Context Dependent Semantic Parsing: A Survey

Zhuang Li, Lizhen Qu, Gholamreza Haffari
Faculty of Information Technology
Monash University
firstname.lastname@monash.edu

Abstract

Semantic parsing is the task of translating natural language utterances into machine-readable meaning representations. Currently, most semantic parsing methods are not able to utilize contextual information (e.g. dialogue and comments history), which has a great potential to boost semantic parsing performance. To address this issue, context dependent semantic parsing has recently drawn a lot of attention. In this survey, we investigate progress on the methods for the context dependent semantic parsing, together with the current datasets and tasks. We then point out open problems and challenges for future research in this area. The collected resources for this topic are available at: https://github.com/zhuang-li/Contextual-Semantic-Parsing-Paper-List.

1 Introduction

Semantic parsing is concerned with mapping natural language (NL) utterances into machine-readable structured *meaning representations* (*MRs*). These representations are in the formats of formal languages, e.g. Prolog, SQL, and Python. A formal language is typically defined by means of a formal *grammar*, which consists of a set of rules. Following the convention of the chosen formal language, *MRs* are also referred to as logical forms or programs. An *MR* is often executable in a (programming) environment to yield a result (e.g. results of SQL queries) enabling automated reasoning (Kamath and Das, 2018).

Most research work on semantic parsing treats each NL utterance as an *independent* input, ignoring the text surrounding them (Kamath and Das, 2018), such as interaction histories in dialogues. The surrounding text varies significantly across different application scenarios. In a piece of free text, we refer to the surrounding text of a current utterance as its *context*. The context is different with respect to different utterances. In our sequel, we differentiate between context *independent* semantic parsing (CISP) and context *dependent* semantic parsing (CDSP) by whether a corresponding parser utilizes context information. A knowledge base or a database (on which a *MR* is executed for the purpose of question answering) can be considered as context as well (Krishnamurthy and Mitchell, 2012; Liang, 2016). This type of context does not change with respect to the utterances. In this survey, we only consider the former kind of context which does vary with different utterances.

D	Database about pets
Q_1	What are the different pet types?
S_1	SELECT DISTINCT pettype FROM pets
Q_2	For each of those, what is the maximum age?
S_2	SELECT max(pet_age), pettype FROM pets GROUP BY pettype
Q_3	What about the average age?
S_3	SELECT avg(pet_age), pettype FROM pets GROUP BY pettype

Table 1: An example of CDSP from SPARC (Yu et al., 2019b), where each SQL query S_i is the *MR* of the question Q_i.

The utilization of context in semantic parsing imposes both challenges and opportunities. As shown in Table 1, one challenge is to resolve references, such as *those* in "For each of those, what is the maximum

Proceedings of the 28th International Conference on Computational Linguistics, pages 2509–2521
Barcelona, Spain (Online), December 8-13, 2020

age”. This example shows also another challenge caused by elliptical (incomplete) utterances. The sentence “What about the average age?” alone misses information about the database table and the column *pettype*. The incomplete meaning needs to be complemented by the discourse context. Compared with CISP, which usually assumes that the information within the utterance is complete, CDSP is expected to tackle challenges posed by involving context in the parsing process (Liang, 2016; Suhr et al., 2018; Zhang et al., 2019; Liu et al., 2020). In addition, tackling the above challenges provides us with more opportunities to inspect the linguistic phenomena which could influence semantic parsing. Our survey on CDSP fills the gap in the literature, as the recent surveys in the semantic parsing research mainly focus on CISP (Kamath and Das, 2018; Zhu et al., 2019).

This paper is organised as follows. We start with providing a brief and fundamental understanding of CISP in §2. We then present a comprehensive organization of the recent advances in CDSP in §3. We discuss current CISP tasks, datasets, and resources in §4. Finally, we cover open research problems in §5, and conclude by providing a roadmap for future research in this area.

