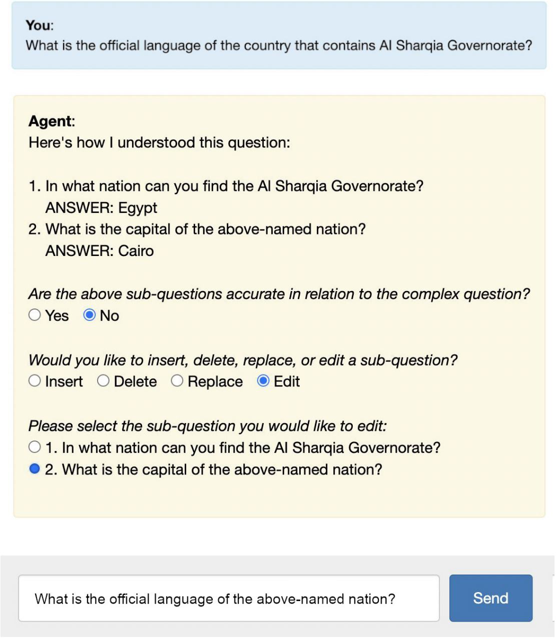
We present our findings from a usability study of an interactive semantic parsing system for knowledge-based question answering (KBQA). The system is designed to help users access information within a knowledge base without having to know its query language. The system translates the user's question into the query language, retrieves an answer, then presents an English explanation of the process so that the user can make corrections if necessary. To our knowledge, our work is the most thorough usability study conducted for such a system and the only one that uses crowdworkers as participants to verify that the system is usable for average users. Our crowdworkers participate in KBQA dialogues using 4 versions of a system based on the framework by Mo et al. (2022) and answer surveys about their experiences. Some key takeaways from this work are: 1) we provide evidence for the benefits of interactivity in semantic parsing with human users and using generated questions in lieu of templated representations, 2) we identify limitations of simulations and provide contrasting evidence from actual system use, and 3) we provide an examination of crowdsourcing methodology, in particular the trade-offs of using crowdworkers vs. a specially trained group of evaluators.

Keywords: interactive semantic parsing, user study, usability study, crowdsourcing

1. Introduction

Semantic parsing endeavors to translate everyday language into formal representations of meaning, encompassing structures like API invocations, SQL, and SPARQL queries. Even as large language models (LLMs) make headway on the task (Xu et al., 2023), parsing still grapples with significant hurdles: (1) parsers struggle with accuracy, unable to reliably handle the inherent ambiguity and diverse paraphrases characteristic of natural language queries, and (2) the parsing process remains opaque to users, impeding their ability to comprehend and verify results (Chen et al., 2023).

Addressing these obstacles, recent work delves into *interactive semantic parsing*, which involves human users in providing feedback to enhance system accuracy. Mo et al. (2022) does this for knowledge-base question answering (KBQA), developing an interactive framework to correct parses that offers a step-by-step explanation of the predicted query and allows users to correct each step using natural language. In addition to this system, they release the INSPIRED (**IN**teractive **S**emantic **P**ars**I**ng for Cor**RE**ction with **D**ecomposition) dataset to facilitate further exploration of interactive semantic parsing for KBQA. This dataset uses target questions from the COMPLEXWEBQUESTIONS (CWQ) dataset (Talmor and Berant, 2018).



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

Please select the sub-question you would like to edit:
○ 1. In what nation can you find the Al Sharqia Governorate?
● 2. What is the capital of the above-named nation?

What is the official language of the above-named nation?

Figure 1: Example of the user interface on Amazon Mechanical Turk, created using the ParlAI/Mephisto system (Miller et al., 2017; Urbanek and Ringshia, 2023).

Using the INSPIRED dataset and the framework developed by Mo et al. (2022), we conduct a usability study in which we have Amazon Mechanical Turk (henceforth MTurk) workers test four different versions of a system using an interface developed using the ParlAI/Mephisto platform (Miller et al., 2017; Urbanek and Ringshia, 2023), which allows dialogue tasks to be run on MTurk. An example of the

Proceedings of the HumEval @LREC-COLING 2024, pages 47–62
Turin, Italy. May 21st, 2024. ©2024 European Language Resources Association (ELRA)

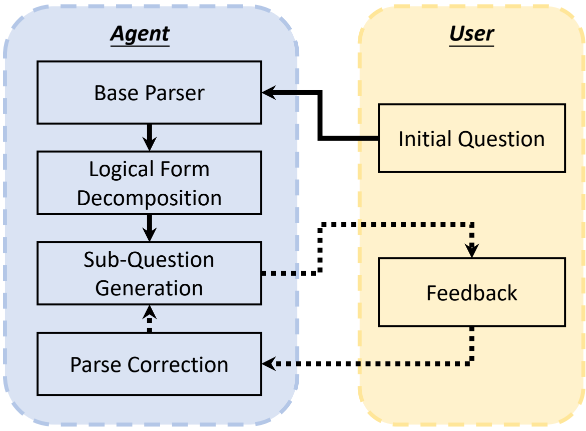


Figure 2: A high level overview of the framework described by Mo et al. (2022) for their interactive semantic parsing framework for KBQA.

interface can be seen in Figure 1, where the user is given the target question: *What is the official language of the country that contains [the] Al Sharqia Governorate?* For each dialogue, users are given a target question (originally from the CWQ dataset). These questions are multi-hop, meaning that they require more than one reasoning step to answer (hence the *complex* in ComplexWebQuestions). Mo et al. (2022) use a finetuned Transformer model (Vaswani et al., 2017) as a base automatic semantic parser to convert the question into a SPARQL query. They then decompose the query into pieces and translate those pieces to natural English sub-questions using a generation model.

We divert slightly from this method by first translating the query pieces into templated English using rule-based translation and then using a T5-based seq2seq model to translate the templates to natural language. This added step was implemented because query languages are unlikely to be well-represented in the pretraining data of the transformer, which makes the task more difficult (Kale and Rastogi, 2020). We then show these questions, along with intermediate answers retrieved from Freebase, to the user who can then use radio buttons to indicate whether or not the sub-questions and answers are correct. If they are not, the user can provide edits via further radio button selection (*insert, delete, replace,* or *edit*) and a text box for typing new questions (see Figure 1). The system then takes these edits and uses a parse correction model to try to generate a new parse, which again gets translated to natural-language sub-questions. This process, in theory, can repeat as many times as necessary. Figure 2 shows an overview of the system framework. To evaluate the viability of this system for real use, we recruited 48 crowdworkers to conduct dialogues in various conditions.

The contributions of this paper are as follows: 1) We provide a methodology for a thorough usability study of the system using crowdworkers (Section 2). This methodology consists of a careful task design in which crowdworkers perform tasks in two systems and compare them. 2) We provide evidence of the benefits of the system designed by Mo et al. (2022) and identify areas for improvement (Section 3). In particular, we validate the choice to use interactivity, which had been shown to be helpful in previous work (Tian et al., 2023; Narechania et al., 2021; Yan et al., 2023; Elgohary et al., 2021), though mostly through user simulation. We provide external validity with human users. We also validate the use of a generation model instead of templated representations of questions. In terms of areas for improvement, we find that there are gains to be made in how dialogue context is best leveraged. 3) We examine the usability study with a critical eye and outline some lessons learned, which can help future research and usability studies (Section 4). We outline difficulties of Amazon Mechanical Turk and examine the trade-offs of using crowdworkers instead of a dedicated group of evaluators.

All data and models can be found on Github [1].

2. Usability Study Design

Our motivations for conducting a usability study were three-fold: 1) We wanted to verify that our system allows users of all types, even novices, to query knowledge bases with complex questions. The goal of our system is to elucidate the parsing process by decomposing the parse and translating it to understandable English sub-questions. It is designed to be intuitive and easy to use, so wanted to verify that this is indeed the case. 2) We wanted to identify problems and areas of improvement for the system. 3) We wanted to analyze how different aspects of the system influence user experience and success. We achieved this through ablations, testing a total of four systems.

