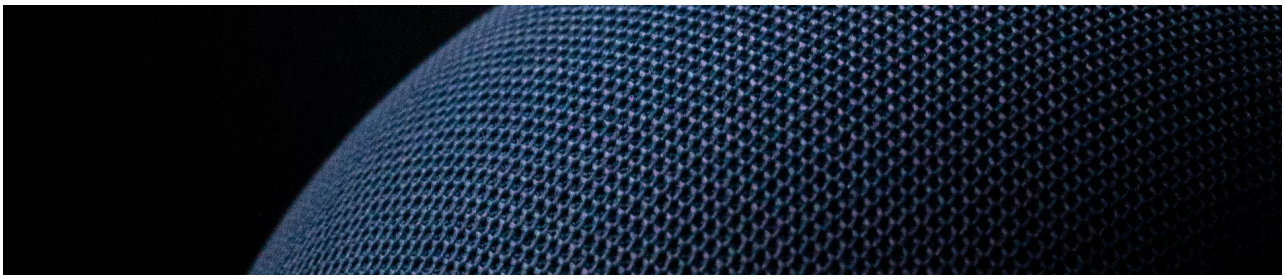
Understanding and Answering Incomplete Questions

Angus Addlesee
Heriot-Watt University
Edinburgh, United Kingdom
a.addlesee@hw.ac.uk

Marco Damonte
Amazon Alexa
Cambridge, United Kingdom
dammarco@amazon.co.uk



ABSTRACT

Voice assistants interrupt people when they pause mid-question, a frustrating interaction that requires the full repetition of the entire question again. This impacts all users, but particularly people with cognitive impairments. In human-human conversation, these situations are recovered naturally as people understand the words that *were* uttered. In this paper we build answer pipelines which parse incomplete questions and repair them following human recovery strategies. We evaluated these pipelines on our new corpus, SLUICE. It contains 21,000 interrupted questions, from LC-QuAD 2.0 and QALD-9-plus, paired with their underspecified SPARQL queries. Compared to a system that is given the full question, our best partial understanding pipeline answered only 0.77% fewer questions. Results show that our pipeline correctly identifies what information is required to provide an answer but is not yet provided by the incomplete question. It also accurately identifies where that missing information belongs in the semantic structure of the question.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**; *Accessibility technologies*; *Sound-based input / output*.

KEYWORDS

voice user experience, accessibility, semantic parsing, knowledge base question answering, human agent interaction

ACM Reference Format:
Angus Addlesee and Marco Damonte. 2023. Understanding and Answering Incomplete Questions. In *ACM conference on Conversational User Interfaces (CUI '23), July 19–21, 2023, Eindhoven, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3571884.3597133

CUI '23, July 19–21, 2023, Eindhoven, Netherlands

ACM ISBN 979-8-4007-0014-9/23/07.
https://doi.org/10.1145/3571884.3597133

User:	EVA, when is the next solar...
EVA:	Sorry, I'm not sure about that
User:	Eclipse
User:	EVA, eclipse
EVA:	[Error Sound]
User:	EVA, when is the next solar eclipse?

Table 1: An interruption caused by a pause.

1 INTRODUCTION

We all pause mid-sentence during our everyday conversations, actively trying to conjure the word we have forgotten. These pauses are so pronounced that in human-human interaction, mid-utterance pauses are longer on average than gaps between turns [17, 31, 81, 85]. When interacting with voice assistants however, this short silence can trigger end-of-turn detection early - interrupting and frustrating the user [42, 60, 63]. Consider the following interaction between a user and an Everyday Voice Assistant (EVA) in Table 1.

This impacts the experience of all users, but certain user groups are impacted more than others - often the people that can benefit the most from voice assistants [2]. For example, people with dementia pause more frequently mid-utterance as they try to remember the next word [15, 82]. With national dementia charities promoting the use of voice assistants [23, 35, 56, 69], industry promoting HIPAA-compliant healthcare skills with senior living providers [16, 43], and voice assistant features released specifically for people with dementia [26, 74], we argue this presents a valuable opportunity to work on voice accessibility. Not only will this assist people with cognitive impairments and their families, the solution extends to benefit all users.

Spoken language unfolds over time. Our interlocutors process each token as it is uttered, maintaining a partial representation of what has been said [44, 52, 55]. That is, we understand the words that *were already said* if someone pauses mid-sentence. To avoid waiting indefinitely while a conversation partner is pausing, humans either prompt the turn-holder to continue (often using a sluice [34, 37]) to collaboratively complete the question [70] as shown in Table 2, or suggest sentence completions themselves (referred to

User:	EVA, Is Alex Rodriguez dating...
EVA:	Sorry, I didn't catch that. Dating who?
User:	Jennifer Lopez
EVA:	Yes, they are currently dating.

Table 2: Collaborative completion from understanding.

User:	EVA, when is the next solar...
EVA:	The next solar eclipse is on the 20th April 2023

Table 3: Prediction of question completion.

as cross-person compound contributions or gap-fillers [40, 41, 71]) shown in Table 3. In this paper, we implement both approaches to answer people's interrupted questions. We therefore consider Knowledge Base Question Answering (KBQA), where a semantic parser is used to convert questions into an executable meaning representation over some given knowledge. For example, a system may be asked to answer "What is the population of Portugal?" when given Wikipedia as a knowledge base.

We hypothesise that although the example in Table 3 is more time-efficient when the completion is correct, the prediction will often be an arbitrary guess (completing "Who wrote..." for example) and risks further frustrating the user without correctly answering many interrupted questions. We also hypothesise that by adding just one additional turn in an interaction, we would be able to answer most interrupted questions. In order to determine whether these hypotheses are true, we need a corpus of incomplete questions and their underspecified meaning representations. The chosen meaning representation language (MRL) must therefore be able to handle: *incrementality*, allowing structurally incomplete meanings to be established over time; and *conjunction*, enabling the semantic representation of both the incomplete utterance and follow-up completion to be consolidated into the representation of the full question. As a suitable corpus does not exist, we must be able to align question text with its semantic structure to 'interrupt' questions in an existing KBQA corpus.

