Semantic Parsing for Conversational Question Answering over Knowledge Graphs

Laura Perez-Beltrachini[1], **Parag Jain**[1], **Emilio Monti**[2], **Mirella Lapata**[1]
[1] School of Informatics, University of Edinburgh
[2] Amazon Alexa
{lperez,parag.jain,mlap}@inf.ed.ac.uk, monti@amazon.co.uk

Abstract

In this paper, we are interested in developing semantic parsers which understand natural language questions embedded in a *conversation* with a user and *ground* them to formal queries over definitions in a general purpose knowledge graph (KG) with very large vocabularies (covering thousands of concept names and relations, and millions of entities). To this end, we develop a dataset where user questions are annotated with SPARQL parses and system answers correspond to execution results thereof. We present two different semantic parsing approaches and highlight the challenges of the task: dealing with large vocabularies, modelling conversation context, predicting queries with multiple entities, and generalising to new questions at test time. We hope our dataset will serve as useful testbed for the development of conversational semantic parsers.[1]

1 Introduction

Conversational information seeking is the process of acquiring information through conversations (Zamani et al., 2022). Recent years have seen an increasing number of applications aiming to build conversational interfaces based on information retrieval (Radlinski and Craswell, 2017) and user recommendation (Jannach et al., 2021). The popularity of intelligent voice assistants such as Amazon's Alexa or Apple's Siri has further stimulated research on question answering over general purpose knowledge graphs (e.g., Wikidata). Key to question answering in this context is the ability to ground natural language onto concepts, entities, and relations in order to produce an *executable* query (e.g., SPARQL) which will retrieve an answer or *denotation* from the knowledge graph (KG).

This grounding process, known as *semantic parsing* has been studied in the context of one or few domain-specific databases (Yu et al., 2019a; Jain and Lapata, 2021; Suhr et al., 2018) or without taking the conversational nature of the task into account (Reddy et al., 2014; Yih et al., 2016; Dubey et al., 2019; Gu et al., 2021). However, due to the complexities of the semantic parsing task, there are no large scale datasets consisting of information seeking conversations with executable queries against a KG. Conversational semantic parsing over KGs requires handling very large vocabularies covering thousands of concept names and relations, and millions of entities rather than specialized terms consisting of hundreds of tables and column names. Moreover, information seeking conversations are by nature incremental involving interrelated rather than isolated questions.

In this work, we create SPICE, a **S**emantic **P**ars**I**ng dataset for **C**onversational qu**E**stion answering over Wikidata. SPICE consists of user-assistant interactions where natural language questions are paired with SPARQL parses and answers provided by the system correspond to SPARQL execution results. We derive this dataset from CSQA (Saha et al., 2018), an existing benchmark originally proposed for retrieval-based conversational question answering (Lan et al., 2021). Although CSQA does not have executable queries, it contains a large number of natural language questions and their corresponding answers, highlighting a range of conversational phenomena such as coreference, ellipsis, and topic change as well as different types of questions exemplifying varying intents.

Table 1 shows a conversation from SPICE illustrating how questions (utterances on the left) are annotated with SPARQL queries (SP on the right blue box). To create a large-scale dataset (197k conversations), we develop SPARQL templates for different question intents; entity, relation, and class symbols are initially under-specified and subsequently filled automatically to generate full SPARQL queries. CSQA questions have been previously associated with logical forms generated with

[1] Our dataset and models are released at SPICE.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2507–2522
May 2-6, 2023

	Utterances	Annotations	Actions and Semantic Parses
$\mathcal{T}_1$	U: *Which tournament did Detroit Tigers participate in?* S: *1909 World Series*	INTENT=Simple Question|Single Entity ENT=[Q650855 (*Detroit Tigers*)] , REL=[P1923 (*participating team*)], TYP=[Q500834 (*tournament*)], TRIPLE=[(Q500834,P1923,Q650855)], GOLD =[Q846847 (*1909 World Series*)]	AS: [filter_type, find_rev, Q650855,P1923,Q500834] SP: SELECT ?x WHERE { ?x wdt:P1923 wd:Q650855 . ?x wdt:P31 wd:Q500834 . }
$\mathcal{T}_2$	U: *Which sports team was the champion of that tournament?* S: *Pittsburgh Pirates*	INTENT=Simple Question|Single Entity|Indirect ENT=[Q846847 (*1909 World Series*)] , REL=[P1346 (*winner*)], TYP=[Q12973014 (*sports team*)], TRIPLE=[(Q846847,P1346,Q12973014)], GOLD=[Q7199360 (*Pittsburgh Pirates*)]	AS: [filter_type, find, Q846847, P1346, Q12973014] SP: SELECT ?x WHERE { wd:Q846847 wdt:P1346 ?x . ?x wdt:P31 wd:Q12973014 . }
$\mathcal{T}_3$	U: *Does that sports team belong to Sacile?* S: *No*	INTENT=Verification|2 entities, subject is indirect ENT=[Q653772 (*Pittsburgh Pirates*), Q53190 (*Sacile*)], REL=[P17 (*country*)], TYP=[Q15617994 (*designation admin. territorial entity*)], TRIPLE=[(Q653772,P17,Q53190)], GOLD=[False]	AS: [is_in, Q53190, find, Q653772, P17] SP: ASK {wd:Q653772 wdt:P17 wd:Q53190 . }

Table 1: Example conversations from $\mathbb{SPICE}$. The left column shows dialogue turns ($\mathcal{T}_1$–$\mathcal{T}_3$) with user (U) and system (S) utterances. The middle column shows the annotations provided in CSQA. Blue boxes on the right show the sequence of actions (AS) and corresponding SPARQL semantic parses (SP).

custom-made grammars (Guo et al., 2018; Kacupaj et al., 2021; Marion et al., 2021). As a result, semantic parsers based on them operate with different sets of grammar rules and are not strictly comparable, since the grammars may have different coverage and semantics (e.g., terminal symbols may encapsulate different degrees of execution complexity). In $\mathbb{SPICE}$, questions are represented with SPARQL, a standard query language for retrieving and manipulating RDF data.[2] This allows us to compare parsers developed on the dataset on an equal footing and facilitates further extensions (e.g., new question intents), without the need to redefine the grammar and its execution engine. In an attempt to build semantic parsers which generalise to new user questions, we further create different data splits where new intents appear only at test time (Finegan-Dollak et al., 2018).

For our semantic parsing task, we establish two strong baseline models which tackle the large vocabulary problem and the prediction of logical forms in different ways. The first approach (Gu et al., 2021) uses *dynamic vocabularies* derived from KG subgraphs for each question and a simple sequence-to-sequence architecture to predict *complete* SPARQL queries. The other approach (Kacupaj et al., 2021) predicts SPARQL *query templates* and then fills in entity, relation, and type slots by means of an entity and ontology classifier. Our experiments reveal several shortcomings in both approaches, such as not being able to encode large sets of KG elements and generate the same entity several times. Both approaches struggle with ellipsis, they cannot resolve coreference when the referent appears in the conversation context beyond the previous turn, have reduced performance on questions with multiple entities, and struggle with unseen question intents. We discuss these challenges and outline research directions for conversational semantic parsing.