2.1. System Designs

The first system is called the Full system, as it has no ablations. In this system, we use "fully contextualized" parse correction and generation models (see Figure 3 for further explanation). We use a model very similar to the best parse correction model from Mo et al. (2022), which is a finetuned T5 seq2seq model that takes the user's current correction, the original target question, and any previous sub-questions as input and outputs a corrected partial parse. The only difference in our model is that, instead of outputting a partial SPARQL query, it outputs a templated English version of the query. As mentioned, we do this because seq2seq models

---

[1] `https://github.com/ashleylew/KBQA_Interactive_Semantic_Parsing`

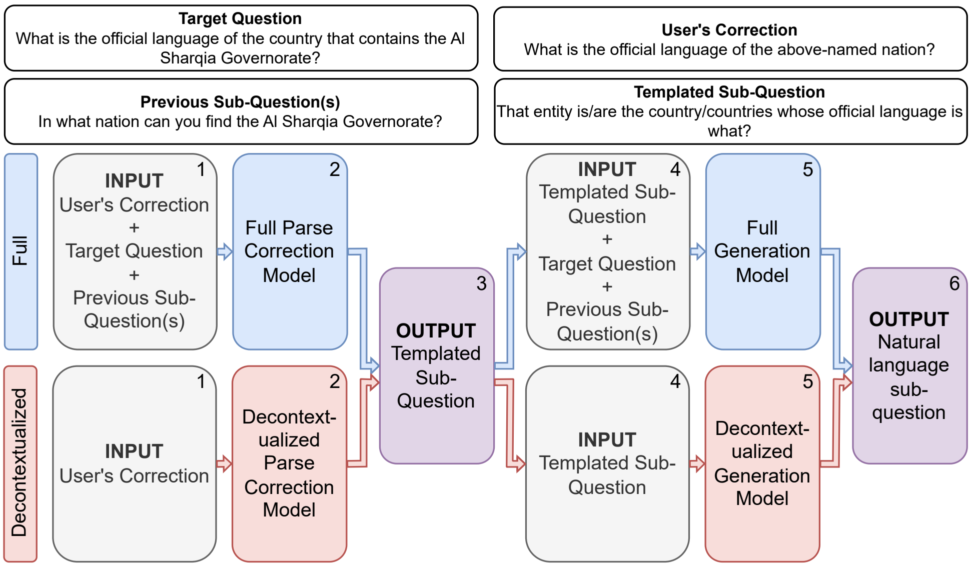

Target Question
What is the official language of the country that contains the Al Sharqia Governorate?
User's Correction
What is the official language of the above-named nation?
Previous Sub-Question(s)
In what nation can you find the Al Sharqia Governorate?
Templated Sub-Question
That entity is/are the country/countries whose official language is what?
Full
Decontextualized
INPUT 1
User's Correction + Target Question + Previous Sub-Question(s)
2 Full Parse Correction Model
3 OUTPUT Templated Sub-Question
INPUT 4
Templated Sub-Question + Target Question + Previous Sub-Question(s)
5 Full Generation Model
6 OUTPUT Natural language sub-question
1 INPUT User's Correction
2 Decontextualized Parse Correction Model
4 INPUT Templated Sub-Question
5 Decontextualized Generation Model

Figure 3: A comparison of the Full and Decontextualized systems, which differ in the inputs on which the models are trained and tested. As you can see, the Full system is given more grounding information in the inputs of each model to use to produce an output. Example content can be found at the top of the diagram, which corresponds to the example in Figure 1. Not shown here is the Templated system which would follow the same method as the Full system but would terminate at step 3, as it does not use a generation model. The Non-Interactive system does not use either model.

tend to perform better on text-to-text tasks as opposed to text-to-query-language, as the query language is unlikely to be nearly as well represented in its training data (Kale and Rastogi, 2020).

Likewise, the generation model is also fully contextualized and translates templated English questions to natural language ones. Further, the first author conducted extensive experimentation to improve the generation model and thus we use the best version from those experiments. Full details can be found in Appendix A.

The remaining three systems each ablate an aspect of the Full system. The second system is a non-interactive baseline in which the user is simply shown the target question and its decomposition (natural language sub-questions and intermediate answers) and asked if it is correct. Users cannot make edits in this system. This allows us to gauge how informative the decomposition process is and establish a baseline by which to assess the value of interaction. While many papers do this via simulation (Mo et al., 2022; Yao et al., 2019; Yan et al., 2023; Elgohary et al., 2021), we wanted to know *in practice* how helpful it is.

The third system is decontextualized, meaning that neither the parse correction nor the generation models have access to the target question or previous sub-questions in their input. This means that they are mirrors of each other; the parse correction model takes a natural language sub-question (from the user) and produces a templated sub-question which can be translated to a partial SPARQL query using rule-based translation. The generation model, meanwhile, takes that templated sub-question and outputs a natural language version. Mo et al. (2022) show that a contextualized model has better accuracy than a decontextualized one and thus in this study we seek to verify that this is the case. A comparison of the the Full and Decontextualized systems can be seen in Figure 3, which uses the same example as Figure 1.

The fourth system shows the templated sub-questions directly, not using the generation model at all. The other components remain the same as the Full system. We expected that the templates would be harder for users to understand in relation to the target question, based on the difference in BLEU scores between the natural language questions and templates in the INSPIRED dataset when referenced with the target question. The templates have an average of 2.13% for BLEU-1 scores, while the natural language questions score 17.59%. Together, the third and fourth systems allow us to assess the impact of the generation model and the third system allows us to investigate the role of context in both models.

To compare these systems, we set up three head-to-head comparisons of the Full system versus each of the ablated systems. A given crowdworker moves through a pipeline in which they complete 10 dialogues in one system (full or ablated), complete a 5-question survey about those dialogues, then complete 10 more dialogues in the other system and the survey about the second group of 10 dialogues, plus 3 questions that compare the two systems. At the end of each survey there is a com-

Comparison	System	Officially Correct	Inferably Correct	User Deemed Correct
Full vs. Temp	Full	**57.75**	**61.75**	**74.25**†
	Temp	56.50	58.50	65.00
Full vs. Decon	Full	**59.25**	**62.00**	69.75
	Decon	58.75	61.75	**70.75**
Full vs. Non	Full	**59.00**†	**62.00**†	70.75
	Non	16.25	19.25	-

Table 1: Success rates of each system in each comparison (Full vs. Templated, Full vs. Decontextualized, and Full vs. Non-Interactive). † indicates a significant difference at a p-value of less than 0.05 using a paired-sample T-test.

ment box for any other feedback.

We estimated that workers would be able to complete one round of the comparison pipeline in 1 hour, estimating that each dialogue takes 2-3 minutes to complete and the surveys about 2 minutes apiece. We allow workers to do all three comparisons if their work is adequate, though workers never see a given target question more than once.

2.2. Item Selection

For this study, we wanted to balance selecting target questions that are representative of the INSPIRED test set, but also are exceptionally challenging, to thoroughly test and tease out the differences between systems. After conducting a power analysis with an estimated effect size of 0.2, we selected 400 questions that adequately represent all the categories and facets in the INSPIRED dataset (and thus the CWQ dataset on which it is based), but also over-represent characteristics that make queries more difficult for the system to predict. An overview of this selection can be found in Appendix B.

3. Results

The following sections outline the results of the usability study. In Section 3.1, we look at the success of the dialogues in each system and number of edits used. In Section 3.2, we compare the systems based on the surveys that workers completed.

3.1. Dialogue Results

To gauge the differences between the systems, we first look at the success rates of the 400 dialogues in each system. By *success*, we mean that the user is able to reach a correct final answer. This measure, however, is slightly more nuanced than it might first seem. In Table 1, we use three different measures of success. The first is *Officially Correct*, which means that the user confirmed that the sub-questions and answers were correct and the final answer exactly matched the gold answer for that query. The second is *Inferably Correct*, which means that, when the user confirms that the decomposition is correct, the correct answer is easily inferable from the final answer.