In this work we introduce the task of understanding and answering incomplete questions, and present a new corpus of 21,000 interrupted questions paired with their underspecified meaning representations. We introduce four interruption recovery pipelines based upon human repair strategies, and release the corpus to encourage further improvements over the proposed pipelines. We show that automatic completion of the question is not viable, while collaborative completion successfully repairs and answers most questions with only one additional turn. Currently a user has to repeat their entire question again, so this one additional turn avoids frustration by enabling a more naturally interactive user experience inspired by human-human conversation.

2 RELATED WORK

RDF [49, 53] is often used as a semantic parsing MRL [11, 86] to describe knowledge graphs with triple statements (for example: "Tuvalu", "part of", "Polynesia"). See Figure 1 for an illustration of two triples. There are huge RDF resources for linguistics [20, 57, 62],

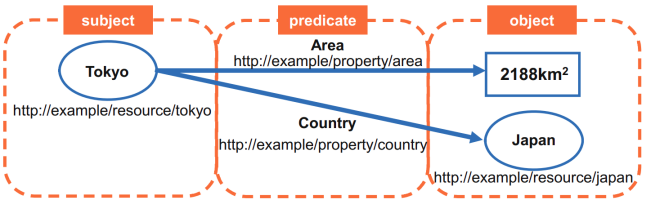


Figure 1: An example illustration of RDF [78].

more generally [12, 58, 61], and work already exists exploiting RDFs underspecification and conjunction properties for incremental semantic parsing [3]. Unfortunately, corpora that contain text/RDF pairs do not contain questions [4, 36, 86] and are therefore not fit for our domain. SPARQL [67, 68] is the standard RDF query language, similar to SQL, and is consequently more suited to represent questions. As SPARQL clauses directly contain RDF statements, our required underspecification and conjunction properties are preserved. Some target knowledge base (KB) is necessary however, so questions can not be represented if their constituents are not present in the KBs ontology. For example when asked "What is the CBI expansion rate of Kingstown?", there must be some RDF property to represent "CBI expansion rate" in the target KBs ontology. In order to measure how effective an interrupted recovery pipeline is, we must be able to determine whether the question is ultimately answered correctly. Using SPARQL over a target KB, we can easily return question's answers and even use the KBs ontology to inform our response generation (see Section 5.1).

Freebase [14] popularised SPARQL semantic parsing [13, 92] with many corpora targeting it [38, 46, 83]. Google eventually shut down Freebase [64, 79] to integrate its knowledge with Wikidata [89]. Both Wikidata and DBpedia [5] are the central open-domain KBs updated live today, and both are used to create KBQA corpora [6, 18, 30, 65]. DBpedia is updated automatically by live-extraction from Wikipedia [50, 59], whereas Wikidata is collaboratively edited by its community [90]. In fact, Wikipedia now incorporates content from Wikidata on almost every page in every language [32]. This can only be achieved by administering a cohesive and controlled ontology. In DBpedia however, India and Ireland are attached to their populations with different predicates [1]. This would negatively impact a model's performance as it would have to guess one of many predicates. We selected LC-QuAD 2.0 [30] and QALD-9-plus [65] with Wikidata as the target KB for this reason.

The second Largescale Complex Question Answering Dataset (LC-QuAD 2.0) contains 30,000 questions with their corresponding SPARQL queries targeting Wikidata for KBQA [66]. In addition to standard questions, LC-QuAD 2.0 incorporates multi-fact questions, temporal questions, questions that utilise qualifiers, and questions that contain dual user intents. Recent work [10] posed their T5-based model as the new state-of-the-art on LC-QuAD 2.0, although it actually addresses a different, much simpler task. They provide their model with each specific question's gold Wikidata entities and properties *as input*, transforming this complex task into a jigsaw puzzle. In other work [94], the authors correctly identify that LC-QuAD 2.0 contains various mistakes, but they then describe correcting the mistakes. They evaluated their model on their cleaned

test set which make their results incomparable. They did not release their model, code, or cleaned corpus to enable any comparison. Contrarily, the authors of a DeepPavlov model [33] released their work in 2020 with full documentation to re-implement their approach. Using the trained model provided by the authors, we evaluated it with the same local copy of Wikidata used in the paper, and with Wikidata's online API. We could not recreate their results[1]. In addition, their model exploits the templative generation of LC-QuAD 2.0 so much that the performance drops dramatically when asked natural questions. This includes the paraphrased questions that are present in the corpus itself. The leading sequence-to-sequence (seq2seq) model is EIneuQA [28].

3 GENERATING SLUICE - A CORPUS OF INTERRUPTED QUESTIONS

LC-QuAD 2.0 was created using a templative generation approach with 22 templates, populated with over 22,500 Wikidata entities and predicates. Each question in LC-QuAD 2.0 was also paraphrased using crowdsourcing, which we utilise below to increase the size of our corpus. This templative generation facilitates its large size, but does often generate questions that would not likely ever be asked by a human. For example: "What is the MIA constituent ID for Johannes Gutenberg?". This has the potential to bias our final evaluation results as our model to predict question completions, pre-trained on human language, will be tasked to complete a question never asked by a human. We therefore also selected a corpus of real human-asked questions that is not large enough to assist training, but can serve as an additional test set unbiased against the prediction pipelines. QALD-9-plus contains 500 questions manually created and translated by humans into eight other languages. We will only be using the English questions, paired with their corresponding Wikidata SPARQL queries.

All of the questions in both LC-QuAD 2.0 and QALD-9-plus are complete questions that can be answered directly. In order to investigate recovery strategies when a voice assistant interrupts a user's question, we must artificially 'chop' these complete questions. We considered splitting the questions at random, but found that mid-utterance pauses usually precede named entities due to word-finding problems [22, 80, 82]. Apple used this linguistic observation to improve their entity recognition on user data in English and French [27]. We therefore decided to use Named Entity Recognition (NER) to identify questions that end with named entities, 'chopping' the question where the user is most likely to pause. This location also ensures that a full semantic recovery is possible. Pauses before named entities earlier in the question would be un-recoverable, for example, "EVA, in".