2 Background

CISP aims to learn a mapping $\pi_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, which translates an NL utterance $x \in \mathcal{X}$ into an *MR* $y \in \mathcal{Y}$. An *MR* y can be executed in a programming environment (e.g. databases, knowledge graphs, etc.) to yield a result z, namely denotation. The structure of an *MR* takes a form of either a tree or graph, depending on its underlying formal language. The languages of *MRs* are categorized into three types of formalism : logic based (e.g. first order logic), graph based (e.g. AMR (Banarescu et al., 2013)), and programming languages (e.g. Java, Python) (Kamath and Das, 2018). Some semantic parsers explicitly apply a production grammar to yield *MRs* from utterances. Such a grammar consists of a set of production rules, which define a list of candidate derivations for each NL utterance. Each derivation deterministically produces a grammatically valid *MR*.

2.1 Semantic Parsing Models

Given an utterance $x \in \mathcal{X}$ and its paired *MR* $y \in \mathcal{Y}$, a CISP model can form a *conditional* distribution $p(y|x)$. The model learning can be supervised by either utterance-*MR* pairs or merely utterance-denotation pairs. If only denotations are available, a widely used approach (Kamath and Das, 2018) is to marginalize over all possible *MRs* for a denotation z, which leads to a *marginal* distribution $p(z|x) = \sum_y p(z, y|x)$. A parsing algorithm aims to find the optimal *MR* in the combinatorially large search space. We coarsely categorize the existing models into: symbolic approaches, neural approaches, and neural-symbolic approaches based on the category of machine learning methodology and whether any production grammars are explicitly used in models.

Symbolic Approaches A symbolic semantic parser employs production grammars to generate candidate derivations and find the most probable one via a scoring model. The scoring model is a statistical or machine learning model. Each derivation is represented by handcrafted features extracted from utterances or partial *MRs*. Let $\Phi(x, d)$ denote the features of a pair of utterance and derivation, and $G(x)$ be the set of candidate derivations based on x. A widely used scoring model is the log linear model (Zettlemoyer and Collins, 2012; Kamath and Das, 2018).

$$p(d|x) = \frac{\exp(\boldsymbol{\theta}\Phi(x, d))}{\sum_{d' \in G(x)} \exp(\boldsymbol{\theta}\Phi(x, d'))} \tag{1}$$

where $\boldsymbol{\theta}$ denotes the model parameters. If only utterance-denotation pairs are provided at training time, a model marginalizes over all possible derivations yielding the same denotations by $p(z|x) = \sum_d p(z, d|x)$ (Krishnamurthy and Mitchell, 2012; Liang, 2016). Those corresponding parsers further differentiate between graph-based parsers (Flanigan et al., 2014; Zettlemoyer and Collins, 2012) and shift-reduce parsers (Zhao and Huang, 2014) due to the adopted parsing algorithms and the ways to generate derivations. From a machine learning perspective, these approaches are also linked to a structured prediction problem.

Neural Approaches Neural approaches apply neural networks to translate NL utterances into *MRs* without using production grammars. These approaches formulate semantic parsing as a machine translation problem by viewing NL as the source language and the formal language of *MRs* as the target language.

Most work in this category adopts SEQ2SEQ (Sutskever et al., 2014) as the backbone architecture, which consists of an encoder and a decoder. The encoder projects NL utterances into hidden representations, whereas the decoder generates linearized *MRs* sequentially. Both encoders and decoders employ either recurrent neural networks (RNN) (Goodfellow et al., 2016) or Transformers (Vaswani et al., 2017). Note that, these methods do not apply any production grammars to filter out syntactically invalid *MRs*.