> (1) **Target Question:** When did the sports team that plays at Hilltop Park last win a championship?
>
> **Gold Decomposition:**
>
> 1. What team plays at Hilltop Park?
>   *Answer: San Francisco Giants*
> 2. What World Series has that team won?
>   *Answer: 2010 World Series, 2012 World Series*
> 3. When did these events occur?
>   *Answer: 2010 World Series: 2010, 2012 World Series: 2012*
> 4. Of these, which is the most recent?
>   *Answer: 2012 World Series*

In (1), one can easily discern the final answer from the answer of sub-question 2. This is not always the case for superlative questions; often the final two questions are necessary. However, in cases where it is not, we count the dialogue as Inferably Correct.

The third category is *User Deemed Correct*, in which we determine that the user is satisfied with the answer they receive even if it is not correct. We include this category due to a curious trend we observe in which users seem to frequently misunderstand the target question. This is not a new phenomenon; Yao et al. (2019) observe the same issue in the human evaluation of their system. Our suspicions are strengthened by two factors: first, at the end of each dialogue and survey, users are given a text box to give feedback and flag any confusions or concerns, in which they frequently comment on the difficulty of understanding the target question. Second, we strongly emphasize in the tutorial and crowdworker communications that for each dialogue they can perform a maximum of 5 edits and that they must use all 5 if the decomposition and/or answer is incorrect. We find, however, that users frequently do *not* do this and in the majority of those cases, they have reached an answer to the dialogue (meaning that the final answer is not *None*). This seems to indicate that they believe the answer to be correct.

Thus, for the *User Deemed Correct* category, we include dialogues in which either 1) the user does not use all 5 of their available edits but reaches an answer even if it is incorrect, or 2) the user comments that they believe the decomposition to be correct but the final answer is *None*. We suspect that more questions could fall into this category, particularly because the user must answer affirmatively the question *Are the above sub-questions accurate in relation to the complex question?* in order to end the dialogue. However, because we cannot verify the user's interpretations and it is possible that they answer *yes* to this confirmation question for reasons other than believing it to be accurate (such as wanting to move on to the next task), we conservatively define this category. We also omit a small number of cases in which there were generation model errors (18 in total), in case they misled users into thinking a decomposition was correct when it was not. Further explanation and examples of these omissions can be found in Appendix C. The issue of users misunderstanding the target question will be discussed further in Section 3.3.

Table 1 shows that the Non-Interactive system is significantly less successful on all measures than the Full system, demonstrating that interactivity greatly improves performance, which validates results seen in simulation by others (Elgohary et al., 2020, 2021; Yan et al., 2023; Yao et al., 2019). We also see that the Decontexualized system does not perform significantly differently than the Full system, perhaps indicating that context is not as important as we anticipated. Last, we find that the Templated system performs significantly worse than the Full system in the category of User Deemed Correct, which seems to indicate that users found it easier to extract meaning out of the generated sub-questions than the templated ones.

Figure 4 shows the breakdown of how many edits were used for each dialogue in the Templated and Full systems and the success rates (based on the *inferably correct* category). It shows that dialogues tend to be most successful after 1 or 2 edits, and success drops dramatically around 4 edits. The same graph for the Full and Decontextualized comparison, which had no significant differences, is in Appendix D. The comparison between the Full and Non-Interactive systems is not shown because the latter does not allow for edits.

Figure 4 also shows that there are many more 5-edit unsuccessful dialogues in the Templated system than the Full system. This seems to be due in large part to users stopping earlier with the Full system because the questions were easier to understand. Of the dialogues that have 4+ edits, we observe that 36.05% of them in the Templated system are cases where the user was able to get the correct final answer, but continued making edits, compared to 20.56% in the Full system. We assume this is because the templated questions were confusing enough to be misleading. It is noteworthy that our parse correction model was blocked from producing the same corrected question twice, except when the user used the exact same language in their feedback. This means that sometimes, particularly in the Templated system, the user did not recognize that they had the correct answer and continued editing, unable to reproduce the correct answer again. See Appendix E for an example.

Comparison	System	Effort Metric	Difference
Full vs Temp	Full Temp	0.2877 0.2358	$0.0519^{\dagger}$ p = 0.0054
Full vs Decon	Full Decon	0.2780 0.2795	0.001498 p = 0.9463

Table 2: Efficiency of dialogue systems. The table summarizes the effort metric comparisons between Full and Templated systems, and Full and Decontextualized systems. Statistical significance is indicated with †, which represents a p-value less than 0.05 using an approximate randomization test.

In evaluating the efficiency of the dialogue systems and assessing the relative worth of each edit made, we employed a metric of effort defined as the ratio of successful (inferably correct) dialogues to the total number of edits made, regardless of the dialogue's outcome. These numbers can be seen in Table 2. To rigorously test the significance of this observed difference, we employed approximate randomization testing. Details of this can be found in Appendix F.

The significant difference in effort between the Full and Templated systems strongly suggests that the type of system indeed impacts the efficiency of dialogues. The difference between the Full and Decontextualized system was non-significant, however, again suggesting that the systems did not perform substantially differently. This result seems to demonstrate the superior efficiency of the Full system over its Templated counterpart in facilitating successful dialogues with fewer edits.

This result, along with the *User Deemed Correct* difference seen in Table 1, provides evidence that the natural language questions are easier to understand than the templated questions. This tracks with our assumption that the natural language questions in INSPIRED more closely resemble the target question.

3.2. Survey Results

Table 3 shows the results of the surveys that each worker completed (2 per worker, per comparsion). The top section (questions 1-5) shows the average scores on the survey that workers completed after every batch of 10 dialogues in a given system. These results give us insights into user impressions of the different systems and we can see that they tend to cluster around the middle values. There are no significant differences between the Full and Decontextualized systems and only one significant result between the Full and Templated systems, for *I felt confident using the system*, in favor of the

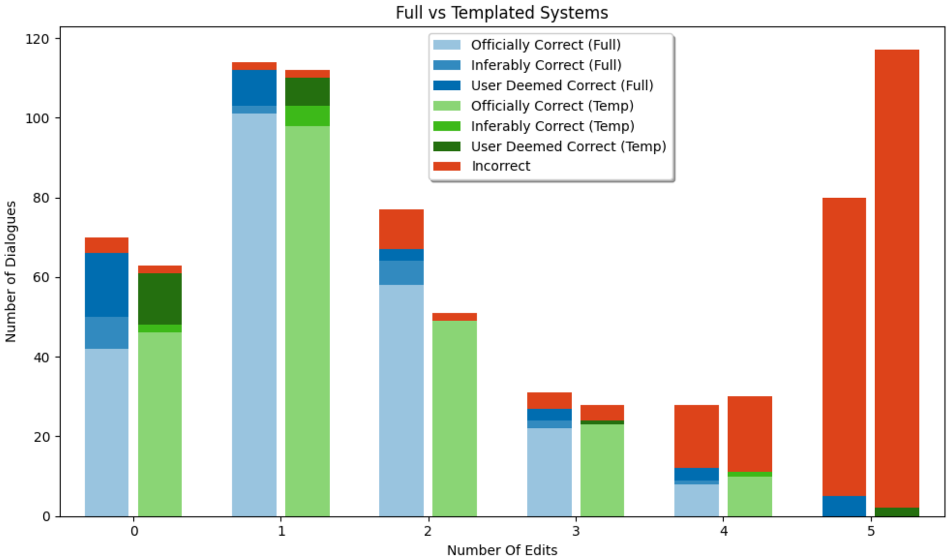


Figure 4: A comparison of the success rates for the Full vs. Templated systems (using three different categories – officially correct, inferably correct, and user deemed correct) and the number of edits attempted by the user.