Wikidata entities are linked to their human readable labels in various languages including English. If spaCy NER [39] identified a question ending with a named entity, we compared the NER tagged text with the English label of each Wikidata entity in the corresponding SPARQL query. When the NER tagged text and entity label matched[2], they were 'chopped' accordingly. In a similar process used to handle incomplete instructions in robotics [19], we took advantage of underspecification in SPARQL to indicate incompleteness with a variable (we used "?unknown"). With all of this in mind, we designed two chopping methods: (1) removing the NER tagged text from the question, and replacing the corresponding entity with a SPARQL variable; and (2) carrying out chopping method (1), and then in the case that the chopped entity is the *object* of the triple (where a triple follows the pattern: subject, predicate, object), replacing the predicate property with its superclass in Wikidata's ontology. This second method generalises the MRL when an entity is removed. For example, when asked "What is the highest mountain in Europe?", the answer is connected to "Europe" with the "continent" predicate. The interrupted question "What is the smallest mountain in" could be completed with "Egypt" however, which is not a continent. By replacing "continent" with its superclass "located in/on", we expand the scope of the MRL to allow more completions. We illustrate this difference in Table 4, and hereafter refer to **C**hopping **M**ethods (1) and (2) as **CM**-Simple and **CM**-Super respectively.

Method	Relevant SPARQL Triple	Property and Label
Original	{?ans1 wdt:P30 wd:Q46}	wdt:P30 - continent
CM-Simple	{?ans1 wdt:P30 ?unknown}	wdt:P30 - continent
CM-Super	{?ans1 wdt:P706 ?unknown}	wdt:P706 - located in

Table 4: Comparison of the two question chopping methods transforming "What is the smallest mountain in Europe?" into "What is the smallest mountain in". Note - wd:Q46 has the label "Europe".

As mentioned in Section 3, LC-QuAD 2.0 has been paraphrased - which we can use to double the number of questions with gold SPARQL queries. For example, the generated question "What was the population of Somalia in 2009?" was paraphrased to "As of 2009, how many people lived in Somalia?" and both have the exact same meaning representation. We can therefore chop this question twice, one underspecifying the time-constraint ("What was the population of Somalia in"), and the other underspecifying the location ("As of 2009, how many people lived in"). There were some additional queries that could be 'chopped' relatively easily, but that did not end with named entities. These were questions that ended with filter constraints. For example, "What German dog breed contains the word Weimaraner in its name?" and "What is the art form that begins with the letter s?". When questions fit this structure, we underspecified the filter in both the question and the SPARQL query. We repeated the above steps to interrupt human-asked questions found in QALD-9-plus.

With the above complete, we present SLUICE: SPARQL for Learning and Understanding Interrupted Customer Enquiries. SLUICE contains 21,000 artificially interrupted questions with their underspecified SPARQL queries. A more detailed breakdown can be found in Table 5[3].

[1]We contacted the authors about failing to reproduce their results. They said they are planning to address this in the future.

[2]If the strings had a similarity ratio above 0.7, using Levenshtein Distance, they were considered a match. We tweaked this similarity ratio by manually checking the quality of the additional questions generated with lower values.

[3]For reproducibility and future research, SLUICE and a guide on how to expand our approach to other corpora with different KBs and graph MRLs can be found here: https://github.com/AddleseeHQ/SLUICE

Source Corpus	Train	Test	Total
LC-QuAD 2.0	16,650	4,153	20,803
QALD-9-plus	NA	197	197
Total (SLUICE)	16,650	4,350	21,000

Table 5: SLUICE train and test set statistics.

4 BASELINES

We need to establish a suitable KBQA baseline to evaluate the performance of our interruption recovery strategies against. We expect this baseline to take a question as input, send a generated SPARQL query to Wikidata, and return the response.

It has been shown that enabling the use of pointer networks [88] to "copy" entity and relation mentions is crucial to achieve state-of-the-art KBQA performance [77]. To follow suit, we trained our model to output SPARQL queries containing pointers when given a text question. Inspired by an architecture designed for task-oriented semantic parsing [76], we trained an attentive seq2seq model [7] with a pretrained RoBERTa encoder [51], and transformer decoder [87]. Our model was trained with Adam [47] on a P3 AWS machine. The pointers output by our semantic parser must be resolved into their corresponding Wikidata IDs, requiring *entity linking*. We utilised features of the RDF triplestore in which Wikidata is contained for entity linking. Wikidata's query service runs on Blazegraph[4] which supports a full text indexing (FTI) and search facility powered by Apache Solr[5]. We used this to build an FTI across the entirety of Wikidata, enabling configurable matching on tokenized RDF literals (strings, numbers, and dates) with the '*bds*' vocabulary. When multiple entities match with the exact same score, we rank the results by sitelinks - the number of links on the entity's Wikipedia page. An example can be seen in Listing 1. Once the entity linker has fully resolved the SPARQL query, it can be used to query Wikidata for an answer. This system is illustrated in Figure 2.

```
SELECT ?qcode ?score ?num
WHERE {
?o bds:search "Paris" ;
   bds:minRelevance "0.7" ;
   bds:relevance ?score .

?qcode rdfs:label ?o ;
       wikibase:sitelinks ?num .

FILTER(langMatches(lang(?o), "EN"))
}
ORDER BY DESC(?score) DESC(?num)
```

Listing 1: SPARQL query using Blazegraph's FTI to search for entities labeled "Paris" in English with a minimum score of 0.7 - ranked by score and sitelinks. This returns the correct Wikidata entity identifier for the city Paris: Q90.

[4]https://github.com/blazegraph/database
[5]https://solr.apache.org/

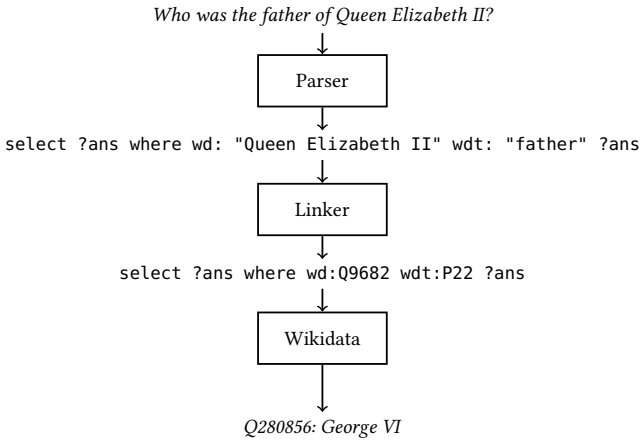


Figure 2: The full answer pipeline when asked "Who was the father of Queen Elizabeth II?".