2 The $\mathbb{SPICE}$ Dataset

The CSQA dataset (Saha et al., 2018) aims to facilitate the development of QA systems that handle complex and inter-related questions over a knowledge graph. In contrast to simple factual questions that can be answered with a single KG triple (i.e., {subject, relation, object}), complex questions require manipulating sets of triples and reasoning over these. In Table 1, a question like *How many sports teams participated in that tournament?* requires numerical reasoning and answering the question in turn $\mathcal{T}_2$ relies on correctly interpreting $\mathcal{T}_1$.

Questions and answers in this dataset were elicited from human experts playing user and system roles as well as from crowd-workers. In a second stage, templates derived from the human-authored QA pairs were used to automatically augment the dataset. Human experts also suggested complex reasoning questions and derived templates thereof. Conversations were built as sequences of

[2] https://www.w3.org/TR/sparql11-query/

Nb. instances	197K
Nb. entities	12.8M
Nb. relations	2738
Nb. types	3064
Avg. turn length	9.5
Avg. entities per conversation	7.6
Avg. types per conversations	6.5
Avg. neighbourhood per turn	181.4 triples
Logical Reasoning, Quantitative Reasoning, Comparative Reasoning, Quantitative Reasoning Count, Comparative Reasoning Count, Verification, Simple Question	
Clarification, Coreference, Ellipsis	

Table 2: SPICE statistics (top); general question types (middle); linguistic phenomena (bottom).

QA pairs exploring paths in the KG. By construction, the QA pairs in a conversation are connected through one or several entities in the KG. Questions fall into two coarse categories, *simple* and *reasoning*-based, and the way QA pairs are organised in a sequence introduces various *conversational phenomena* which we summarize below.

Simple Questions are factoid questions, seeking information related to an entity (e.g., *Which tournament did Detroit Tigers participate in?* in Table 1) or set of entities (e.g., *What are the countries of those sports teams?*).

Reasoning Questions are complex questions which require the application of numerical and logical operators over sets of entities. For instance, to answer the question *How many sports teams participated in that tournament?* requires finding the set of sports teams that participated in a given tournament (e.g., *1909 World Series*) and taking its count. Questions in this category also involve General Entities (GE) such as *tournament*, in addition to Named Entities (NE), and multiple entities (both NE and GE) in a single question (e.g., *Which tournaments have less number of participating sports teams than 1909 World Series?*). Some question types also combine multiple reasoning operators.

Conversations contain sequences of mixed-initiative interactions where the system requests clarification on ambiguous questions. Conversations also include discourse phenomena such as coreference (e.g., *Which sports team was the champion of that tournament?* in Table 1) and ellipsis (e.g., *And what about 1910 World Series?* as a follow up question to *How many sports teams participated in that tournament?*).

There are 10 question types and 47 question subtypes. In Table 2, we only list question types but provide all subtypes in Table 9 in Appendix A.

2.1 Question Semantics Described by Actions

Saha et al. (2018) envisaged CSQA as a benchmark for retrieval-based conversational question answering (Bordes et al., 2015; Dong et al., 2015; Jain, 2016; Lan et al., 2021). These methods embed natural language questions and KG triples into high dimensional spaces and rely on neural reasoning modules to match questions to candidate answers. Hence, questions do not have associated logical forms, only gold answers are available.

Our success in creating semantic parse annotations is partly due to the fact that CSQA provides useful KG information. Each interaction (i.e., user and system turn) comes with annotations about KG entities, types, and relation symbols as well as some information about the triple patterns involved in the question (illustrated in Table 1 with ENT, REL, TYP, and TRIPLE fields). It also provides information pertaining to question types and subtypes (see INTENT in Table 1).

Taking advantage of these annotations, follow-on work (Guo et al., 2018) defined a semantic parsing task over CSQA, modeling the meaning of questions as a sequence of actions. The set of actions encompasses find (or find_rev when the entity is in object position) to retrieve sets of entities in a subject (object) position, as well as actions operating on sets of entities (e.g., filter_type). For instance, the question in turn $\mathcal{T}_1$ in Table 1 would be parsed to [filter_type, find_rev, Q650855, P1923, Q500834], meaning "find the set of entities that are in relationship *participating team* with *Detroit Tigers* and then filter those that are of type *tournament*". A breadth-first search algorithm generates action-grammar annotations for each question and a sequence of grammar-actions is considered correct if upon execution it returns the gold answer. Subsequent work (Shen et al., 2019; Kacupaj et al., 2021; Marion et al., 2021) expanded this action-grammar greatly improving its coverage (i.e., the number of successfully annotated questions).

2.2 From Actions to SPARQL Queries

In this work, we take a step further and map CSQA natural language questions into vanilla SPARQL queries. We first analysed how intent is expressed

	ATIS	SParC	CoSQL	SPICE
Nb. Instances	1,658	4,298	3,007	197K
Avg. turn length	7.0	3.0	5.2	9.5
Domain	Single	Multi	Multi	Wikidata
Logical form	SQL	SQL	SQL	SPARQL
Database type	Rel	Rel	Rel	KG

Table 3: Conversational semantic parsing datasets (Rel: relational database; KG: knowledge graph).

in question types and subtypes and then manually defined SPARQL templates for each question subtype. A SPARQL template is a query with unspecified triple patterns in the WHERE clause. For instance, the template for the question in turn $\mathcal{T}_1$ is {SELECT ?x WHERE **triple**(?x, ENTITY, RELATION). ?x wdt:P31 TYPE.}. We finally modified the tool provided in Kacupaj et al. (2021) to automatically instantiate the SPARQL templates, providing annotations for the entire dataset (e.g., by filling missing slots and determining subject/object positions for **triple**(·) elements as SELECT ?x WHERE {?x wdt:P1923 wd:Q650855. ?x wdt:P31 wd:Q500834.}).

We imported the Wikidata snapshot provided by Saha et al. (2018) into a KG in a SPARQL server (see Appendix B for more details) and assessed the correctness of SPARQL queries by executing them and comparing results to gold answers. For some questions the annotation procedure did not produce a SPARQL parse that recovered the gold answer. In these rare cases, we redefined the answer if it did not affect the conversation flow or truncated the conversation up to that point.

Table 2 shows various statistics for SPICE while Table 3 compares it to related conversational datasets such as ATIS (Suhr et al., 2018), SParC (Yu et al., 2019b), and CoSQL (Yu et al., 2019a). As can be seen, SPICE contains a sizeable number of training instances, its conversations are longer, and the semantic parsing task is real-scale.

3 The Semantic Parsing Task

We consider the semantic parsing task over a sequence of dialogue turns $d = (d_1, d_2, \cdots, d_{|d|})$, where turn d_t corresponds to a user-system interaction with user question x_t and system answer a_t. Each turn has a conversation context c_t made of interactions d_i such that $i < t$. Given interaction d_t with context c_t and user question $x_t = (x_{t1}, x_{t2}, \cdots, x_{t|x_t|})$, our goal is to predict a SPARQL query $y_t = (y_{t1}, y_{t2}, \cdots, y_{t|y_t|})$ that represents the intent of x_t and, upon execution over knowledge-graph $\mathcal{K}$, yields denotation a_t. y_t is a sequence over a target vocabulary $\mathcal{V} = \mathcal{V}_f \cup \mathcal{V}_{\mathcal{K}}$ where $\mathcal{V}_f$ is fixed and contains SPARQL keywords (e.g., SELECT) and special tokens (e.g., beginning of sequence token, BOS), and $\mathcal{V}_{\mathcal{K}}$ contains all knowledge-graph symbols (e.g., entity IDs such as Q76 for *Barack Obama*).