	Question	**Comparison 1** **Full**	 **Temp**	**Comparison 2** **Full**	 **Decon**	**Comparison 3** **Full**	 **Non**
1	I thought the system was easy to use.	2.17	2.35	2.10	2.05	2.11	**1.67**†
2	I would imagine that most people would learn to use this system quickly.	2.24	2.56	2.17	2.05	2.16	**1.71**†
3	I felt confident using the system.	**1.93**†	2.42	1.98	1.95	2.11	1.93
4	The system was usually able to answer the questions correctly.	2.43	2.49	2.24	2.19	**2.32**†	3.00
5	It took a reasonable amount of time to complete a dialogue.	2.29	2.19	2.10	2.00	2.25	**1.67**†
6	Which of the two systems was easier to use?	58.97	41.03	43.90	56.10	30.00	**70.00**†
7	Which system answered more questions correctly?	56.41	43.59	53.66	46.34	**67.50**†	32.50
8	Which system gave you more confidence that the final answers were correct?	58.97	41.03	43.90	56.10	**70.00**†	30.00

Table 3: Survey results for each comparison. Users completed a survey of the first 5 questions after completing 10 dialogues in a given system. They answered using Likert scale responses: 1) Strongly Agree, 2) Agree, 3) Disagree, 4) Strongly Disagree (Lower scores are better). Significance was determined using a paired sample t-test. The last three questions were answered after completing a full comparison pipeline (10 dialogues in each system) and users selected only one of the two options for these questions. Significance was determined using a binomial test. † indicates a significant difference with a p-value of less than 0.05.

Full system. This indicates that the templated subquestions increased users' confusion and eroded confidence. Unsurprisingly, there are significant results in the Full vs. Non-Interactive systems comparison. For the first two questions (*I though the sytem was easy to use* and *I would imagine that most people would learn to use this system quickly*) and the final one (*It took a reasonable amount of time to complete a dialogue*), the difference is in favor of the Non-Interactive system, but this likely has a great deal to do with the lack of interaction. Users merely had to answer the question "Are these sub-questions correct in relation to the Target Question?” and leave comments if necessary, making the task much easier and less time-consuming than the other systems. It is notable that question 4 had a significant result in favor of the Full system, again suggesting that interactivity greatly boosts system accuracy and user confidence in the final answer.

The bottom section of Table 3 (questions 6-8) shows the results of the comparative survey questions, which users answered after completing 10 dialogues in each system (Full vs. ablated system). Significant results only appear in the Full vs. Non-Interactive comparison for the reasons given above.

3.3. Discussion

This usability study revealed results that could be useful for semantic parsing research, in particular about the importance of interactivity, the benefit of generated questions over templated ones, and users’ ability to successfully navigate such a system. However, there were fewer significant results than we anticipated, in particular between the Full and Decontextualized systems. This discrepancy, while somewhat disappointing, is elucidating in many respects. First, this demonstrates the difference between *simulation* and *real use*. Mo et al. (2022) use BART-large (Lewis et al., 2020) and QGG (Lan and Jiang, 2020) to simulate user feedback in their framework and find that they can get exact-match accuracy up to 73.5%, which is 14.25% higher than the best accuracy in our usability study (see Table 1). It is important to note, however, that their metric is calculated over the entire test set of INSPIRED, while ours is over a subset of 400 questions that were selected for their difficulty. They also find that using context in the model inputs leads to accuracy gains in simulation, a trend that did not bear out with human users in our study. Other works, such as (Elgohary et al., 2020, 2021; Yan et al., 2023; Yao et al., 2019), also show substantial gains using simulated users.

Our usability study shows how using a human user changes the overall success of the system. Our users, of course, are not *real* users, meaning that they were not using our system to answer their own questions, but rather questions given to them. There are two main reasons for this design choice; the first is that we wanted to directly test different systems on the same questions for the sake of a clear comparative analysis. Secondly, we wanted to lower the burden on the crowdworkers, who might have had difficulty coming up with their own complex questions that could be answered by the knowledge base.

One major factor to this difference in simulation and human users is the amount of noise that can be found in the dataset. While the INSPIRED dataset underwent a fair amount of cleaning to ensure that the target questions matched their SPARQL queries, problems still remained. The questions, which came from the COMPLEXWEBQUESTIONS dataset (Talmor and Berant, 2018), are often very challenging to understand. They were created from the WEBQUESTIONSSP dataset (Yih et al., 2016) by combining two simple questions together with a rule-based method and then having crowdworkers rephrase them into a single, more natural-sounding question. This results in complex questions that are often quite unnatural and unlikely to be asked by a real human user, which in turn makes them more difficult to understand for our crowdworkers. We notice too a number of errors in the CWQ dataset – a mismatch between the target question and the meaning of the associated SPARQL query. This does not, however, seem to prevent gains in terms of parsing accuracy – systems such as Mo et al. (2022) and Niu et al. (2023) find that they are still able to make significant progress on correcting parses despite such errors.

In our study, however, users report frequently that the target questions were unnatural, incomprehensible, or unclear, demonstrating that such errors have a much bigger impact when using human users and in natural language generation settings. It also did not help that we chose the hardest questions available, which was intended to demonstrate that the system works well even in these cases. While this does appear to be the case the majority of the time, we underestimated users’ fatigue and frustration with simply *understanding* the target question. Yao et al. (2019), as mentioned, notice similar problems in their user evaluation.

4. Lessons Learned: Crowdsourcing

The issue of the target questions appears to cause substantial problems in the overall accuracy of the systems, but also in terms of user investment in the task. We know based on worker feedback that these issues led to frustration and a slow-down of the tasks. We also hypothesized that workers’ performance might improve over time, which did not occur. Looking at the first 10 and last 10 dialogues in the Full system of the 41 workers who did at least two comparisons, we find that the first 10 have an average success rate of 65.37% and the last ten have a rate of 62.44%.

Cultivating worker investment in a crowdsourcing task is a challenging but very important problem to consider. We carefully planned the crowdsourcing effort with several factors in mind, including ensuring that workers were compensated fairly and that the task took a reasonable amount of time to complete. We decided on a base rate for each dialogue according to the minimum wage of our state and the amount of time on average a dialogue might take, estimated from internal testing. However, because users needed to read a tutorial and complete a qualification quiz prior to being granted access to our task, we also ensured that they were compensated for that time spent by granting them a bonus once they complete the whole pipeline, which we anticipated taking at most 1 hour. If a worker completed a full comparison they collected a total of $15. Because workers could complete all three comparison pipelines (assuming there were dialogues available), they could receive a maximum of $45. This payment rate is significantly higher than what is usually available on MTurk.

It is important to note, however, that using MTurk presents a great number of challenges. First, it is not really designed for dialogue tasks and external software is needed to do this. To this end, we utilized the ParlAI and Mephisto platforms (Miller et al., 2017; Urbanek and Ringshia, 2023), which are designed to run more complex tasks (particularly dialogue) than the regular MTurk platform. However, we find that the connection between the platforms can be very brittle and we encountered both a number of connectivity problems for our users.

Secondly, the issue of bots on MTurk poses a major challenge. Careful design went into preventing bots from accessing our task and eliminating ones that were able to get through, while not making the task too cumbersome for real crowdworkers. For example, we implemented a wait time for repeated attempts on the qualification quiz to prevent repeated random guessing on the questions until the correct answers were found. We also monitored how long it took a user to complete a task and manually reviewed tasks that were notably shorter than average. We also reviewed cases where there were no edits on several tasks in a row and where the same edit was submitted for every turn.

Further complicating this issue is that it is very important to *not* reject crowdworkers' completed tasks unless we are absolutely certain that it came from a bot. Due to the structure of the interaction between crowdworkers and requesters on MTurk, workers are unduly harmed by the rejection of work – it can irreparably damage their reputation on the platform and prevent them from being able to get future work. Thus, in cases of inadequate performance, if there is any doubt whether the worker is a bot or a human, we err on the side of caution and accept the work, but prevent that MTurk user from completing more tasks. We discarded the data from our analysis. In our experience, this problem appears to be worsening over time.