The variants of the SEQ2SEQ based models also explore structural information of *MRs*. SEQ2TREE (Dong and Lapata, 2016) utilizes a tree-structured RNN as the decoder, which constrains generated *MRs* to take syntactically valid tree structures. The COARSE2FINE model (Dong and Lapata, 2018) adopts a two-stage generation for the task. In the first stage, a SEQ2SEQ model is applied to generate *MR* templates, which replace entities in *MRs* by slot variables for a high-level generalization. In the second stage, another SEQ2SEQ model is applied to fill the slot variables with the corresponding entities.

Neural-Symbolic Approaches In order to ensure the generated *MRs* to be syntactically valid without compromising the generalization power of neural networks, neural-symbolic approaches fuse both symbolic and neural approaches by applying production grammars to the generated *MRs*; then the derivations are scored by neural networks.

The majority of these methods linearize derivations such that they are able to leverage SEQ2SEQ (Liang et al., 2016; Yin and Neubig, 2018; Guo et al., 2019b). At each time step, the decoder of these methods emits either a parse action or a production rule, leading to a grammatically valid *MR* at the end. these works produce derivations by varying grammars. NSM (Liang et al., 2016) uses a subset of Lisp syntax. TRANX (Yin and Neubig, 2018) defines the grammars in Abstract Syntax Description Language, while IRNET (Guo et al., 2019b) considers the context-free grammar of a language called SemQL.

There are also neural-symbolic approaches adopting neural architectures other than SEQ2SEQ. One of such examples is (Andreas et al., 2016), which adopts a dynamic neural module network (DNMN) to generate *MRs*.

2.2 Evaluation

In semantic parsing, *exact match accuracy* is the most commonly used evaluation metric. With *exact match accuracy*, the parsing results are considered correct only when the output *MR*/denotations exactly match the string of the ground truth *MR*/denotations. One flaw of the evaluation metric is that some types of MRs (e.g., SQL) do not hold order constraints. Yu et al. (2018) proposed a metric *set match accuracy* to evaluate the semantic parsing performance over SQLs, which treats each SQL statement as a set of clauses and ignore their orders.

Due to the variety of domains and languages over different datasets, it is difficult to measure all semantic parsing methods in a unified framework. To address this issue, Yu et al. (2018), Yu et al. (2019b) and Yu et al. (2019a) built different shared-task platforms with leaderboard for semantic parsing evaluation on the common datasets and consistent evaluation metrics.

3 Context Dependent Semantic Parsing

Context dependent semantic parsing involves modelling of context in the parsing process. For each current NL utterance, we define its *context* as the information beyond this utterance. With this definition, there are two types of context for semantic parsing, *local* context and *global* context. The *local* context for an utterance is the text and multimedia content surrounding it, which is meaningful only for this utterance. In plain texts, the concept of local context is also quite close to discourse, which is defined as a group of collocated, structured, coherent sentences (Parsing, 2009). In contrast, its *global* context is the information accessible to more than one utterance, including databases and external text corpora, images or class environment (Iyer et al., 2018). The content of local context varies for each NL utterance while the global context is always static. The work in our survey is only concerned with local context. Therefore, we always refer to ”local context” as ”context” in the following sections.

Context provides additional information to resolve ambiguity and vagueness in current utterances. For semantic parsing, one type of ambiguity is caused by references in current utterances, which need to be resolved to previously mentioned objects and relations. References may include explicit or implicit lexical triggers, such as *those* in ”For each of those, ...” in our introductory example (Table 1). Another ambiguity illustrated by the same example is resulted by ellipsis. The previous context provides constraints to restrict the scope of possible *MRs* indicated by current utterances. In addition, context provides information to disambiguate word senses and entities, and link them to knowledge bases to enable complex reasoning. However, semantic parsing literature largely neglects word sense disambiguation, which is regarded as an AI complete problem (Navigli, 2009). Last but not least, context allows to exploit discourse coherence for semantic parsing. Coherence relations characterize structural relationships between sentences, thus limit the search space of parse candidates for the following utterances of current ones.