We propose two approaches for this semantic parsing task which establish strong baseline performance and highlight various challenges. These differ in the way they handle large KG vocabularies and how they generate logical forms. Figure 1 provides a sketch of the two models discussed below.

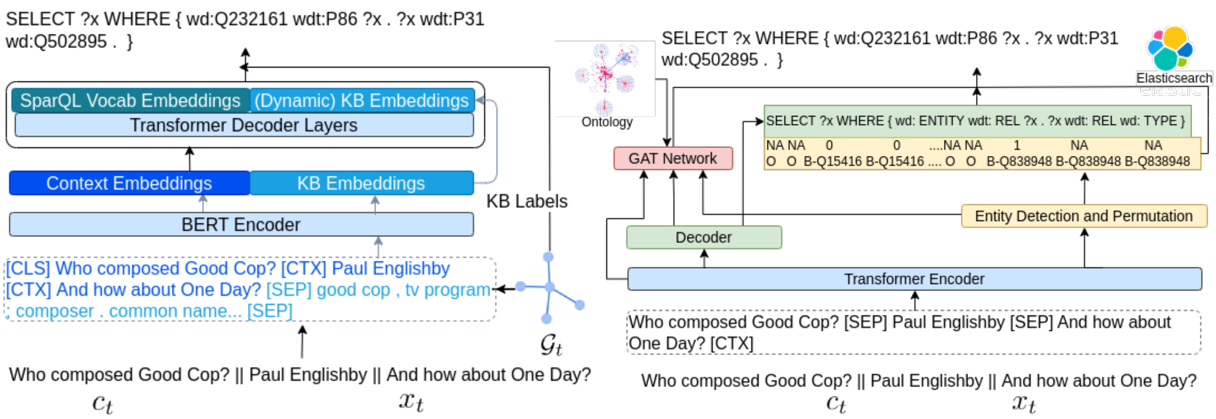
3.1 Parsing with a Single Decoder and Knowledge Subgraphs

Our first model is parameterised by an encoder-decoder Transformer neural network (Vaswani et al., 2017), and an adaptation of the semantic parsing architecture proposed in Gu et al. (2021).

Dynamic Vocabulary Since the KG vocabulary $\mathcal{V}_{\mathcal{K}}$ can be extremely large, we parse question x_t with a smaller vocabulary $\mathcal{V}_t \subseteq \mathcal{V}_{\mathcal{K}}$ which only contains KG symbols related to x_t. Following previous work (Gu et al., 2021; Marion et al., 2021), we assume the symbols related to x_t are those appearing in subgraph $\mathcal{G}_t$ of knowledge-graph $\mathcal{K}$, $\mathcal{G}_t \subseteq \mathcal{K}$. Given question x_t and its context c_t, we identify KG entities $\mathcal{E}_t = \{e_{t1}, e_{t2}, \cdots, e_{t|\mathcal{E}_t|}\}$ which correspond to mentions in x_t and c_t. We then obtain $\mathcal{G}_t$ by taking the one-hop neighbourhood for each entity $e_{ti} \in \mathcal{E}_t$. In other words, we include all KG triples (s, r, o) where the entity appears in subject $(s = e_{ti})$ or object position $(o = e_{ti})$. When e_{ti} is a subject, we include triple (e_{ti}, r, τ_o) where τ_o is the type of entity o; analogously, when e_{ti} appears in an object position, we add (τ_s, r, e_{ti}). For entities e_{ti} we include their types $\tau_{e_{ti}}$. When e_{ti} is a general entity (e.g., a type such as *tournament*) we add relations from $\mathcal{K}$ that have instances of type e_{ti} as their subject (object). The final vocabulary $\mathcal{V}_t$ contains all entities in $\mathcal{E}_t$, all relations r and types (τ_o, τ_s, and $\tau_{e_{ti}}$) found in the set of triples in $\mathcal{G}_t$.

Note that context c_t is defined as a window over the conversation so far. Following previous work (Marion et al., 2021; Kacupaj et al., 2021), we set the conversation context to the previous user-system interaction $c_t = \{d_{t-1}\}$.

Encoder-Decoder Model Our encoder is a BERT (Devlin et al., 2019) model fine-tuned on



(a) Encoder-decoder model with dynamic vocabularies. (b) Semantic parser based on Lasagne multi-task model.

Figure 1: Two modeling approaches to conversational semantic parsing.

our semantic parsing task. The decoder is a randomly initialised Transformer network (Vaswani et al., 2017). To account for the difference in initialisation between the encoder and decoder networks, we follow the training scheme proposed in Liu and Lapata (2019). We provide details in Appendix C.

The input to our semantic parser is a tuple $(x_t, c_t, \mathcal{G}_t)$ consisting of natural language question x_t, its context c_t, and subgraph $\mathcal{G}_t$ which we adapt to BERT's input format as follows (Gu et al., 2021). We concatenate the sequence of natural language questions and answers appearing in c_t and x_t, using the special token [CTX] as a delimiter and prepend the [CLS] token in the beginning of the sequence. Special token [SEP] denotes the end of sequence followed by the linearised KG subgraph $\mathcal{G}_t$. The linearisation procedure goes over entities in $\mathcal{G}_t$, enumerating their types and relations. Importantly, we denote entities by their label rather than their KG identifiers. The order of entities in $\mathcal{G}_t$ is random. Figure 1(a) shows an example of the input to our BERT-based encoder.

More formally, the encoder takes token sequences $x'_t = [\text{CLS}]x'_{t_{\text{text}}}[\text{SEP}]x'_{t_{\text{graph}}}[\text{SEP}]$ as input where $x'_{t_{\text{text}}}$ is the natural language subsequence and $x'_{t_{\text{graph}}} = (g^t_1, \cdots, g^t_{|\mathcal{G}_t|})$ is the sequence of knowledge-graph symbols from the linearised graph $\mathcal{G}_t$. Note that these knowledge-graph symbols constitute the target dynamic vocabulary $\mathcal{V}_t$ and $|\mathcal{G}_t|$ represents the number of KG symbols which is equal to the size of the target vocabulary $|\mathcal{V}_t|$. The encoder maps input sequences x'_t into sequences of continuous representations $\mathbf{z_t} = (\mathbf{z}_{t1}, \ldots, \mathbf{z}_{t|x_t|})$, and the decoder then generates the target SPARQL parse $y_t = (y_{t1}, \ldots, y_{t|y_t|})$ token-by-token autoregressively, hence modelling the conditional probability: $p(y_{t1}, \ldots, y_{t|y_t|}|x'_t)$.

The linearised graph $\mathcal{G}_t$ can exceed BERT's maximum number of input positions (which is 512). To avoid throwing away useful information, we adopt a solution similar to Gu et al. (2021). For question x_t with $\mathcal{G}_t$ containing k entities, we create k input sequences $x'^1_t, \cdots, x'^k_t$. These k sequences share the natural language subsequence but have different KG symbol subsequences. Given an input sequence $x'^1_t, \cdots, x'^k_t$, we obtain contextualised representations as $\mathbf{z}^1_t, \cdots, \mathbf{z}^k_t = \text{BERT}(x'^1_t, \cdots, x'^k_t)$.