Thirdly, we suspect that because MTurk tasks are typically shorter and less involved, workers might have been less inclined to do our task in the first place and/or felt incentivized to go through the task as quickly as possible, which might have resulted in a decrease in performance.

Comparison	System	Officially Correct	Inferably Correct	User Deemed Correct
Full vs. Temp	Full	**72.2**	**77.8**	**81.5**
	Temp	68.5	68.4	68.4
Full vs Decon	Full	**63**	**65.2**	**67.4**
	Decon	52.9	52.9	54.9
Full vs. Non	Full	**63.0**	**68.5**	76.0
	Non	18.0	24.5	-

Table 4: Percentage success rates of a smaller, possibly more dedicated, working group of 6 crowdworkers that participated in our task. These numbers represent success out of 60 items (10 per worker).

This method of human evaluation stands in contrast with a method that seems increasingly typical: hiring a dedicated, small team of evaluators and specifically training them to do the task. There are obvious benefits to this approach: it is much easier to ensure that your workers have understood the task, it is possible to conduct more complicated/involved tasks, and you can communicate directly and easily as problems arise. It is thus much easier to ensure quality results.

However, drawbacks of this include the difficulty of finding such workers, the added expense involved, and – importantly – the possible detraction from the generalizability of the results. We opted to use crowdworkers in large part because we wanted to ensure that the task was easily understandable and doable for an average person without a great deal of guidance. However, we find that this benefit seems to be getting washed out by the difficulties of conducting a crowdsourcing task on MTurk. This finding is supported by looking at the performance of a particular subset of our crowdworkers. On a previous crowdworking task, the first author established a relationship with a "working group" on MTurk. Such groups have organically formed on online spaces; workers collaborate to share information about tasks they can work on, requesters, etc. This particular group had produced excellent work in a previous task, so the first author reached out to them again and invited them to work on this one. Six of them agreed and did all three comparisons. By looking at their work in isolation, we see that the results look much more like what we expected. The success rates of the working group can be seen in Table 4. This suggests that a more dedicated group might have yielded clearer results than a general crowdsourcing task.

5. Related Work

As mentioned, this usability study is, to our knowledge, the most thorough one conducted on an interactive semantic parsing system. There are a number of systems developed similar to that of Mo et al. (2022) though mostly in the realm of text-to-SQL parsing instead of KBQA. This includes recent work such as Chen et al. (2023), who use CodeBERT as an error detector for SQL parsing tasks, which they envision being incorporated into an interactive system. Also included is Eyal et al. (2023) which uses a decomposition strategy for SQL queries, Stengel-Eskin and Van Durme (2023) who leverage confidence measures in an interactive SQL parsing system to help annotators make corrections, and Yan et al. (2023) who simulate natural language feedback.

In terms of KBQA, Xu et al. (2023) fine-tune LLaMA (Touvron et al., 2023) and pair it with GPT-3 to produce parses, while Niu et al. (2023) use a decomposition strategy to improve a parser's understanding of natural language. However, these systems stand in contrast to that of Mo et al. (2022) as neither of these systems are interactive.

Narechania et al. (2021), Ning et al. (2023), Yao et al. (2019), and Tian et al. (2023) all create interactive systems for text-to-SQL parsing and conduct user studies for their systems using in-house participants (students or employees from their institutions), many of which have experience with SQL. Additionally, Tian et al. (2023) and Narechania et al. (2021) allow their participants to ask questions about the task in real time. While these studies have useful insights into how their systems work, they may be more limited in their ability to generalize to average users who might eventually use the system in real contexts.

Tian et al. (2023) and Narechania et al. (2021) also use rule-based template translations directly for simplicity. We observe that SQL queries are easier to translate with templates because they contain more keywords, whereas SPARQL queries emphasize relationships between entities. In SQL, relationships between entities are specified via JOIN operations using foreign keys; spelling these out with templates yields rather unnatural, low-level descriptions.

6. Conclusion

This paper underscores the crucial role of interactivity in semantic parsing for knowledge-based question answering systems, demonstrating through a comparative study and user feedback the heightened effectiveness and satisfaction derived from allowing participants to directly interact with the query translation process. It emphasizes the need for systems to convey complex queries in clear language for user corrections, thus lowering barriers to knowledge base access. The study advocates for human-user interactions and the outlines the potential benefits and drawbacks from crowdsourcing.

Future work involves leveraging advanced Large Language Models (LLMs) like ChatGPT, LLaMA, and Gemini to enhance parsing accuracy and user interaction quality (Achiam et al., 2023; Team et al., 2023; Touvron et al., 2023). These models could provide finer interpretations of user queries, thereby enhancing precision. Additionally, developing more intuitive interfaces and feedback mechanisms, coupled with testing systems using real user queries, could offer significant insights into system usability and performance in real-world scenarios. This strategy aims to better meet user needs and understand system efficacy in actual use cases.

7. Ethical Considerations

Approval from the Institutional Review Board (IRB). Before starting our crowdsourcing usability study, we secured approval from the Institutional Review Board (IRB) at our institution. This process classified our data collection as Exempt Research, indicating that participation posed minimal or no risk to our human subjects. We did not gather any personal information from participants, except for basic demographic details like their native language, to confirm their eligibility for the task. No identifiable information was collected. Additionally, all participants were required to review and consent to an informed consent form prior to engaging in the study. The identities of MTurk crowdworkers were also anonymized automatically by the platform.

Paying Crowdworkers. To ensure quality data collection and fair treatment of crowdworkers, we meticulously formulated our compensation strategy for the MTurk task. Following internal testing, we assessed the average time needed for a task and adjusted the compensation accordingly to meet the minimum wage standards in our state. This adjustment led to a payment of 40 cents per task and $1 per survey. The survey payment was designed to act as both payment and a small bonus. Moreover, to encourage ongoing participation and compensate for the time invested in tutorials and qualification tasks, we provided $5 bonuses upon the completion of each stage in the pipeline.

8. Acknowledgments

We would like to thank the reviewers for their time, careful consideration, and helpful comments. We would also like to thank the Clippers discussion group at The Ohio State University for their feedback. Marie-Catherine de Marneffe is a research associate of the Fonds de la Recherche Scientifique – FNRS. Huan Sun and Lingbo Mo were partly supported by NSF CAREER #1942980.

9. Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Austin, Texas. Association for Computational Linguistics.

Shijie Chen, Ziru Chen, Huan Sun, and Yu Su. 2023. Error detection for text-to-SQL semantic parsing. *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. 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, Online. Association for Computational Linguistics.

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.

Ben Eyal, Moran Mahabi, Ophir Haroche, Amir Bachar, and Michael Elhadad. 2023. Semantic decomposition of question and SQL for text-to-SQL parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13629–13645, Singapore. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

Yunshi Lan and Jing Jiang. 2020. 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, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Xintong Li, Symon Stevens-Guille, Aleksandre Maskharashvili, and Michael White. 2021. Self-training for compositional neural NLG in task-oriented dialogue. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 87–102, Aberdeen, Scotland, UK. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Will Monroe and Christopher Potts. 2015. Learning in the rational speech acts model. In *Proceedings of 20th Amsterdam Colloquium, Amsterdam.)*.

Arpit Narechania, Adam Fourney, Bongshin Lee, and Gonzalo Ramos. 2021. DIY: Assessing the correctness of natural language to SQL systems. In *26th International Conference on Intelligent User Interfaces*, pages 597–607.

Zheng Ning, Zheng Zhang, Tianyi Sun, Yuan Tian, Tianyi Zhang, and Toby Jia-Jun Li. 2023. An empirical study of model errors and user error discovery and repair strategies in natural language database queries. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 633–649.