Formally, a context dependent parser takes both an input utterance x_i and its context C_i, where C_i could include a broad range of multimedia content. And we consider a group of inter-related utterances with the union set of their context as one *interaction*, $I = (\mathbf{x}, \mathbf{C})$, where $\mathbf{x} = [x_1, ..., x_i, ..., x_T]$ and $\mathbf{C} = \cup_{i=1}^{T} C_i$. Currently, most CDSP work focus on the research problems of context C_i regarding the history utterances, *MRs*, denotations. Such a parser learns a mapping from a current utterance x_i to an *MR* y_i by $\pi_\theta(x_i, C_i)$.

3.1 Symbolic Approaches

Existing symbolic approaches formulate CDSP as a structured prediction problem by including contextual information into their feature models. Their models capture $p(d_i|x_i, C_i)$ by including context as a condition. Both Zettlemoyer and Collins (2009) and Srivastava et al. (2017) divide the parsing process into two steps: i) generate initial parses using CISP; ii) complete initial parses using contextual information. In contrast, Long et al. (2016) parses a sequence of utterances in one step. In all those work, symbolic features are used to represent contexts.

In two-step approaches, Zettlemoyer and Collins (2009) and Srivastava et al. (2017) differ in the details of individual steps. In the first step, Zettlemoyer and Collins (2009) extends *MRs* with predicates representing references, while Srivastava et al. (2017) generates a set of context independent parses for each utterance. In the second step, Zettlemoyer and Collins (2009) collects possible derivations by applying three heuristic rules to replace references with entities in context and extend initial *LFs* with constraints, then finds the best derivations according to a linear model. In (Srivastava et al., 2017), their model expands the initial parse set with parses selected from context using heuristic rules, then finds the best parses in the expanded set. Their feature model includes a multinomial random variable indicating the current hidden state of discourse.

The shift-reduce parser in (Long et al., 2016) generates derivations for a whole utterance sequence. This method stores the previously generated derivations in a stack, performs a sequence of *shift* and *build* operations to generate *LFs*. In its feature model, a context is represented by a sequence of past *LFs* and a random variable denoting the current world state.

Utterances and *MRs* histories form a context of a CDSP parser. The common practice is to extract handcrafted features from both utterances and *MRs* to represent contexts. Some typical feature patterns are as follows:

Utterance In (Srivastava et al., 2017), they consider indicator features of lexical triggers, whether the current utterance is repeated, as well as the position of the current utterance in an interaction.

Meaning Representations In (Zettlemoyer and Collins, 2009), there is a feature indicating if the predicates exist in the history *LFs*. Such feature allows the model to learn to copy the segments from the context that contains the expected predicates. Long et al. (2016) adopts the feature indicating if the argument in current *MR* is one of arguments in the last *MR*. Srivastava et al. (2017) uses the combinations of predicates in successive turns as the indicator features.

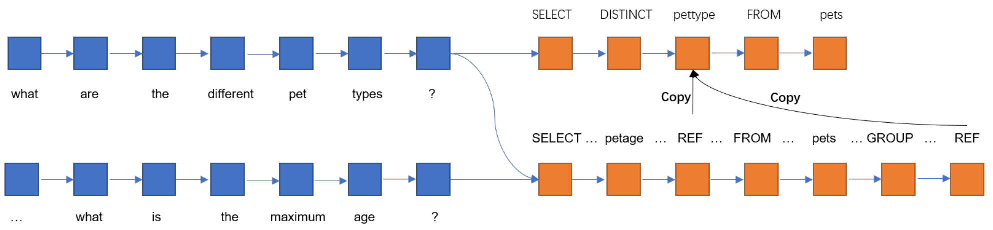

Figure 1: Coreference resolution architecture of (Chen and Bunescu, 2019). Considering the example in Table 1, Chen and Bunescu (2019) firstly generates a *MR* template for Q_2 as ”SELECT max(petage), *REF* FROM pets GROUP BY *REF*”. The *REF* tokens would then be replaced with the ”pettype” from the precedent *MR*.