In Table 6, we compare the open source DeepPavlov with our baseline on both the original LC-QuAD 2.0 corpus and the paraphrased version. Despite not being able to reproduce the results reported by Evseev and Arkhipov [33], the DeepPavlov model's exploitation of the corpus's template generation leads to a large performance drop when asked non-templative questions. SLUICE utilises these paraphrased questions, and questions from QALD-9-plus that are not generated from the 22 templates used in LC-QuAD 2.0, so DeepPavlov is an unsuitable baseline. We also report the results of the state-of-the-art sequence-to-sequence model by Diomedi and Hogan [28], which we outperform by a large margin. We therefore selected our model as the foundation of our partial understanding pipelines below.

5 ANSWERING INTERRUPTED QUESTIONS

As discussed in Section 1, people recover from mid-utterance pauses in dialogue by either: (1) using their partial understanding to initiate collaborative completion; or (2) predicting the end of their interlocutor's sentence. We have implemented both human-inspired approaches for evaluation.

5.1 Collaborative Completion Pipeline

To illustrate our desired interaction with the user, a simple example from SLUICE is shown in Table 7.

We started building a pipeline to support this interaction by retraining our top-performing baseline on SLUICE, expecting the model to output a SPARQL query that not only identifies the variable that represents the answer to the user's question, but also the variable that represents what knowledge is underspecified and still required to answer the question. The pointers were then resolved into their Wikidata identifiers using the FTI linker shown in Listing 1. This resolved SPARQL query will not return the correct answer, due to the "?unknown" variable, so we elicit a follow-up response from the user.

In order to generate the "who?" response in Table 7, we use the underspecified SPARQL query and the ontology of Wikidata itself. As described in Section 2, we targeted Wikidata over DBpedia

Model	LC-QuAD 2.0 Answer %	Paraphrase Answer %	Macro F1
DeepPavlov	**56.30*** (34.11)	(22.87)	-
Our Baseline	50.33	**42.25**	**32.99**
EIneuQA	-	-	26.90

Table 6: Comparison of potential baseline models on the LC-QuAD 2.0 corpus, and the paraphrased LC-QuAD 2.0 questions. *We tried but could not recreate this result, our result is the number in brackets.

User:	EVA, who was the father of...
EVA:	Sorry, I didn't catch all of that. Of who?
User:	Queen Elizabeth II
EVA:	George VI was the father of Queen Elizabeth II

Table 7: An ideal interaction with a user.

because of its more cohesive and controlled ontology. You cannot claim that the 'father' of a human is a caravan, for example, as Wikidata's ontology will not allow it. There are class constraints on what entity types the predicate 'father' can link to, and we can therefore ask Wikidata what class the "?unknown" variable must be. In Figure 2 we illustrate the resolved SPARQL query representing the question "Who was the father of Queen Elizabeth II?". Our model trained on SLUICE outputs the underspecified query when asked "Who was the father of": *SELECT ?ans WHERE {?unknown wdt:P22 ?ans .}*. You can see the "?unknown" variable is the subject of the predicate "P22" - *father*. We can ask Wikidata what class this variable is with the SPARQL query shown in Listing 2.

```
SELECT ?constraint WHERE {
  wd:P22 p:P2302 _:b1.
  _:b1 ps:P2302 wd:Q21503250;
    pq:P2308 ?constraint.
}
```

Listing 2: SPARQL query returning the type constraints for Wikidata property P22 - father. This returns Q5 - human (amongst others e.g. "fictional character"). The term "wd:Q21503250" can be replaced by "wd:Q21510865" when the "?unknown" variable is the object of the predicate and Wikidata's value-type constraints are required. Some variables in this query have no human-readable label (e.g. b1) as we are querying constraints in Wikidata's structure.

In order to respond more intelligently, we must 'learn' which constraints generate which responses. For example, responding: (1) "painted **what**?" When asked "Who painted", (2) "in **where**?" When asked "What is the smallest mountain in", and (3) "dating **who**?" When asked "Is Alex Rodriguez dating". We made use of SLUICEs metadata to do this, as we stored the NER tag that the chopped entity was assigned during the corpus's generation. There are only 12 of these tags, so we manually labelled each of these tags with the appropriate response if it was missing. To continue the above examples, we labeled (1) 'WORK OF ART' with 'what', (2) 'GPE' with 'where', and (3) 'PERSON' with 'who' - as these are the appropriate responses when that type of entity is missing. We then ran the queries in Listing 2 to return Wikidata's class

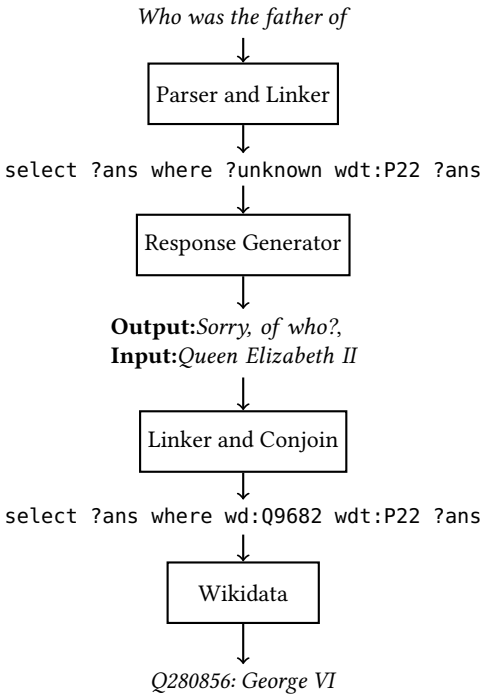


Figure 3: The interrupted question recovery pipeline when asked "Who was the father of". The response generation elicits the question completion from the user, which is conjoined with the parser's underspecified SPARQL query with the Blazegraph FTI linker.

constraints and matched them with the label that the NER tag was assigned. Due to NER errors, some classes were labeled with multiple appropriate responses (for example, Q5 - 'human' was attached to 'where' a few times). To tackle this, we only assigned constraints to appropriate responses if they matched at least five times. We additionally matched a class's superclass to the same response in order to guide our responses even if an unseen class constraint was returned. For example, 'human' is a subclass of 'person' so our pipeline will still respond 'who' if a new subclass of 'person' is returned in future. This process lets us create a dictionary of class constraints and their corresponding responses, which have an agreement rate of 70% with the NER tag responses (100% is not possible due to NER errors).