Yilin Niu, Fei Huang, Wei Liu, Jianwei Cui, Bin Wang, and Minlie Huang. 2023. Bridging the gap between synthetic and natural questions via sentence decomposition for semantic parsing. *Transactions of the Association for Computational Linguistics*, 11:367–383.

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. 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 4060–4067, Minneapolis, Minnesota. Association for Computational Linguistics.

Elias Stengel-Eskin and Benjamin Van Durme. 2023. Did you mean...? confidence-based trade-offs in semantic parsing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2621–2629, Singapore. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. 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.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Yuan Tian, Zheng Zhang, Zheng Ning, Toby Li, Jonathan K. Kummerfeld, and Tianyi Zhang. 2023. Interactive text-to-SQL generation via editable step-by-step explanations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16149–16166, Singapore. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jack Urbanek and Pratik Ringshia. 2023. Mephisto: A framework for portable, reproducible, and iterative crowdsourcing. *arXiv preprint arXiv:2301.05154*.

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.

Silei Xu, Shicheng Liu, Theo Culhane, Elizaveta Pertseva, Meng-Hsi Wu, Sina Semnani, and Monica Lam. 2023. Fine-tuned LLMs know more, hallucinate less with few-shot sequence-to-sequence semantic parsing over Wikidata. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5778–5791, Singapore. Association for Computational Linguistics.

Hao Yan, Saurabh Srivastava, Yintao Tai, Sida I Wang, Wen-tau Yih, and Ziyu Yao. 2023. Learning to simulate natural language feedback for interactive semantic parsing. *arXiv preprint arXiv:2305.08195*.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. 2019. Model-based interactive semantic parsing: A unified framework and a text-to-SQL case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5447–5458, Hong Kong, China. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

A. Generator Development

A.1. Motivation

In order for a user to know when a parse is correct, the generated question needs to accurately and clearly reflect the content of the parse, or meaning representation (MR), that it represents. Mo et al. (2022) experimented with various generation models, finding that BART-large performed best and that including dialogue context in the model input improved performance. Furthermore, they find that including templated English versions of the SPARQL queries, which were created to help their crowdworkers understand the meaning of queries in order to rephrase them, also boosted performance.

As mentioned in Section 2.1, we improve the generator by transforming the partial SPARQL query MRs into templated English using a rule-based translation, following the example of Kale and Rastogi (2020). This improvement boosts the generator's performance, though errors still occur. In an analysis of 100 random generated questions, the

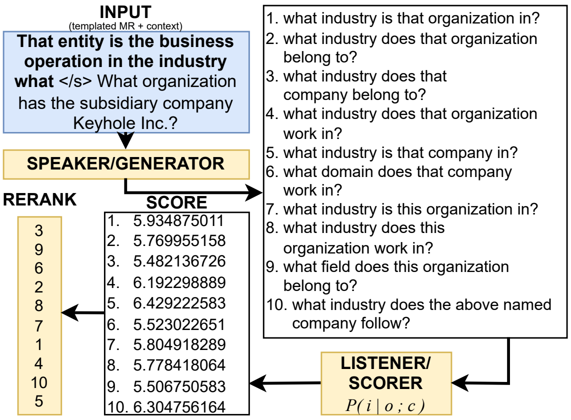
	Specificity (8)	**Fluency (1)**	**Correctness (2)**
Templated	What is the **composer** of the **music** “My Baby Understands”?	What is the jurisdiction of the governmental body **Hawaii** House of Representatives?	Of which, what is the administrative area of the type UK constituent country?
Human-Written	Who **wrote** the **song** “My Baby Understands”?	Which government includes the **Hawaii** House of Representatives?	Of the above listed, which one **is a UK constituent** country?
Generated	Who **wrote** “My Baby Understands”?	What state has the **Hawaraii** House of Representatives as part of its government?	Which of these countries is where **that UK constituent** country is in?

Table 5: Three instances of errors in an analysis of 100 random generated sub-questions compared to human-written and templated versions. The bold text indicates meaningful differences between them. The numbers in parenthesis indicate the percentage of error cases out of the 100 items examined.

first author finds that errors occur in 11% of them. These errors can be grouped into three categories: specificity, correctness and fluency.

Examples of these three error types can be seen in Table 5. In this table, one can see that under the *specificity* category, the generated question does not mention that the writer is a composer nor that the piece is a song or piece of music, making it less exact in comparison to both the templated question and human-written (gold) question, taken from the INSPIRED dataset. Specificity errors occur in 8 out of 100 instances. *Fluency* errors occur when the generated question has nonsensical grammar or spelling, such as spelling *Hawaii* as “Hawaraii”, occurring only once. Lastly, *correctness* errors occur when the generated question does not produce the correct meaning for the LF, which occurs twice.

Though these errors may seem small, they can have a big impact on the usability of the overall system. The specificity error in Table 5, for example, can be interpreted in a number of ways. The question *Who wrote X?* is not specific enough for the user to know that the underlying MR is correct or not, as this could correspond to other closely related meanings such as the writer of a book, lyrics, music, a film, etc. Thus the underlying MR could be incorrect but the user would have a difficult time identifying *how* it is incorrect, in turn making it difficult to provide feedback. The ambiguities and errors that can occur in generated questions are thus important problems to be addressed.



That entity is the business operation in the industry what </s> What organization has the subsidiary company Keyhole Inc.?

1. what industry is that organization in?
2. what industry does that organization belong to?
3. what industry does that company belong to?
4. what industry does that organization work in?
5. what industry is that company in?
6. what domain does that company work in?
7. what industry is this organization in?
8. what industry does this organization work in?
9. what field does this organization belong to?
10. what industry does the above named company follow?

Figure 5: An example of the protocol for experiment 1.

A.2. Rational Speech Acts (RSA) Modeling

To this end, we employ the linguistic framework of Rational Speech Acts (RSA), which views dialogue as a recursive reasoning process between speakers and listeners to convey meaning (Monroe and Potts, 2015; Andreas and Klein, 2016). The underlying idea is that a listener is reasoning about what “state of the world” is likely to be given that a cooperative speaker produced a particular question, while the speaker is reasoning about how a listener is most likely to interpret the question they produce. Though RSA models focus primarily on the pragmatics of dialogue and thus the information is communicated beyond what is literally said, the concept of recursive reasoning between a speaker and listener can have useful applications for the conveyance of literal meaning in questions by a text generation model.

The work of Shen et al. (2019) demonstrates this effectively by drawing on RSA methods to improve text generations by modeling a listener via *information preservation* (reconstructor-based) and *modeling of distractors* (distractor-based).

A.3. Experiment 1: Reconstructor-Listener

We adapt the reconstructor-based approach of Shen et al. (2019) to this work, using a base speaker model, S^0, to translate MRs to natural language questions and using beam search to output a 10-best list. We then use a reverse model, L^R to represent the listener, which is trained to predict (or reconstruct) MRs from the natural language outputs of the speaker model. We use L^R as a scorer, outputting negative log-likelihood scores for the original MR given each question in a 10-best list from S^0: $L^R(i|o)$.

	% of items
Equal Preference	
1. Listener rank + Speaker rank	
2. Select lowest combined rank (best)	
Listener Preference	
1. If Listener rank = 1 and Speaker rank ≤ 3, select	28.7
2. Elif Listener rank = 2 and Speaker rank ≤ 3, select	25.7
3. Else:	45.6
For Listener ranks 1-5: Listener rank + Speaker rank select lowest combined rank	
Speaker Preference	
Opposite of Listener Preference	32.8
	25.7
	41.5

Table 6: The three methods of selecting a question in each preference model, using rankings. The right side of the table shows the percentages of the items that were selected using that step.

	Best Option	Breakdown
Speaker	17	C 6 S 9 F 2
Listener	**19**	C 6 S 10 F 3
Same	58	
Neither	6	

Model	Best Option
Speaker	75
Listener	77
Blend (EP)	79
Blend (SP)	80
Blend (LP)	**83**
Oracle	94

Table 7: (Left) Comparison between pairs of #1 Listener utterances and #1 Speaker utterances of 100 items. Each error was categorized into one of three categories, as seen in the "Breakdown" column. C refers to *correctness*, S refers to *specificity*, and F refers to *fluency*.