3.2 Neural Approaches

Existing neural CDSP methods extend the SEQ2SEQ architecture to incorporate contextual information in two ways. The first approach is to build context-aware encoders to encode historical utterances or *MRs* into neural representations, which provide decoders contextual information to resolve ambiguity in current utterances. As previously predicted *MRs* provide the constraints and information missed in current utterances, the second approach is to utilize context-aware decoders to reuse or revise those predicted *MRs* for generating current *MRs*.

Context-aware Encoders Encoders of CDSP methods differentiate between utterance encoders and *MR* encoders. Utterance encoders construct neural representations for both current and historical utterances, while *MR* encoders build neural representations based on on historical *MRs*.

Utterance encoders aim to embed rich information hidden in utterances into fixed-length representations, which provide contextual information in addition to current utterances for decoders. They apply first an RNN to map each utterance into a continuous vector of fixed-size. Then there are three ways to encode utterances in context into a fixed-size neural representation.

- For each utterance in a dialogue, a straightforward method is to concatenate its previous $k-1$ utterances with current utterance in order and encode them with the RNN (Suhr et al., 2018; Suhr and Artzi, 2018). As a result, decoders have access to information in at most k utterances. However, this method fails to access information beyond the k utterances. In addition, it is computationally expensive because if an utterance belongs to multiple contexts, it would be repeatedly encoded for modelling all the contexts.

- To overcome the above weakness, an alternative method is to treat a sub-sequence of utterances up to time t as a sequence of vectors, and project them into a *discourse state* vector by using a turn-level RNN (Suhr et al., 2018; Zhang et al., 2019; He et al., 2019). In another word, those models apply hierarchical RNNs to map each context into a fixed-size vector. In this method, each utterance is encoded only once and reused for modelling different contexts. However, this approach often leads to significant information loss (Pascanu et al., 2013; Khandelwal et al., 2018) due to the challenges imposed by encoding sequences of utterances into single vectors.

- In order to focus on history utterances most relevant to current decoder states or utterances, soft attention (Bahdanau et al., 2014) is applied to construct context vectors. The query vectors are either the hidden state of an decoder (Suhr et al., 2018; He et al., 2019; Suhr and Artzi, 2018) or an utterance vector (Liu et al., 2020; Zhang et al., 2019). To differentiate between positional information, token embeddings of history utterance are concatenated with their position embeddings (Suhr et al., 2018; He et al., 2019), which encode the positions of history utterances relative to the current utterances. This method reflects the observation that similar utterances tend to share relevant information, such as references of the same entities. Both discourse states and attended representations are also widely used by the neural dialogue models (Zhang et al., 2018), thus suffer from the same problems caused by composition complexity. As a result, the trained models are found insensitive to utterance order and word order in context (Sankar et al., 2019).

MR encoders construct a neural context representation at time t based on the *MRs* predicted before t. As *MRs* are expressed in a formal language, *MR* encoders also apply RNNs to encode each *MR* or segments of *MRs* into embedding vectors. Then *MR* encoders build context representations of historical *MRs* in the same spirit as utterance encoders. In (Guu et al., 2017), they only concatenate the embeddings of k most recent history *MR* tokens as they assume current *MR* is always an extension of previous *MRs*. In (Suhr et al., 2018), a bidirectional RNN is applied to construct a vector for each segment, which is extracted from historical *MRs*. Soft attention is also applied in (Zhang et al., 2019) for building context vectors, which uses the current hidden state of their decoder as the query vector to attend over the token embeddings of the previous *MR*.

Context-aware Decoders Decoders in CDSP models produce *MRs* based on the neural representations provided by their encoders. Such a decoder yields an *MR* by generating a sequence of *MR* tokens according to model distribution $P(\mathbf{y}|\mathbf{x}, \mathbf{C})$, where $\mathbf{C}$ denotes context information. There are three major ways to utilize context information.