With the response generation above, and the Blazegraph FTI entity linker described in Section 4, we can now handle interrupted questions as desired. In Figure 3 we depict this pipeline for clarity.

5.2 Predicting Question Completions

Considering the example "When is the next solar", it is clear that predicting the completion of a question could improve a voice assistants's user experience. We therefore evaluated two T5 language models [72] against our partial understanding pipelines as a state-of-the-art comparison [21, 45, 54]. It is possible to fine-tune T5 with domain-specific examples, and with new task *contexts*. For example, you can send text to T5 with the "translate English to German" context, "summarize" context, or "answer" context when the text is a question. We found a T5-base model fine-tuned on a question answering corpus [75] that could be easily utilised through Hugging Face [91]. This model was fine-tuned on SQuAD v1.1 [73], a machine comprehension corpus containing over 100,000 question/answer pairs posed by crowdworkers on Wikipedia articles. We passed every question in SLUICE's test set through this model for completion prediction with the "question" context. Additionally, Using SLUICE's training set and a new task context "complete the question", we fine-tuned our own T5-base model specifically tailored to completing interrupted questions. Five examples were randomly selected, and you can see this model's predictions in Table 8. The T5 model fine-tuned on SQuAD v1.1 predicted that examples 2 and 5 (in Table 8) were already complete, predicted "the book", and "the girl" for examples 3 and 4 respectively, and rewrote example 1 - providing no additional information. It is clear that our T5 model fine-tuned on SLUICE generates realistic context-aware completions (e.g. predicts a comic in example 5). It does appear that although the predictions make sense, they are still just guesses and are therefore incorrect.

6 FINAL EVALUATION

When a user's question is interrupted by a voice assistant, their ultimate goal is to have their question answered. We therefore consider the % of questions answered correctly as the central metric to decide which approach maximises benefit to the user. We evaluated four recovery strategies, and compared their performance to the top-performing baseline when it is given the original full question. In Table 9 we present results for the following answer pipelines:

- The top-performing baseline (given full questions): To determine the effectiveness of our interruption recovery strategies, our baseline model was evaluated using the original, un-interrupted, questions in SLUICE.
- Partial CM-Simple: The pipeline in Figure 3 trained on the CM-Simple version of SLUICE.
- Partial CM-Super: The pipeline in Figure 3 trained on the CM-Super version of SLUICE.
- T5 SQuAD: The top-performing baseline answering questions completed by T5 fine-tuned on SQuAd v1.1.
- T5 SLUICE: The top-performing baseline answering questions completed by T5 fine-tuned on SLUICE.

The two T5 prediction models perform poorly compared to the partial understanding approaches, with the model fine-tuned on SLUICE outperforming the SQuAD v1.1 model. From the sample of predictions in Table 8, and a wider manual inspection, we expect this poor performance is caused by arbitrary guesses (e.g. completing "Who wrote"). As discussed in Section 3, the T5 prediction models, pretrained on human language, could be punished when asked questions originating from LC-QuAD as they were not asked by humans. If the T5 models outperformed the partial understanding models on the 197 human-asked questions, we could determine that the results were indeed biased. However, this is not the case. The T5 models also perform poorly in the "Answer:Human" column in Table 9.

The partial CM-Super pipeline is outperformed by the partial CM-Simple pipeline across all metrics. Although the CM-Super chopping method seems logical (see Table 4), this is not often the case to a machine. The pointer mechanism in particular identifies tokens in the input question to be resolved by the linker. By replacing properties with their superclass, pointers are no longer resolved to the correct property in the query, and we expect this is the reason for the reduced performance.

The partial CM-Simple pipeline is the best of our interruption recovery pipelines - answering only 0.77% fewer questions correctly than the baseline given complete questions. To emphasise this remarkable result, the parser within the partial CM-Simple pipeline must generate a valid SPARQL query identifying not only the answer variable, but the "?unknown" variable representing what the model does *not yet know*. The correct answer is only provided if the parser accurately identifies where this unknown variable is located within the query structure, attaches it to the right property, and the linker returns the exact Wikidata ID. In contrast, the baseline is given all information as input.

The SPARQL Exact Match score is a strict metric, with a literal interpretation of *exact*. If the triple is reversed (e.g. 'x dates y' instead of 'y dates x') or even if all the triples are correct in the query, just in the wrong order - the queries are not considered a match. This explains how the T5 SQuAD model answered 11.3% of questions correctly with zero exact query matches, and how the partial CM-Simple model achieved the highest SPARQL exact match score without the highest answer score.

7 CONCLUSION AND FUTURE WORK

In this paper we presented four interruption recovery pipelines, based on human recovery strategies, and evaluated these against a SotA-level baseline given fully completed questions. These incomplete questions (e.g. "Who wrote") can not currently be answered by today's voice assistants without full repetition of the entire question. This is not a natural interaction, frustrates users, and severely impacts the accessibility of voice assistants for people with cognitive impairments. We found that predicting question completions would likely frustrate the user further, often resorting to arbitrary guesses. In contrast, we found that understanding what *was said* by the user and collaboratively completing the question did benefit the user. This pipeline only answered 0.77% fewer questions than a state-of-the-art baseline given the full question as input.