Table 8: (Right) Comparison of generated questions selected by different ranking methods on 100 items.

We then use these scores to rerank the questions in order of preference by the listener model. Figure 5 demonstrates this process using an example MR.

In order to assess the differences between the listener model ranking preferences and the speaker model preferences, we select 100 MRs and the corresponding pair of the top ranked speaker and listener questions. We select these items by calculating Levenshtein edit distance between them, filtering out duplicate MRs, and selecting the 100 pairs with the largest edit distance score. The goal of this is to find the cases in which the listener model preference and speaker model preference differ the most, which seems to also signal the cases that are most challenging for the models. We mask the labels of the two questions and randomize them, then manually compare based on differences in correctness, specificity, and fluency (see Table 5 for examples of these criteria). Table 7 shows the results of this analysis. The majority of the examples were of equal quality, though in six instances, neither question reflected the meaning of the original MR. The Listener #1 questions were slightly more successful than the Speaker #1 questions in expressing the meaning of the MR, though the non-negligible success of the Speaker model seems to indicate that there could be benefit to balancing the ranks of the two models.

To this end, we experiment with various methods of blending the scores (as Shen et al. (2019) does) or rankings and find that using rankings has the best results. We experiment with equally weighting the ranks of the two models and preferring one over the other. Table 6 shows the procedure for each of these strategies.

A.4. Experiment 2: Self-Training

Because the reconstruction-based approach is computationally heavy and unrealistic to use at run time in real dialogues due to latency issues, we experiment with self-training to create a single generation model that is informed by this process.

In the COMPLEXWEBQUESTIONS dataset, there are 24,147 complex questions in the training set that did not get used in the INSPIRED dataset. Thus, we can transform these into our templated representations, giving us more than 50,000 MRs without natural-language counterparts. Using the strategy above, we use the base speaker model to generate a 10-best list of sub-questions for each MR, use the reconstructor model to score them, then use the best blending strategy (listener preference) to select the best single question. We then train a new generation model on this “silver” data for 10 epochs, then 4 epochs on the gold data (from the INSPIRED dataset). We then generate a single sub-question for each of of the 50k MRs using this new model. We iterate this process a second time but find that performance seems to plateau. This is unsurprising; previous work has found that self-training performance usually plateaus after very few iterations (Li et al., 2021). Table 9 shows those results. In this table we use an automatic metric we call *reconstruction accuracy*, which is the percentage of times that the listener model could recover the original MR when given the generated question as input. This gives us a measure of how well the meaning of the MR is preserved in the generated question.

There are a few noteworthy points about these results. First, the speaker, listener, and blend numbers have decreased a bit in this evaluation. The reason for this is basically that, in some cases, even though the question they chose was good, the self-trained model came up with a question that was even better, meaning that it was more specific or context-aware, for example. The self-train number

Model	**Reconstruction Accuracy (%)**	**Manual Evaluation (%)**
Base Speaker Model	80.86	71
Blended Ranking	81.74	75
Self-Trained	**86.50**	**96**
Self-Trained Round 2	85.73	94

Table 9: Comparison of the base speaker model (S_0, the best blended rank model (listener preference), and two iterations of the self-trained model.

is also higher than the oracle number in Table 8. This basically means that in two cases, the self-trained model's sub-question was a valid one when the base speaker model did not have a valid question in its list of 10 possibilities. Table 10 shows these two cases.

Because we see substantial gains using the self-trained model at one iteration, we employ this model as our generator in this work.

#1 utterance	**Example 1**	**Example 2**
Template (MR)	What is the **composer** of the music [my baby understands]?	What is the jurisdiction of the governmental body [**Hawaii** House of Representatives]?
Speaker	Who **wrote the song** "my baby understands"?	What state has the **Hawaraii** House of Representatives as part of its government?
Listener	"My baby understands" is **credited to** whom?	In what state's government can you find the **Hawaraii** House of Representatives?
Blend	Who **wrote** "my baby understands"?	In what state's government can you find the **Hawaraii** House of Representatives?
Self-Trained	Who is the **composer** of the **song** "my baby understands"?	What state has the **Hawaii** House of Representatives as part of its government?

Table 10: The two examples where the self-trained model was able to produce a better question than any of the 10-best items produced by the speaker model. Red indicates problem areas, including ambiguity and misspellings, and green indicates contrasting successful areas.

Question Types	**% INSPIRED (N = 3441)**	**% User Study (N = 400)**
Composition	43.30	48.00
Conjunction	45.13	37.25
Comparative	6.02	8.25
Superlative	5.55	6.5
Question Facets		
Filter	13.02	15.00
Restriction	22.38	44.75
Union	1.60	5.00
Sparse Predicates	10.61	15.00
Double Restrictions	0.44	1.00
Required Edits		
0	56.81	7.50
1	28.74	65.25
2	13.28	25.75
3	1.08	1.50
4	0.09	0

Table 11: Breakdown of question types, facets, and number of required edits in the INSPIRED dataset and selected questions for the user study (in percentages).

B. Question Selection

There are 4 main question types: composition, conjunction, comparative, and superlative. There are also many facets possible for a given SPARQL query/question pair, including filters, restriction predicates, and unusual compositions such as union types. We also identify queries that contain a sparse predicate, meaning it appears three or fewer times in the training set of INSPIRED. Note that *predicates* refer to the relations between entities in the knowledge base. See Mo et al. (2022) for explanations of the various question types and facets. Table 11 shows the breakdown of the INSPIRED test set and the 400 selected items within that set used for the user study.

C. Generation Errors

Figure 6 and 7 show examples of errors made by the generation model. In Figure 6 (which is using the Full system), you can see that the error is quite subtle; the parse correction model predicted the question *Of which, what is the film with **a story by** [Ethan Coen]?* instead of *Of which, what is the film **written by** [Ethan Coen]?* These are in fact different expressions; a film might be based on a story by an author, but the screenplay written by a different author. Because these are very closely related concepts (that are not even made particularly clear by the different SPARQL predicates – *film.film.story_by* versus *film.film.written_by*), the generation model seems to confuse their meanings. This leads to a difference in the underlying SPARQL and template and the natural language question that the user sees, leading them to believe that the decomposition is correct when it is not.

Figure 7 demonstrates an error by the initial decomposition (in this case using the Decontextualized system) in which the entity name is misspelled, making a sub-question seem incorrect when it is in fact correct.

The first author conducted an error analysis of the dialogues that met the criteria of the *User Deemed Correct* category described in Section 3.1. In those 125 dialogues, 18 were removed due to generation errors. 12 occur in the Full system (across all comparisons, N=96) and 6 occur in the Decontextualized system (N=29). It was considered an error if the natural language question was ambiguous or too vague, for example if it selected the word *location* when the predicate specified *country*, if it contained any misleading language or if it better expressed a different (often closely related) predicate instead of the one it was supposed to be representing (see Figure 6), or if there were fluency or grammatical problems (see Figure 7).