The model further splits the sequence of continuous representations $\mathbf{z}^j_t$ into textual representations $\mathbf{z}^j_{t_{\text{text}}}$ and knowledge-graph symbols $\mathbf{z}^j_{t_{\text{graph}}}$ both of which are contextualised. We then average representations $\mathbf{z}_{t_{\text{text}}} = \text{AVG}(\mathbf{z}^j_{t_{\text{text}}})$ and feed them as input to the decoder (see Figure1(a)). From representations $\mathbf{z}^j_{t_{\text{graph}}}$, we derive the embeddings for the elements in the target dynamic vocabulary $\mathcal{V}_t$. Recall that the decoder parses input questions x_t using target vocabulary $\mathcal{V} = \mathcal{V}_f \cup \mathcal{V}_t$ which consists of a set of fixed ($\mathcal{V}_f$) and dynamic ($\mathcal{V}_t$) target tokens. The decoder then predicts the probability of each SPARQL token y_{ti} as $p(y_{t_i}|y_{t_{<i}}, x'^1_t, \cdots, x'^k_t) = \text{softmax}(\mathbf{W}_o \mathbf{h}^L_i)$ where $\mathbf{h}^L_i$ is the decoder top layer hidden representation at time step i. $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}_f \cup \mathcal{V}_t|}$ is the output embedding matrix with $\mathbf{W}_o = [\mathbf{W}_f; \mathbf{W}_t]$, where $[;]$ denotes matrix concatenation, $\mathbf{W}_f$ is the embedding matrix for the fixed target vocabulary, and $\mathbf{W}_t$ is derived from the encoder representations $\mathbf{z}^j_{t_{\text{graph}}}$.

3.2 Parsing with Multiple Decoders and an Ontology Classifier

Our second model is an adaptation of the Lasagne architecture proposed in Kacupaj et al. (2021).

Lasagne generates logical forms following a multi-stage approach where a backbone sketch is first predicted and then fleshed out. Their sketch is a sequence of actions from a custom grammar which we modify to be a sketch of SPARQL queries.

SPARQL Template Prediction Lasagne employs an encoder-decoder model based on Transformers (Vaswani et al., 2017) to convert a user question x_t in a conversation into a logical form template. The input to the encoder is the conversation context $c_t = \{d_{t-1}\}$ and user question x_t. Utterances are separated via [SEP] tokens, while the special context token [CTX] denotes the end of sequence (see Figure 1b). The input sequence is encoded via multi-head attention (Vaswani et al., 2017) to output contextualized representations which are then fed to the decoder to predict a sequence of actions (without grounding to KG elements) token-by-token. Instead, our decoder predicts SPARQL queries with place-holders for KG symbols. For instance, for the WHERE clause of turn $\mathcal{T}_1$ in Table 1, it predicts {ENTITY RELATION ?x. ?x wdt:P31 TYPE.} instead of {wd:Q5582479 wdt:P161 ?x. ?x wdt:P31 wd:Q502895.})

Entity Recognition and Linking An entity recognition module detects entities in the input and links them to the KG (Shen et al., 2019; Kacupaj et al., 2021). Initially, entity spans are identified using an LSTM which performs BIO sequence labelling.[3] Entity spans are subsequently linked to KG entities via an inverted index (created using Elasticsearch[4]) which maps entity labels to entity IDs. Once identified, the entities are further filtered and reordered so that they match their order of appearance in the SPARQL (see Figure 1(b)).

Predicting Types and Relations Finally, an ontology graph with types and relations appearing in $\mathbb{SPICE}$'s KG is constructed.[5] The graph is encoded with a Graph Attention Network (GAT; Velickovic et al. 2018) and the prediction of type and relation fillers for the SPARQL template is modeled as a classification task over graph nodes, given the conversational context and the decoder hidden state.

Learning All modules outlined above are trained in a multi-task manner, optimizing the weighted average of the following individual losses $L = \lambda_1 L^{\mathrm{F}} + \lambda_2 L^{\mathrm{G}} + \lambda_3 L^{\mathrm{R}} + \lambda_4 L^{\mathrm{O}}$ where L^{F} is the loss of the SPARQL template decoder, L^{G} is the type and relation prediction loss using the GAT network, L^{R} is the entity recognition loss, and L^{O} the entity reordering loss (and weights $\lambda_{1:4}$ are learned during training). We refer the interested reader to Kacupaj et al. (2021) for mode details.

[3] BIO labels for training are obtained by preforming string matching between entity annotations and user utterances.

[4] https://www.elastic.co/

[5] This graph would be substantially bigger for a semantic parsing system operating over the full Wikidata KG.

4 Results

We examine how the two models just described fare on different question types and subtypes. We report results on $\mathbb{SPICE}$ i.i.d train/valid/test splits (containing 152,391/16,813/27,797 conversations, respectively) but also create new splits that assess out-of-distribution generalisation. In all cases, following previous work (Saha et al., 2018; Kacupaj et al., 2021), we use execution-based automatic metrics. Micro *F1-score* evaluates question parses that return a set of entities, while *Accuracy* is used for question parses that evaluate to True/False or return a numerical value. In addition, we report *Exact Match* (EM) against the gold SPARQL parse.

4.1 Performance per Question Type

Table 4 shows our results on the $\mathbb{SPICE}$ i.i.d test split. BertSP variants differ in how they obtain the set of KG entities $\mathcal{E}_t$ (cf. Section 3.1) to build the dynamic vocabulary. BertSP_{G} has access to oracle entities, types, and coreference annotations which allows us to disassociate the complexity of the SPARQL generation task from the problem of grounding and disambiguating entities to KG symbols. Variants BertSP_{S} and BertSP_{A} do not have access to oracle annotations. BertSP_{S} grounds mentions to KG entities with a simple algorithm based on string matching (Marion et al., 2021); while BertSP_{A} relies on AllenNLP's Named Entity Recognizer (NER) and the Elasticsearch inverted index for Named Entity Linking (NEL). Both have to identify coreferring entities using the conversation context c_t. Both BertSP_{S} and BertSP_{A} use string matching for type linking (i.e., grounding general entities to KG types).

Note that it is not straightforward to perform oracle analysis for LasagneSP without compromising the model structure which predicts entities, their types, and relations in multiple stages.