No suitable dataset existed to train or evaluate models for this task. We therefore created SLUICE - a corpus of 21,000 interrupted questions and their underspecified SPARQL queries. We experimented with two generation methods to create SLUICE, and will release the CM-Simple version as it outperformed CM-Super across all metrics. Alongside the corpus, we will release a guide on how to extend our question interruption approach to other corpora. In this paper we have shown that interruption recovery is possible

Ex	Interrupted Question	Original	T5 SLUICE
1	Franz Waxman won what award at the 23rd	Academy Awards	Academy Awards
2	In what area does the Rideau Canal join	the Ottawa River	the Ottawa River
3	Who wrote	Harry Potter	The Great Gatsby
4	Who was the father of	Queen Elizabeth II	Sigmund Freud
5	Who created the comic	Captain America	X-Men

Table 8: Comparison of the original question completions, and our T5 model fine-tuned on SLUICE.

Answer Pipeline	SPARQL Exact Match	Answer	Answer Delta	Answer:Human
Top-performing Baseline (given full questions)	28.59	**46.40**	-	**30.96**
Partial CM-Simple	**33.65**	45.63	**0.77**	25.38
Partial CM-Super	29.86	42.01	4.39	23.35
T5 SQuAD	0.00	11.30	35.10	15.74
T5 SLUICE	1.08	15.45	30.95	18.27

Table 9: Final evaluation results on the SLUICE test set. SPARQL Exact Match: % of SPARQL queries that *exactly* match the reference, Answer: % of questions answered correctly, Answer Delta: the difference between the % of questions answered by the top-performing baseline (given the full question) and the recovery pipeline, and Answer:Human: the Answer metric but only on the *human-asked* questions (originally from QALD-9-plus) in the SLUICE test set.

and worthwhile with remarkable results, so this guide will allow future research to generate even larger interrupted corpora. People commonly pause before named entities in other languages, like French [27], so our approach could also be applied to generate a multilingual corpus.

In order to apply our findings in practice, we would have to reliably identify incomplete questions from the user. A classifier could be trained to do this, or our recovery system could be invoked as a fallback when the question cannot be answered with a high confidence. Accessibility settings, in Siri for example [84], allow users to modify how long a voice assistant waits until it decides that a sentence is complete. This is a wonderful temporary solution for people with more progressed cognitive impairment, but it is not naturally interactive. As mentioned, we all forget words mid-sentence, but this is more common as conditions like dementia or mild cognitive impairment (MCI) progress [2]. People with dementia or MCI do not pause in the middle of all sentences, however, so waiting for long durations between *every* turn would be frustrating. Our more sophisticated and human-inspired approach improves system accessibility without sacrificing system naturalness or fluidity.

SPARQL is a KBQA-specific MRL requiring relations in a question to be represented by the target KBs ontology. In order to parse interrupted utterances more generally, we do not need to use a representation that is directly executable for an answer. Abstract Meaning Representation (AMR) [8, 9] can represent *all* sentences as labeled graphs, taking *all* words into account in a reasonably consistent manner [9]. Previous work on incremental AMR parsing has exploited its underspecification and conjunction properties [25], and AMR corpora do contain natural language questions [24, 48]. We plan to use recently released alignments between text/AMR pairs [29, 93] to interrupt sentence/AMR pairs and explore what types of semantic information are particularly easy or difficult to recover using a similar approach.

ACKNOWLEDGMENTS

This project was completed during an Amazon Alexa internship in Cambridge. We would like to thank our colleagues for feedback and advice during the completion of this work. This work was also partially funded by the EU H2020 program under grant agreement no. 871245 (https://spring-h2020.eu/)

We would also like to thank our anonymous reviewers for their time and valuable feedback.

REFERENCES

[1] Angus Addlesee. 2019. Constructing More Advanced SPARQL Queries. https://medium.com/wallscope/constructing-more-advanced-sparql-queries-72d5ade1eedc [Online; accessed 19-May-2022].

[2] Angus Addlesee. 2023. Voice Assistant Accessibility. In *Proceedings of The 13th International Workshop on Spoken Dialogue Systems (IWSDS) 2023.*

[3] Angus Addlesee and Arash Eshghi. 2021. Incremental Graph-Based Semantics and Reasoning for Conversational AI. In *Proceedings of the Reasoning and Interaction Conference (ReInAct 2021).* 1–7.

[4] Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv preprint arXiv:2010.12688* (2020).

[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.

[6] Michael Azmy, Peng Shi, Jimmy Lin, and Ihab Ilyas. 2018. Farewell Freebase: Migrating the SimpleQuestions Dataset to DBpedia. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2093–2103. https://aclanthology.org/C18-1178

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[8] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2012. Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL.* 1533–1544.

[9] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*. 178–186.

[10] Debayan Banerjee, Pranav Ajit Nair, Jivat Neet Kaur, Ricardo Usbeck, and Chris Biemann. 2022. Modern Baselines for SPARQL Semantic Parsing. *arXiv preprint arXiv:2204.12793* (2022).

[11] Brahim Batouche, Claire Gardent, Anne Monceaux, and France Blagnac. 2014. Parsing text into RDF graphs. In *Proceedings of the XXXI Congress of the Spanish Society for the Processing of Natural Language.*

[12] Bradley R Bebee, Daniel Choi, Ankit Gupta, Andi Gutmans, Ankesh Khandelwal, Yigit Kiran, Sainath Mallidi, Bruce McGaughy, Mike Personick, Karthik Rajan, et al. 2018. Amazon Neptune: Graph Data Management in the Cloud.. In *International Semantic Web Conference (P&D/Industry/BlueSky).*

[13] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing.* 1533–1544.

[14] 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.* 1247–1250.

[15] Veronica Boschi, Eleonora Catricala, Monica Consonni, Cristiano Chesi, Andrea Moro, and Stefano F Cappa. 2017. Connected speech in neurodegenerative language disorders: a review. *Frontiers in psychology* 8 (2017), 269.

[16] Lois Bowers. 2019. Amazon announces HIPAA-compliant skills for Alexa, with senior living parent companies in the mix. https://www.mcknightsseniorliving.com/home/news/amazon-announces-hipaa-compliant-skills-for-alexa-with-senior-living-parent-companies-in-the-mix/ McKnights Senior Living. [Online; accessed 17-May-2022].