D. Full vs. Decontextualized System Comparison

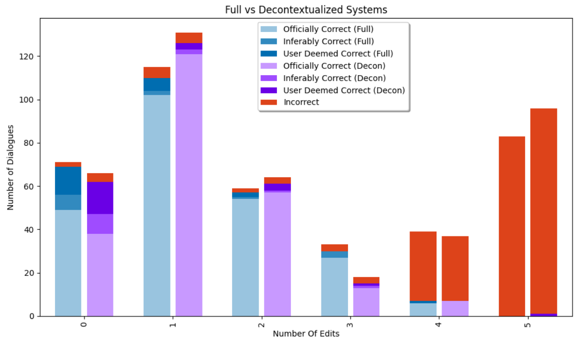
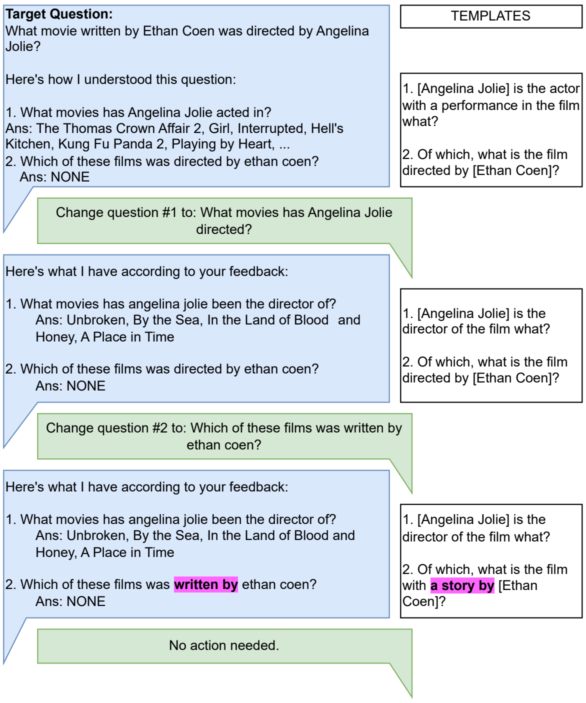


Figure 8: A comparison of the success rates for the Full vs. Decontextualized systems (using three different categories – officially correct, inferably correct, and user deemed correct) and the number of edits attempted by the user.

E. Appendix: User Errors in Templated System

Figure 9 shows a dialogue in the Templated system in which a user achieves the correct decomposition and final answer, but continues to make edits. We assume that this occurs when the user is unsure whether or not a sub-question is correct due to a confusing template.



Target Question:
What movie written by Ethan Coen was directed by Angelina Jolie?

Here's how I understood this question:

1. What movies has Angelina Jolie acted in?
Ans: The Thomas Crown Affair 2, Girl, Interrupted, Hell's Kitchen, Kung Fu Panda 2, Playing by Heart, ...
2. Which of these films was directed by ethan coen?
Ans: NONE

TEMPLATES

1. [Angelina Jolie] is the actor with a performance in the film what?

2. Of which, what is the film directed by [Ethan Coen]?

Change question #1 to: What movies has Angelina Jolie directed?

Here's what I have according to your feedback:

1. What movies has angelina jolie been the director of?
Ans: Unbroken, By the Sea, In the Land of Blood and Honey, A Place in Time

2. Which of these films was directed by ethan coen?
Ans: NONE

1. [Angelina Jolie] is the director of the film what?

2. Of which, what is the film directed by [Ethan Coen]?

Change question #2 to: Which of these films was written by ethan coen?

Here's what I have according to your feedback:

1. What movies has angelina jolie been the director of?
Ans: Unbroken, By the Sea, In the Land of Blood and Honey, A Place in Time

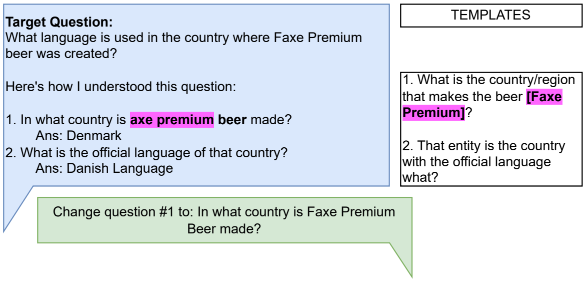
2. Which of these films was **written by** ethan coen?
Ans: NONE

1. [Angelina Jolie] is the director of the film what?

2. Of which, what is the film with **a story by** [Ethan Coen]?

No action needed.

Figure 6: Generation error by the Full System.



Target Question:
What language is used in the country where Faxe Premium beer was created?

Here's how I understood this question:

1. In what country is **axe premium** beer made?
Ans: Denmark
2. What is the official language of that country?
Ans: Danish Language

TEMPLATES

1. What is the country/region that makes the beer **[Faxe Premium]**?

2. That entity is the country with the official language what?

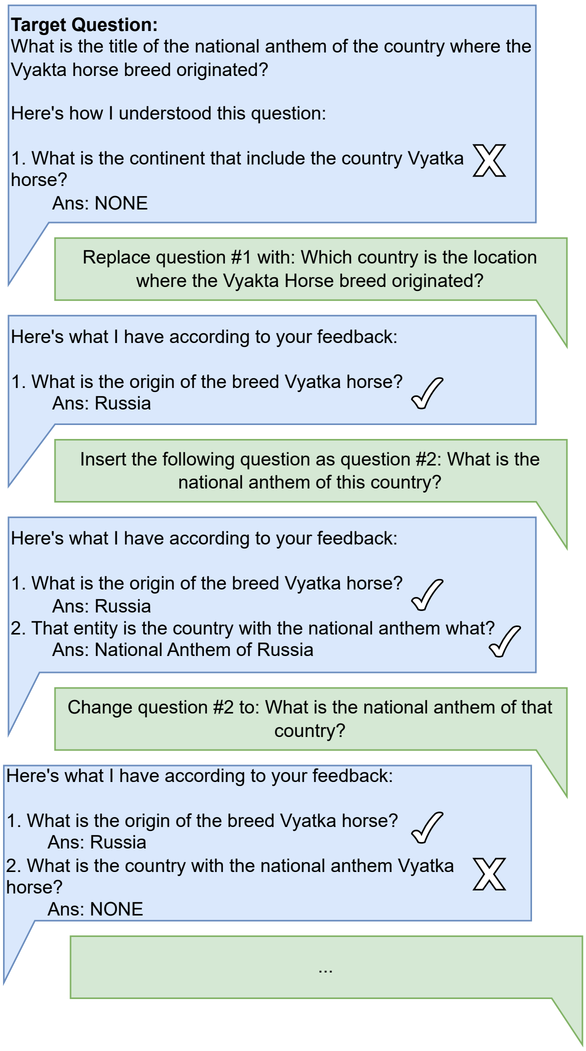
Change question #1 to: In what country is Faxe Premium Beer made?

Figure 7: Generation error by the Decontextualized system.

F. Approximate Randomization Test for Significance

To rigorously test the significance of the observed difference in our metric of effort, we employed approximate randomization testing which is a non-parametric statistical test that does not assume a specific distribution of the data. In this case, this method involves virtually flipping a coin to decide whether to swap the success and edit counts between systems for each dialogue. Through this random reassignment, we generated new sets of data under the null hypothesis that system type does not affect the efficiency metric.

After 10,000 such trials, the proportion of randomized trials yielding a difference in effort scores greater than or equal to the observed difference of 0.05192 was 0.0054 for the Full vs. Templated system. This p-value indicates that the observed difference in effort between the Full and Templated systems is statistically significant, strongly suggest-



Target Question:
What is the title of the national anthem of the country where the Vyakta horse breed originated?

Here's how I understood this question:

1. What is the continent that include the country Vyatka horse? ✗
Ans: NONE

Replace question #1 with: Which country is the location where the Vyakta Horse breed originated?

Here's what I have according to your feedback:

1. What is the origin of the breed Vyatka horse? ✓
Ans: Russia

Insert the following question as question #2: What is the national anthem of this country?

Here's what I have according to your feedback:

1. What is the origin of the breed Vyatka horse? ✓
Ans: Russia
2. That entity is the country with the national anthem what? ✓
Ans: National Anthem of Russia

Change question #2 to: What is the national anthem of that country?

Here's what I have according to your feedback:

1. What is the origin of the breed Vyatka horse? ✓
Ans: Russia
2. What is the country with the national anthem Vyatka horse? ✗
Ans: NONE

...

Figure 9: Example of an instance in which the template confused a user into thinking a decomposition was incorrect when it was not.

ing that the type of system indeed impacts the efficiency of dialogues.