Exact Match Performance We observe that execution based metrics (F1-score and Accuracy) are generally higher than EM. This is because in some

	BertSP$_G$		BertSP$_S$		BertSP$_A$		LasagneSP	
Question Type	F1	EM	F1	EM	F1	EM	F1	EM
Clarification	84.89	82.53	80.21	**77.69**	83.91	76.58	86.29	73.41
Logical Reasoning (All)	90.61	82.90	85.55	**66.89**	22.74	28.61	88.80	57.41
Quantitative Reasoning (All)	94.42	88.55	82.95	66.40	76.20	59.01	94.90	**91.47**
Comparative Reasoning (All)	96.23	87.39	90.44	73.80	69.56	39.37	94.20	**85.05**
Simple Question (Coreferenced)	88.96	86.53	83.19	**69.87**	76.51	58.83	84.73	60.90
Simple Question (Direct)	91.81	91.59	87.13	**80.69**	71.43	58.71	87.21	66.88
Simple Question (Ellipsis)	79.51	89.71	72.50	**71.67**	58.14	50.90	74.35	61.53
	AC	EM	AC	EM	AC	EM	AC	EM
Verification (Boolean)	90.10	77.24	79.72	**62.62**	37.16	24.90	34.89	26.72
Quantitative Reasoning (Count)	87.91	84.97	76.88	**73.20**	50.86	48.44	60.51	56.15
Comparative Reasoning (Count)	90.05	85.99	73.18	66.79	43.48	40.67	89.09	**83.69**
Overall	81.50	85.74	81.18	**70.96**	59.00	48.60	79.50	66.32

Table 4: Accuracy (AC), and exact match (EM) on $\mathbb{SPICE}$ i.i.d test split. BertSP$_G$ has access to oracle entities, types, and coreference annotations. Best EM predictions are shown in bold.

cases the SPARQL parse may be incorrect and still yield some results. For instance, a parse requiring the UNION of two graph patterns may yield a partially correct answer by only including one graph pattern; similarly, a parse can evaluate to False and agree with the gold answer just because it included a wrong relation symbol.

The Importance of Entity Grounding Not surprisingly, the model with access to oracle information (variant BertSP$_G$) obtains the best performance. Results improve not only for questions with entities referring to previous context but also indirectly for other types of questions. Since entities are correctly grounded in previous conversation turns c_t, the model operates with more accurate graphs $\mathcal{G}_t$ and richer dynamic vocabularies $\mathcal{V}_t$.

Both BertSP$_S$ and BertSP$_A$ perform coreference resolution using limited conversation context and thus performance decreases. These models also have to ground named (*Detroit Tigers*) and general (*tournament*) entity mentions to KG symbols. BertSP$_S$ which relies on string matching performs overall better than BertSP$_A$ which struggles with compound named entities such as *President of the Czech Republic*) and disambiguation during NEL (e.g., *Saint Barbara* the painting versus the Saint).

Model Comparison BertSP$_S$ and LasagneSP are similar in the way they handle NER/NEL with a task-specific approach, but differ in their conceptualization of the semantic parsing task (encoder-decoder vs. multi-tasking). LasagneSP outperforms BertSP$_S$ in Comparative, Quantitative, and Comparative-Count questions. These encompass many question subtypes with general entities which are very common in both training and testing. LasagneSP has access to *all* types and relations encoded with the graph network. In contrast, BertSP$_S$ relies on types which in the first place need to be present in the entity neighbourhood subgraph and then be preserved after truncating the input to fit the model's maximum sequence length. An advantage of BertSP$_S$ over LasagneSP, is that it allows for easier adaptation to new types and relations by relying on dynamic vocabularies, while LasagneSP would need to be retrained to accommodate them.

In Simple questions, where each question involves fewer but more diverse types, BertSP$_S$ predicts more accurate types (thanks to the input text and KG symbol contextualisation) and thus performs better. LasagneSP does poorly on Verification, Logical, and Quantitative-Count (which includes logical operators). This can be explained by a modelling limitation, i.e., it is not able to point to the same input entity more than once.

Errors in Predicted SPARQLs Manual inspection of SPARQL predictions revealed several common system errors including: prediction of erroneous entities and relations, failure to enumerate all required entities (for questions with multiple entities), and mistakes in argument order (i.e., entities and variables are correct but placed in incorrect subject/object positions). To a lesser extent, we also observed SPARQL queries with incorrect intent predictions and ill-formed syntax.

Phenomena	BertSP_G	BertSP_S	BertSP_A	LasagneSP
$\text{Coref}_{=-1}$	81.40	70.65	49.39	43.65
$\text{Coref}_{<-1}$	67.82	0	0	0
Ellipsis	75.93	54.33	26.39	46.54
Entities	83.37	65.40	41.64	66.52

Table 5: Exact match (EM) on SPICE i.i.d test set; questions are grouped into linguistic phenomena.

Unseen Combinations (Train/valid/test)	BertSP_S EM	LasagneSP EM
COUNTLOGIC	0.94	0
UNIONMULTI	19.74	16.89
VERIFY3	0	0

Table 6: Exact match (EM) for BertSP_S and LasagneSP on SPICE non-i.i.d splits.

4.2 Linguistic Analysis

Table 5 shows model performance across-different question subtypes aggregated for specific phenomena. These include Coreference, Ellipsis, and Multiple Entities (Entities). We distinguish between cases where coreference can be resolved in the previous turn ($\text{Coref}_{=-1}$) and further back in the conversation history ($\text{Coref}_{<-1}$). In addition, some question subtypes contain plural mentions, i.e., they are linked to multiple entities which the semantic parser must enumerate in order to build the correct parse. Ellipsis can be often resolved within the previous interaction ($\text{Coref}_{=-1}$), but not within the wider discourse context. Questions with multiple entities bring further disambiguation challenges. In Appendix A, we provide the list of question subtypes for each phenomenon in Table 5.

As can be seen, the oracle BertSP_G model which has access to gold annotations is superior to variants which rely on automatic entity and type linking. BertSP_S is better than LasagneSP at handling coreference within immediate context (i.e., $c_t = \{d_{t-1}\}$). Due to the fact that LasagneSP predicts entity positions in SPARQL, it is particularly bad at resolving mentions to multiple entities in the previous context or even multiple mentions of the same entity in the output parse (as is the case with Verification questions). Perhaps unsurprisingly, neither BertSP_S nor LasagneSP can resolve mentions to non-immediately preceding utterances. BertSP_S performs better than LasagneSP in questions with ellipsis; we conjecture that the input context and contextualisation of KG symbols help in grounding elided relation mentions. Ellipsis and multiple entities improve by a large margin with access to gold annotations (see BertSP_G in Table 5).

4.3 Generalisation

We further evaluate the models' ability to generalise by creating "query-based" splits (Finegan-Dollak et al., 2018), i.e., splits with query patterns seen only at test time. Our splits include: (a) question subtypes that involve a count operation over a union operator (COUNTLOGIC; individual operators are seen at training time but not the combination thereof); this split has 153,562/14,262/29,177 instances for training/validation/testing; (b) question subtypes that involve a union operator over two graph patterns with different relations (UNIONMULTI; the union of two graph patterns with the *same* relation is seen during training); this split contains 157,331/14,426/25,244 instances; and (c) verification questions with three entities involving 154,027/13,869/29,105 instances (VERIFY3; questions with 2 entities only are seen during training).

As shown in Table 6, both BertSP_S and LasagneSP perform poorly across different splits. While in some cases the models grasp the overall SPARQL structure for unseen questions (e.g., *Which watercourses are located in the neighbourhood of Bremen or are the tributaries of Ob? in* UNIONMULTI), they ignore specific query details and simply default to familiar patterns seen in the training (e.g., *Which people are the creators of The Theory of Everything or Ten Minutes to Live?*). In the UNIONMULTI split, the models produce an appropriate SPARQL template but systematically copy the *same* relation in *both* graph patterns. BertSP_S performs slightly better than LasagneSP; we hypothesize that contextualised KG embeddings occasionally help the model select different relations. We observe a similar trend for COUNTLOGIC and VERIFY3. Appendix D shows examples of unseen questions, their SPARQLs, and common errors.