[17] Paul T Brady. 1968. A statistical analysis of on-off patterns in 16 conversations. *Bell System Technical Journal* 47, 1 (1968), 73–91.

[18] Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. KQA Pro: A Dataset with Explicit Compositional Programs for Complex Question Answering over Knowledge Base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 6101–6119.

[19] Haonan Chen, Hao Tan, Alan Kuntz, Mohit Bansal, and Ron Alterovitz. 2020. Enabling robots to understand incomplete natural language instructions using commonsense reasoning. In *2020 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 1963–1969.

[20] Christian Chiarcos, Thierry Declerck, and John P Mccrae. 2013. Linguistic Linked Open Data (LLOD) Introduction and Overview. (2013).

[21] Jordan Clive, Kris Cao, and Marek Rei. 2021. Control Prefixes for Text Generation. *arXiv preprint arXiv:2110.08329* (2021).

[22] Bernard Croisile, Bernadette Ska, Marie-Josee Brabant, Annick Duchene, Yves Lepage, Gilbert Aimard, and Marc Trillet. 1996. Comparative study of oral and written picture description in patients with Alzheimer's disease. *Brain and language* 53, 1 (1996), 1–19.

[23] DailyCaring. 2020. Amazon Echo Alexa Helps Seniors with Dementia. https://dailycaring.com/amazon-echo-for-dementia-technology-for-seniors/ [Online; accessed 17-May-2022].

[24] Marco Damonte and Shay Cohen. 2022. Abstract Meaning Representation 2.0-Four Translations. (2022).

[25] Marco Damonte, Shay B Cohen, and Giorgio Satta. 2016. An incremental parser for abstract meaning representation. *arXiv preprint arXiv:1608.06111* (2016).

[26] DBSC. 2020. M4D Radio. https://www.amazon.co.uk/DBSC-LTD-M4D-Radio/dp/B089GQBY5R [Online; accessed 17-May-2022].

[27] Sahas Dendukuri, Pooja Chitkara, Joel Ruben Antony Moniz, Xiao Yang, Manos Tsagkias, and Stephen Pulman. 2021. Using Pause Information for More Accurate Entity Recognition. *arXiv preprint arXiv:2109.13222* (2021).

[28] Daniel Diomedi and Aidan Hogan. 2021. Question Answering over Knowledge Graphs with Neural Machine Translation and Entity Linking. *arXiv preprint arXiv:2107.02865* (2021).

[29] Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramon Fernandez Astudillo. 2022. Inducing and Using Alignments for Transition-based AMR Parsing. *arXiv preprint arXiv:2205.01464* (2022).

[30] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *International semantic web conference.* Springer, 69–78.

[31] Jens Edlund and Mattias Heldner. 2005. Exploring prosody in interaction control. *Phonetica* 62, 2-4 (2005), 215–226.

[32] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing Wikidata to the linked data web. In *International semantic web conference.* Springer, 50–65.

[33] DA Evseev and M Yu Arkhipov. 2020. Sparql query generation for complex question answering with bert and bilstm-based model. In *Computational Linguistics and Intellectual Technologies.* 270–282.

[34] Raquel Fernández, Jonathan Ginzburg, and Shalom Lappin. 2007. Classifying non-sentential utterances in dialogue: A machine learning approach. *Computational Linguistics* 33, 3 (2007), 397–427.

[35] Gillian Fyfe. 2019. Amazon Echo. https://www.alzscot.org/amazon-echo Alzheimer Scotland. [Online; accessed 17-May-2022].

[36] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation.* 124–133.

[37] Jonathan Ginzburg and Ivan Sag. 2000. *Interrogative investigations.* Stanford: CSLI publications.

[38] 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.* 3477–3488.

[39] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python, 2020. *URL https://doi. org/10.5281/zenodo* 1212303, 6 (2020).

[40] Christine Howes, Ptarick GT Healey, Matthew Purver, and Arash Eshghi. 2012. Finishing each other's... responding to incomplete contributions in dialogue. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 34.

[41] Christine Howes, Matthew Purver, Patrick GT Healey, Gregory J Mills, and Eleni Gregoromichelaki. 2011. On incrementality in dialogue: Evidence from compound contributions. *Dialogue & Discourse* 2, 1 (2011), 279–311.

[42] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How do users respond to voice input errors? Lexical and phonetic query reformulation in voice search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval.* 143–152.

[43] Rachel Jiang. 2019. Introducing New Alexa Healthcare Skills. https://developer.amazon.com/blogs/alexa/post/ff33dbc7-6cf5-4db8-b203-99144a251a21/introducing-new-alexa-healthcare-skills Alexa Blogs. [Online; accessed 17-May-2022].

[44] Patrick Kahardipraja, Brielen Madureira, and David Schlangen. 2021. Towards Incremental Transformers: An Empirical Analysis of Transformer Models for Incremental NLU. *arXiv preprint arXiv:2109.07364* (2021).

[45] Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433* (2020).

[46] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713* (2019).

[47] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[48] Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2021. Abstract Meaning Representation (AMR) Annotation Release 3.0. https://doi.org/11272.1/AB2/82CVJF

[49] Ora Lassila, Ralph R Swick, et al. 1998. Resource description framework (RDF) model and syntax specification. *W3C recommendation* (1998).

[50] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6, 2 (2015), 167–195.

[51] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[52] Brielen Madureira and David Schlangen. 2020. Incremental processing in the age of non-incremental encoders: An empirical assessment of bidirectional models for incremental NLU. *arXiv preprint arXiv:2010.05330* (2020).

[53] Frank Manola, Eric Miller, Brian McBride, et al. 2004. RDF primer. *W3C recommendation* 10, 1-107 (2004), 6.

[54] Vinsen Marselino Andreas, Genta Indra Winata, and Ayu Purwarianti. 2022. A Comparative Study on Language Models for Task-Oriented Dialogue Systems. *arXiv e-prints* (2022), arXiv–2201.

[55] William Marslen-Wilson. 1973. Linguistic structure and speech shadowing at very short latencies. *Nature* 244, 5417 (1973), 522–523.