One key problem of CDSP is incomplete information in current utterances. The straightforward way is to take neural context representations $\mathbf{C}$ as additional input of decoders, which are yielded by context-aware encoders. Those context representations contains information from previous utterances, historical *MRs*, or both. The decoders take them as input by concatenating them with the ones from current utterances at each decoding step (Suhr et al., 2018; Liu et al., 2020; Chen and Bunescu, 2019; Zhang et al., 2019; Shen et al., 2019). Thus, the quality of decoding depends tightly on the quality of contextual encoding, which is still a challenging problem (Sankar et al., 2019).

MRs of current utterances often contain segments from previous *MRs* (Suhr et al., 2018). The shared parts are references to previously mentioned entities or constraints implied by context. Reuse of *MR* segments is realized by a designated *copy* component, which selects a segment to copy when the probability of copying is high. As decoders in SEQ2SEQ produce a sequence of decisions for each input, the corresponding model generates a sequence of mixed decisions, including both *copy* of segments and generation of new *MR* tokens. In a similar manner, copying of *MR* tokens from previous *MR* is proposed in (Zhang et al., 2019).

Coreference resolution is explicitly addressed in Chen and Bunescu (2019). As illustrated by the example in Figure 1, a special token *REF* is introduced in the output vocabulary for denoting if an entity in the preceding *MR* is referred in that utterance. If that is the case, the corresponding entity token is copied from the previous *MR* to replace the *REF* token via a pointer network module (Vinyals et al., 2015).

3.3 Neural-Symbolic Approaches

Neural-symbolic approaches introduce grammar into the decoding process or utilize symbolic representations as intermediate representations, while applying neural nets for representation learning. They take advantages from both the good context representation obtained by neural nets and reduced complexity of decoding due to the constraints introduced by grammars. In existing work, those approaches regard the generation of an *MR* as the prediction of a sequence of actions. Neural-symbolic methods normally take the same methods as the neural approaches to encode the contextual information. What differentiate them is the neural-symbolic could handle context by i) designing specific actions, and ii) utilizing symbolic context representations.

The context specific actions proposed in (Iyyer et al., 2017; Sun et al., 2019; Liu et al., 2020) adopt *copy* mechanism to reuse the previous *MRs*. CAMP (Sun et al., 2019) include three actions to copy three different SQL clauses from precedent queries. Liu et al. (2020) allows copying of any actions or subtrees from precedent SQL queries. The *subsequent* action in (Iyyer et al., 2017) adds SQL conditions from the previous query into the current semantic parse to address the ellipsis problem. Different from other approaches, Iyyer et al. (2017) uses a DYNSP, which is in a similar neural network structure as the DNMN, instead of the SEQ2SEQ to generate the action sequences.