5 Related Work

Much previous work on semantic parsing focuses on mapping stand-alone utterances to logical forms. Relatively few datasets have been constructed for *conversational* semantic-parsing (Suhr et al., 2018; Dahl et al., 1994; Yu et al., 2019b,a) partly due to the difficulty of eliciting annotations in an interactive context. As a result, existing benchmarks are either single-domain or small-scale (see the comparison in Table 3). For instance, although ATIS (Suhr et al., 2018) exemplifies several challenging long-range discourse phenomena (Jain and Lapata, 2021), it is restricted to a single domain with a simple database schema. SParC (Yu et al., 2019b) and CoSQL (Yu et al., 2019a) present cross-domain challenges in mapping natural language queries onto SQL, but the conversation length is fairly short and the databases relatively small-scale.

Large KGs, like Wikidata (Vrandečić, 2012), are becoming an increasing valuable source of information. Various question-answering datasets have been recently released (Dubey et al. 2019; Talmor and Berant 2018; Christmann et al. 2019, 2022, *inter alia*), which are either not conversational or contain sequences of dialogue turns but the questions are not annotated with executable queries like SPARQL. The CSQA dataset introduced in Saha et al. (2018) is conversational and covers a wide range of linguistic phenomena (e.g., ellipsis, coreference) but frames the QA task as an information retrieval problem. Follow-on work (Marion et al., 2021; Kacupaj et al., 2021; Shen et al., 2019) has used hand-crafted grammars to automatically obtain semantic annotations which are are not executable with a real KG engine (e.g., Blazegraph), and cumbersome to extend to new question intents.

6 Conclusion

In this work we introduce $\mathbb{SPICE}$, a conversational semantic parsing dataset over knowledge graphs. Our dataset contains SPARQL annotations which are executable on a real KG engine and requires handling complex questions, type, relation, and entity linking on a large scale. Moreover, it showcases multiple linguistic phenomena such as coreference and ellipsis. We establish two strong baselines for the semantic parsing task and present detailed analysis stratifying performance by question type and linguistic phenomena. We also study generalisation to unseen intents and create multiple dataset splits with different query patterns. To move forward conversational semantic parsing over large scale KGs would need to improve entity linking, modelling of conversation context, and generalisation capabilities. We hope our dataset will serve as a useful testbed for the development of conversational semantic parsers.

7 Acknowledgments

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of Amazon and the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1).

8 Limitations

Both models discussed in this work make simplifying assumptions. BertSP variants need to truncate the linearised graphs for computational cost reasons. LasagneSP works with a smaller graph ontology which can easily fit in memory. However, this restricts the model to predicting seen types or relations which is unrealistic. A real-world semantic parser should ideally have access to the full Wikidata. Our results show that both models do not generalise to unseen question intents, which is a known limitation of current neural sequence-to-sequence architectures (Furrer et al., 2020; Finegan-Dollak et al., 2018; Keysers et al., 2020; Kim and Linzen, 2020; Li et al., 2021). Finally, our results also suggest that there is scope for improvement in handling previous context (including questions and answers).

References

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 729–738, Beijing, China. ACM.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Conversational question answering on heterogeneous sources. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 144–154, Madrid, Spain. ACM.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

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

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269, Beijing, China. Association for Computational Linguistics.

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 *The Semantic Web – ISWC 2019*, pages 69–78, Cham. Springer International Publishing.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Daniel Peter Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv e-prints*, page arXiv:2007.08970.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, WWW '21, page 3477–3488, New York, NY, USA. Association for Computing Machinery.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Parag Jain and Mirella Lapata. 2021. Memory-Based Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 9:1197–1212.

Sarthak Jain. 2016. Question answering over knowledge base using factual memory networks. In *Proceedings of the NAACL Student Research Workshop*, pages 109–115, San Diego, California. Association for Computational Linguistics.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Comput. Surv.*, 54(5).

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 850–862, Online. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4483–4491. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4767–4780, Online. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Pierre Marion, Pawel Nowak, and Francesco Piccinno. 2021. Structured context and high-coverage grammar for conversational question answering over knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8813–8829, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc.

Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, page 117–126, New York, NY, USA. Association for Computing Machinery.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 705–713, New Orleans, Louisiana, USA. AAAI Press.

Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-task learning for conversational question answering over a large-scale knowledge base. 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 2442–2451, Hong Kong, China. Association for Computational Linguistics.

Alane Suhr, Srinivasan Iyer, and Yoav Artzi. 2018. Learning to map context-dependent sentences to executable formal queries. 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 2238–2249, New Orleans, Louisiana. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Denny Vrandečić. 2012. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 1063–1064, New York, NY, USA. Association for Computing Machinery.

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.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019a. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. 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 1962–1979, Hong Kong, China. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019b. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2022. Conversational information seeking. Https://arxiv.org/abs/2201.08808.

A The SPICE Dataset: Question Types and Subtypes

Table 9 provides the list of question types and subtypes in SPICE. For each question subtype we provide an example user question. For cases involving ellipsis and coreference, we include the conversation context (in grey colour).

Table 7 provides the list of question subtypes grouped grouped according to linguistic phenomena. Coreference ($_{=-1}$ and $_{<-1}$), Ellipsis, and Multiple Entities.

Coreference
More/Less ǀ Mult. entity type (Coreference) # More/-Less ǀ Single entity type (Coreference) # Single Entity (Coreference) # 2 entities, one direct and one indirect, object is indirect # 2 entities, one direct and one indirect, subject is indirect # 3 entities, 2 direct, 2(direct) are query entities, subject is corefered # one entity, multiple entities (as object) corefered # Count ǀ Logical operators (Coreference) # Count ǀ Single entity type (Coreference) # Count over More/Less ǀ Mult. entity type (Coreference) # Count over More/Less ǀ Single entity type (Coreference)
Ellipsis
Difference ǀ Single Relation (Ellipsis) # Intersection ǀ Single Relation (Ellipsis) # Union ǀ Single Relation (Ellipsis) # More/Less ǀ Mult. entity type (Ellipsis) # More/Less ǀ Single entity type (Ellipsis) # object parent is changed, subject and predicate remain same # Incomplete count-based ques # Count over More/Less ǀ Mult. entity type (Ellipsis) # Count over More/Less ǀ Single entity type (Ellipsis)
Multiple Entities
Difference ǀ Multiple Relation # Intersection ǀ Multiple Relation # Union ǀ Multiple Relation # Atleast/ Atmost/ Approx. the same/Equal ǀ Mult. entity type # Min/Max ǀ Mult. entity type # More/Less ǀ Mult. entity type # More/Less ǀ Mult. entity type (Ellipsis) # More/Less ǀ Mult. entity type (Coreference) # Mult. Entity (Simple Question Direct and Coreference) # one entity, multiple entities (as object) coreferred # Count over Atleast/ Atmost/ Approx. the same / Equal ǀ Mult. entity type # Count ǀ Mult. entity type # Count over More/Less ǀ Mult. entity type # Count over More/Less ǀ Mult. entity type (Ellipsis) # Count over More/Less ǀ Mult. entity type (Coreference)

Table 7: Question subtypes grouped according to linguistic phenomena.