[56] Douglas McClusky. 2021. The Alexa Fund. https://www.battleagainstdementia.org/#:~:text=The%20'Alexa%20Fund'%20has,living%20in%20their%20own%20homes. [Online; accessed 17-May-2022].

[57] John McCrae, Dennis Spohr, and Philipp Cimiano. 2011. Linking lexical resources and ontologies on the semantic web with lemon. In *Extended Semantic Web Conference.* Springer, 245–259.

[58] Peter Mika. 2015. On schema. org and why it matters for the web. *IEEE Internet Computing* 19, 4 (2015), 52–55.

[59] Mohamed Morsey, Jens Lehmann, Sören Auer, Claus Stadler, and Sebastian Hellmann. 2012. Dbpedia and the live extraction of structured data from wikipedia. *Program* (2012).

[60] Mikio Nakano, Yuka Nagano, Kotaro Funakoshi, Toshihiko Ito, Kenji Araki, Yuji Hasegawa, and Hiroshi Tsujino. 2007. Analysis of user reactions to turn-taking failures in spoken dialogue systems. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue.* 120–123.

[61] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. 2015. RDFox: A highly-scalable RDF store. In *International Semantic Web*

Conference. Springer, 3–20.
[62] Finn Nielsen. 2020. Lexemes in Wikidata: 2020 status. In *Proceedings of the 7th Workshop on Linked Data in Linguistics (LDL-2020)*. 82–86.
[63] Laura Panfili, Steve Duman, Andrew Nave, Katherine Phelps Ridgeway, Nathan Eversole, and Ruhi Sarikaya. 2021. Human-AI interactions through a Gricean lens. *Proceedings of the Linguistic Society of America* 6, 1 (2021), 288–302.
[64] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*. 1419–1428.
[65] Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 2022. QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. IEEE. arXiv:2202.00120 [cs.CL]
[66] Aleksandr Perevalov, Xi Yan, Liubov Kovriguina, Longquan Jiang, Andreas Both, and Ricardo Usbeck. 2022. Knowledge Graph Question Answering Leaderboard: A Community Resource to Prevent a Replication Crisis. *arXiv preprint arXiv:2201.08174* (2022).
[67] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. 2006. Semantics and Complexity of SPARQL. In *International semantic web conference*. Springer, 30–43.
[68] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. 2009. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)* 34, 3 (2009), 1–45.
[69] PlaylistForLife. 2021. Testing Voice Activated Technology for Dementia. https://www.playlistforlife.org.uk/wp-content/uploads/2022/04/Testing-voice-activated-technology-for-dementia-report.pdf [Online; accessed 17-May-2022].
[70] Massimo Poesio and Hannes Rieses. 2010. Completions, coordination, and alignment in dialogue. *Dialogue & Discourse* 1, 1 (2010).
[71] Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2003. On the means for clarification in dialogue. In *Current and new directions in discourse and dialogue*. Springer, 235–255.
[72] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).
[73] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).
[74] RiseIQ. 2018. Dementia Skill. https://www.amazon.co.uk/Rise-iq-Dementia-Skill/dp/B07DBDT196 [Online; accessed 17-May-2022].
[75] Manuel Romero. 2021. T5 (base) fine-tuned on SQUAD for QG via AP. https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap.
[76] Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*. 2962–2968.
[77] Rishiraj Saha Roy and Avishek Anand. 2022. Complex Question Answering. In *Question Answering for the Curated Web*. Springer, 37–51.
[78] Seiji Sakakibara, Sachio Saiki, Masahide Nakamura, and Kiyoshi Yasuda. 2017. Generating personalized dialogue towards daily counseling system for home dementia care. In *Digital Human Modeling. Applications in Health, Safety, Ergonomics, and Risk Management: Health and Safety: 8th International Conference, DHM 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II 8*. Springer, 161–172.
[79] Barry Schwartz. 2014. Google To Close Freebase, Which Helped Feed Its Knowledge Graph. https://searchengineland.com/google-close-freebase-helped-feed-knowledge-graph-211103 [Online; accessed 19-May-2022].
[80] Frank Seifart, Jan Strunk, Swintha Danielsen, Iren Hartmann, Brigitte Pakendorf, Søren Wichmann, Alena Witzlack-Makarevich, Nivja H de Jong, and Balthasar Bickel. 2018. Nouns slow down speech across structurally and culturally diverse languages. *Proceedings of the National Academy of Sciences* 115, 22 (2018), 5720–5725.
[81] Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language* 67 (2021), 101178.
[82] Antoine Slegers, Renee-Pier Filiou, Maxime Montembeault, and Simona Maria Brambati. 2018. Connected speech features from picture description in Alzheimer's disease: A systematic review. *Journal of Alzheimer's Disease* 65, 2 (2018), 519–542.
[83] Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for qa evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 562–572.
[84] Apple Support. 2022. Use accessibility features with Siri on iPhone. https://support.apple.com/en-gb/guide/iphone/iphaff1d606/ios [Online; accessed 14-April-2023].
[85] Louis Ten Bosch, Nelleke Oostdijk, and Lou Boves. 2005. On temporal aspects of turn taking in conversational dialogues. *Speech Communication* 47, 1-2 (2005), 80–86.
[86] Trung Tran and Dang Tuan Nguyen. 2020. Webnlg 2020 challenge: Semantic template mining for generating references from rdf. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. 177–185.
[87] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[88] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015).
[89] Denny Vrandečić. 2012. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st international conference on world wide web*. 1063–1064.
[90] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
[91] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).
[92] Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 956–966.
[93] Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. 2021. Structure-aware Fine-tuning of Sequence-to-sequence Transformers for Transition-based AMR Parsing. *arXiv preprint arXiv:2110.15534* (2021).
[94] Jianyun Zou, Min Yang, Lichao Zhang, Yechen Xu, Qifan Pan, Fengqing Jiang, Ran Qin, Shushu Wang, Yifan He, Songfang Huang, et al. 2021. A chinese multi-type complex questions answering dataset over wikidata. *arXiv preprint arXiv:2111.06086* (2021).

Received 20 February 2023; revised 11 April 2023