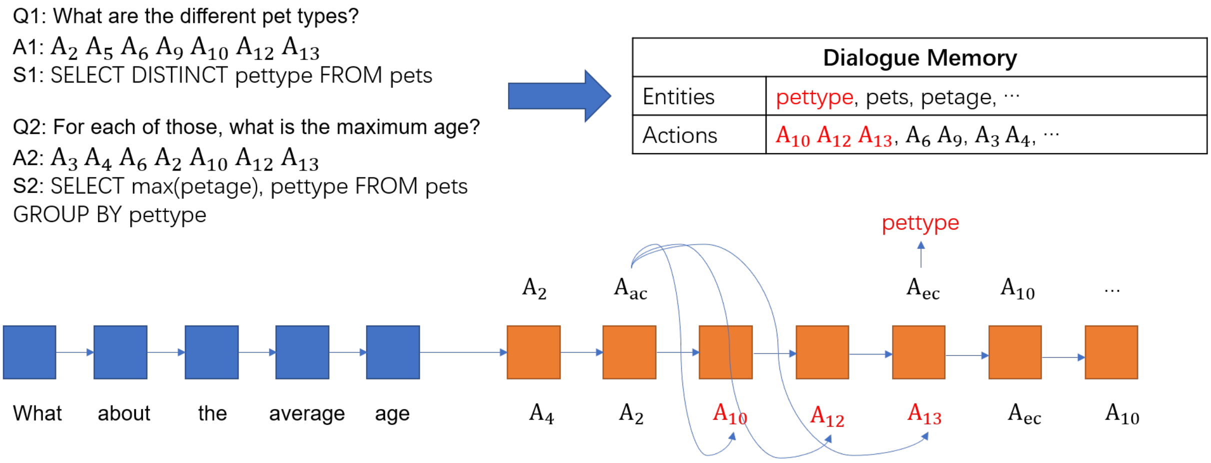
Q1: What are the different pet types?
A1: A2 A5 A6 A9 A10 A12 A13
S1: SELECT DISTINCT pettype FROM pets
Q2: For each of those, what is the maximum age?
A2: A3 A4 A6 A2 A10 A12 A13
S2: SELECT max(petage), pettype FROM pets GROUP BY pettype
Dialogue Memory
Entities
pettype, pets, petage, ⋯
Actions
A10 A12 A13, A6 A9, A3 A4, ⋯
pettype
A2
Aac
Aec
A10
⋯
What
about
the
average
age
A4
A2
A10
A12
A13
Aec
A10

Figure 2: The symbolic memory architecture of (Guo et al., 2018). Considering the example in Table 1, Guo et al. (2018) defines different types of actions, A_{ac} and A_{ec}, to copy action sequence A_{10}, A_{12}, A_{13} and the entity *pettype* from the symbolic memory, respectively.

Production rules are also used to explicitly address the coreference resolution. In (Shen et al., 2019), the authors defined fours actions to instantiate the entities, predicates, types and numbers. Then the pointer network is utilized to find mentions of the four entry semantic categories in the current and history utterances. The entities in utterances are later mapped to entities in knowledge bases by using their entity linking tool.

Instead of directly copying from previous *MRs*, the parser DIALOG2ACTION (Guo et al., 2018) incorporates a dialogue memory, which maintains *symbolic* representations of entities, predicates and action subsequences from an interaction history (Figure 2). That parser defines three types of designated actions to copy entities, predicates and action subsequences from the memory respectively. Instead of decisively copying from memory, each type of action probabilistically selects the corresponding segments conditioning on the *symbolic* representations, which are later integrated into the generated action sequences.

Guo et al. (2019a) employs the same neural-symbolic models as in (Guo et al., 2018) to capture contextual information. Different from other approaches, Guo et al. (2019a) adopts the meta-learning approach to improve the generation ability of CDSP models. Inspired by (Huang et al., 2018), Guo et al. (2019a) utilize the context from other interactions to guide the learning of CDSP over utterances within current interactions via MAML. Guo et al. (2019a) considers an input utterance x_i and its context C_i as an instance. A context-aware retriever would retrieve instances, which are semantically close to the current instances, from other interactions. When learning model parameters, the retrieved instances and the current instances are considered as the support set and test set, respectively, and grouped as tasks as in the common MAML paradigm.

3.4 Comparison between Different CDSP Approaches

In (Liu et al., 2020), 13 different context modeling methods for both neural and neural-symbolic CDSP parsers were evaluated on two benchmark datasets. None of those methods achieve consistent superior results over the others in all experimental settings. Among them, concatenation of k recent utterances for decoders and copy of parse actions from precedent *MRs* are the top performing ones in most settings. Liu et al. (2020) defines 12 fined-grained types summarized with multiple hierarchies according to the contextual linguistic phenomena, and inspects how different linguistic phenomena influence the model behavior. One interesting conclusion is that the methods in their experiments all perform poorly on the instances involving coreference problems that require complex inference. But note that, those methods in that study were not compared with the ones with explicit coreference resolution. Another interesting finding is that all the models perform better on the utterances which only augment the semantics of previous sentences than on the utterances which substitute the partial semantics of the precedent utterances.