Table 8 provides the list of unseen question subtypes for each of the non-i.i.d splits.

COUNTLOGIC
Count ǀ Logical operators # Count ǀ Logical operators (Coreference)
UNIONMULTI
Union ǀ Multiple Relation
VERIFY3
3 entities, 2 direct, 2(direct) are query entities, subject is indirect # 3 entities, all direct, 2 are query entities

Table 8: Unseen question subtypes in SPICE non-i.i.d splits.

B Creating a Knowledge Graph from the CSQA Data

Deploying a full copy of Wikidata locally as a standalone service requires huge resources in addition to cluster dependent tweaking to obtain fast query processing and high-performance.[6] To enable easier deployment and fast access for research purposes we created a smaller graph from the CSQA data files. We mapped the contents of these files[7] onto triples which we subsequently converted to `ttl` format[8] with full URI to allow loading them to the KG query engine. We also filled missing information where possible, for example, missing relations such as "instance of" was filled with relation P31 and added data type information when this was omitted from the original files.

We used Blazegraph[9] to deploy the local server, which uses only CPU-based resources and has access to 150G of RAM. Further details along with the script to host the server will be released upon acceptance.

C BertSP Model Configuration

Our model is implemented using pytorch (Paszke et al., 2019). For all experiments, we used the ADAM optimizer (Kingma and Ba, 2015) with 20,000 BERT warmup steps and 10,000 steps for decoder warm up. We use separate optimizers for the BERT encoder and decoder. BERT was fine-tuned during training with the initial learning rate set to 0.00002. A learning rate of 0.001 was set for the rest of model parameters. Our model was trained for 100,000 steps; we used 4 GPUs with 12GB of memory. We performed model selection on the validation set. We report results with the best performing model which had 6 decoder layers.

D Examples on Generalisation Splits

Table 10 shows examples from the generalisation splits: similar question subtypes see during train-

[6] https://www.mediawiki.org/wiki/Wikidata_Query_Service/User_Manual#Standalone_service

[7] Described at https://amritasaha1812.github.io/CSQA/download_CQA/

[8] http://www.w3.org/TR/turtle/

[9] https://blazegraph.com/

ing, unseen question subtype, and error on prediction. The most common error across different splits is that models use similar SPARQLs seen during training but fail to adapt them to the details (entities, types, relations, argument positions) in the unseen question subtype. Other errors encompass using the incorrect SPARQL query (incorrect question intent) and incorrect entities and types.

Table 9: Full list of question subtypes (intents) in SPICE. For each subt-type we show an example question, whenever the question subtype involves a conversational phenomenon (coreference or ellipsis); previous conversation interactions necessary for the interpretation of the question are shown in grey.

Question Subtype	Example User Question
Clarification	
Simple Question ǀ Single Entity (Coreference)	U: *Which political territory is that sporting event located in?* S: *Did you mean Speed skating at the 2010 Winter Olympics – Men's 500 metres?* U: *Yes*
Logical Reasoning (All)	
Difference ǀ Multiple Relation	U: *Which people were awarded with Order of Merit for Arts and Science and are not working as singer?*
Difference ǀ Single Relation	U: *Which international organizations had Poland but not Bulgaria as their member?*
Difference ǀ Single Relation (Ellipsis)	U: *Which city was Pierre Laffont born in?* S: *Marseille* U: *Which administrative territories are the sister cities of that city?* S: *Shanghai, Odessa, Naples* U: *But not Bologna*
Intersection ǀ Multiple Relation	U: *Which human settlements are situated close to Trave and have an adjacent border with Herzogtum Lauenburg?*
Intersection ǀ Single Relation	U: *Which works of art were filmed at Edinburgh and Berlin?*
Intersection ǀ Single Relation (Ellipsis)	U: *Which language does José María Lassalle speak in?* S: *Spanish* U: *And also Sergio Gil*
Union ǀ Multiple Relation	U: *Which watercourses are located in the neighbourhood of Bremen or are the tributaries of Ob?*
Union ǀ Single Relation	U: *Which people are the creators of The Theory of Everything or Ten Minutes to Live?*
Union ǀ Single Relation (Ellipsis)	U: *What is the profession of Mai Yamada?* S: *announcer* U: *Or Kazimierz Rogoyski?*
Quantitative Reasoning (All)	
Min/Max ǀ Single entity type	U: *Which musical instruments are played by min number of people?*
Min/Max ǀ Mult. entity type	U: *Which organizations are the main building contractors of max number of architectural structures and buildings?*
Atleast/ Atmost/ Approx. the same/Equal ǀ Single entity type	U: *Which musical instruments are played by exactly 5 people?*
Atleast/ Atmost/ Approx. the same/Equal ǀ Mult. entity type	U: *Which events are demonstrated in atleast 3 prints and genres of sculpture?*
Comparative Reasoning (All)	
More/Less ǀ Mult. entity type	U: *Which landforms are known for containing lesser number of chemical compounds or minerals naturally than Stetind pegmatite?*
More/Less ǀ Mult. entity type (Ellipsis)	U: *Which landforms are known for containing lesser number of chemical compounds or minerals naturally than Stetind pegmatite?* S: *Euboea, Izalco, Mount Nyiragongo* U: *And also tell me about Tuften quarry?*
More/Less ǀ Mult. entity type (Coreference)	U: *Which administrative territory is that person a civilian of?* S: *Spain* U: *Which administrative territories are the countries of origin of lesser number of television programs or works of art than that administrative territory?*
More/Less ǀ Single entity type	U: *Which television programs have been dubbed by more number of people than Puss in Boots: The Three Diablos?*
More/Less ǀ Single entity type (Ellipsis)	U: *Which television programs have been dubbed by more number of people than Puss in Boots: The Three Diablos?* S: *House, K-On!, K-On!!* U: *And also tell me about Chip 'n Dale Rescue Rangers?*
More/Less ǀ Single entity type (Coreference)	U: *Which languages are max number of literary works composed in?* S: *English* U: *Which languages are the mother tongues of less number of people than that language?*
Simple Question (Direct)	
Simple Question	U: *Which type of sport did Amel Tuka participate in?*
Single Entity	U: *What is the capital of Sweden?*
Mult. Entity	U: *Who were the writers of On being and essence, De vegetabilis et plantis libri septem and Historia de regibus Gothorum, Vandalorum et Suevorum?*
Simple Question (Ellipsis)	
only subject is changed, parent and relation remains same	U: *Which organizations are the sponsors of Janice Anderson?* S: *Montrail, Patagonia, Inc.* U: *And also tell me about Manikala Rai?*
object parent is changed, subject and relation remain same	U: *Which watercourses are situated nearby Munich?* S: *Eisbach, Würm, Isar* U: *And which river?*
Simple Question (Coreference)	
Mult. Entity	U: *Which releases have Motown as their record label?* S: *What's Going On, Got to Be There, Can't Slow Down* U: *Which genre do those releases belong to?*
Single Entity (Coreference)	U: *Which narrative location is The Penalty set in?* S: *San Francisco* U: *Which color is associated with that film genre?*
Verification (Boolean) (All)	
2 entities, both direct	U: *Is Zugspitze located in Germany?*
2 entities, one direct and one corefered, object is corefered	U: *Which university was Eden Stiles educated at?*

Continued on next page

Table 9 – *Continued from previous page*

Question Type/Subtype	Example User Question
	S: *University of Michigan* U: *And what about C. V. Raman?* S: *University of Madras* U: *Was Ravindra Wijegunaratne educated at that university?*
2 entities, one direct and one corefered, subject is corefered	U: *Which German business organization was Gustav Peichl a member of?* S: *Academy of Arts, Berlin* U: *What was designed by that person?* S: *Millennium Tower* U: *Does that tower block belong to Austria?*
3 entities, 2 direct, 2(direct) are query entities, subject is corefered	U: *Which administrative territory was Gary Collier born at?* S: *Fort Worth* U: *Is that administrative territory a sister city of Adamsville, New Brunswick and Yuen Long Kau Hui?*
3 entities, all direct, 2 are query entities	U: *Is Aix-en-Provence partner town of Baton Rouge and Hemmatabad, Alborz?*
one entity, multiple entities (as object) coreferred	U: *Which armed conflicts are Battle of the Argeş or Battle of the Yellow Sea a part of?* S: *Romania during World War I, Russo-Japanese War* U: *Did those armed conflicts fight in Rui Natsukawa?*
Quantitative Reasoning (Count) (All)	
Incomplete count-based ques	U: *How many people influenced Chris Marker?* S: *1* U: *And also tell me about Ada Yonath?* S: *1* U: *And what about Mikhail Bakunin?*
Count over Atleast/ Atmost/ Approx. the same/Equal|Mult. entity type	U: *How many cities are the terminus locations of atleast 5 thoroughfares and roads?*
Count over Atleast/ Atmost/ Approx. the same/Equal|Single entity type	U: *How many musical instruments are played by exactly 2 people?*
Count | Logical operators	U: *How many bodies of water or watercourses are situated nearby Lübeck?*
Count | Logical operators (Coreference)	U: *Which administrative territory is the native country of Carolina Goic Boroevic?* S: *Chile* U: *Who is the head of the government of that administrative territory?* S: *Michelle Bachelet* U: *What is the capital of that administrative territory?* S: *Santiago* U: *How many capitals or cities are sister towns of that city?*
Count | Mult. entity type	U: *How many people starred in Django Kill or Shatterday?*
Count | Single entity type	U: *How many people starred in Captain America: Civil War?*
Count | Single entity type (Coreference)	U: *Which armed conflict did Lionel of Antwerp, 1st Duke of Clarence take part in?* S: *Hundred Years' War* U: *How many people did that armed conflict engage in?*
Comparative Reasoning (Count) (All)	
Count over More/Less | Mult. entity type	U: *How many administrative territories have adopted lesser number of holidays and people as patron saint than Santo Stefano al Mare?*
Count over More/Less | Mult. entity type (Ellipsis)	U: *How many administrative territories have adopted lesser number of holidays and people as patron saint than Santo Stefano al Mare?* S: *296* U: *And what about San Donato Milanese?*
Count over More/Less | Mult. entity type (Coreference)	U: *Which administrative territories are Luigi Einaudi the head of state of and have UTC+01:00 as their time zone?* S: *Italy* U: *How many administrative territories are the origins of greater number of literary works or releases than that administrative territory?*
Count over More/Less | Single entity type	U: *How many legislatures represent lesser number of states than East Bengal Legislative Assembly?*
Count over More/Less | Single entity type (Ellipsis)	U: *How many legislatures represent lesser number of states than East Bengal Legislative Assembly?* U: *207* U: *And how about Estates of Curaçao?*
Count over More/Less | Single entity type (Coreference)	U: *Which french administrative division was Philippe Esnault born in?* S: *Alençon* U: *Which occupation has that person as his/her 's career?* S: *historian* U: *Which administrative territory is the native country of that person?* S: *France* U: *How many administrative territories inspired less number of fictional locations than that administrative territory?*

	COUNTLOGIC	
SEEN	Union | Single Relation *Which people are the creators of The Theory of Everything or Ten Minutes to Live?*	`SELECT ?x WHERE { {wd:Q15079318 wdt:P162 ?x. ?x wdt:P31 wd:Q502895.} UNION {wd:Q7699260 wdt:P162 ?x. ?x wdt:P31 wd:Q502895.} }`
	Count | Single entity type *How many people starred in Captain America: Civil War?*	`SELECT (COUNT(*) AS ?count) WHERE { wd:Q18407657 wdt:P161 ?x. ?x wdt:P31 wd:Q502895.}`
UNSEEN	Count | Logical operators *How many national association football teams or national sports teams represent Slovenia?*	`SELECT (COUNT (DISTINCT ?x) AS ?count) WHERE { {?x wdt:P1532 wd:Q215. ?x wdt:P31 wd:Q6979593.} UNION {?x wdt:P1532 wd:Q215. ?x wdt:P31 wd:Q1194951.} }`
PRED		`SELECT (COUNT(DISTINCT ?x) AS ?count) WHERE { {?x wdt:P1532 wd:Q215. ?x wdt:P31 wd: Q6979593 .} UNION {?x wdt:P1532 wd:Q215. ?x wdt: P31 wd: Q6979593.} }`
	UNIONMULTI	
SEEN	Union | Single Relation *Which people are the creators of The Theory of Everything or Ten Minutes to Live?*	`SELECT ?x WHERE { ?x wdt:P915 wd:Q1247373. ?x wdt:P31 wd:Q838948.}`
	Count | Mult. entity type *How many people starred in Django Kill or Shatterday?*	`SELECT (COUNT(DISTINCT ?x) AS ?count) WHERE { {wd:Q1261875 wdt:P161 ?x. ?x wdt:P31 wd:Q502895.} UNION {wd:Q7490688 wdt:P161 ?x. ?x wdt:P31 wd:Q502895.} }`
UNSEEN	Union | Multiple Relation *Which administrative territories are the origin of Les Chics Types or are the native countries of Robert Kuraś?*	`SELECT ?x WHERE { {wd:Q3231475 wdt:P495 ?x. ?x wdt:P31 wd:Q15617994.} UNION {wd:Q9310937 wdt:P27 ?x. ?x wdt:P31 wd:Q15617994.} }`
PRED		`SELECT ?x WHERE { {wd:Q3231475 wdt:P495 ?x. ?x wdt:P31 wd:Q15617994.} UNION {wd:Q9310937 wdt:P495 ?x. ?x wdt:P31 wd:Q15617994.} }`
	VERIFIY3	
SEEN	2 entities, both direct *Is Zugspitze located in Germany?*	`ASK {wd:Q3375 wdt:P17 wd:Q183.}`
UNSEEN	3 entities, all direct, 2 are query entities *Is Violet Oakley a civilian of United States of America and Scheden?*	`ASK {wd:Q30 wdt:P27 wd:Q1226556. wd:Q557427 wdt:P27 wd:Q1226556.}`
PRED		`ASK {wd:Q1226556 wdt:P27 wd:Q30. wd:Q557427 wdt:P27 wd:Q557427.}`

Table 10: Generalisation splits, unseen question subtypes, support question subtypes seen during training, and example common errors on unseen predictions.