3.5 Comparison between CDSP and Feedback Semantic Parsing

Feedback/Interactive Semantic Parsing is another line of research in semantic parsing that utilizes context to refine *MRs* in an iterative manner. Most Feedback Semantic Parsing systems (Iyer et al., 2017; Yao et al., 2019b; Yao et al., 2019a; Elgohary et al., 2020) start with using an CiSP parser to parse a given utterance into an initial *MR*. Then the *MR* is interpreted in natural language and sent to a user. The user provides feedback, based on which the systems revise the initial parse. The process repeats till convergence. Therefore, in Feedback Semantic Parsing, interaction histories are only used to revise the parses. In contrast, CDSP focuses on modelling the dependencies between the utterances. Elgohary et al. (2020) empirically compares CDSP with Feedback Semantic Parsing. They train a CDSP model, EditSQL (Zhang et al., 2019), on two CDSP datasets, SParC and CoSQL, and evaluate it on the test set of a feedback semantic parsing dataset, SPLASH. The performance is merely 3.4% and 3.2% in terms of accuracy, indicating that the two tasks are distinct by addressing different aspects of context.

4 Datasets and Resources

Datasets	Reference	#Party	Annotation	MR Language	#Utterance	#Interaction	Avg. #Turns
ATIS	(Price, 1990)	1	MR	SQL/lambda	11,653	1,658	7.0
SequentialQA	(Iyyer et al., 2017)	1	Denotation	self	17,553	6,066	2.9
SParC	(Yu et al., 2019b)	1	MR	SQL	12,726	4,298	3.0
TimeExpression	(Lee et al., 2014)	1	Denotation	lambda	NA	298	NA
TempStructure	(Chen and Bunescu, 2019)	1	MR	self	1,237	NA	NA
SCONE	(Long et al., 2016)	1	Denotation	self	69,755	13,951	5.0
CSQA	(Saha et al., 2018)	1	Denotation	SPARQL/self	˜1.6M	˜200,000	˜10.0
SpaceBook	(Vlachos and Clark, 2014)	2	MR, act	self	2,374	17	139.7
EmailDialogue	(Srivastava et al., 2017)	2	MR	Lisp	4,759	113	42.0
CoSQL	(Yu et al., 2019a)	2	MR, act	SQL	15,598	3,007	5.2

Table 2: The statistics of the context dependent datasets. ”#” denotes the number of the corresponding units (e.g. number of utterances, number of interactions, etc.). ”˜” denotes this is an estimated number. ”NA” denotes that the corresponding statistic data is not applicable. ”self” denotes the target languages in the datasets are only applicable to a small range of datasets.

Table 2 summarizes the basic properties and statistics of existing CDSP datasets. There are two scenarios of the CDSP datasets, Single-party Scenarios and Multi-party Scenarios. In the former scenarios, the user utterances are translated into *MRs* to obtain the execution results from the programming environment. In the latter scenarios, there are systems which respond to the users in natural language based on the user utterances and the execution results. The user utterances are manually labeled with different types of annotations, including *MRs*, denotations, and dialogue acts. The system responses are usually annotated with the dialogue acts. We especially highlight those annotations that explicitly reflect contextual dependencies of utterances in the sequel.

4.1 Scenarios

Single-party Scenarios In SParC (Yu et al., 2019b), SequentialQA (Iyyer et al., 2017) and ATIS, the user utterances within each interaction are around a topic described by the provided text. To collect SParC and SequentialQA, crowd-workers are asked to raise questions to obtain the information that answers the questions sampled from other corpora (Pasupat and Liang, 2015; Yu et al., 2018). But the assumption for SequentialQA is the answers of the current question must be the subset of answers from the last turn. In ATIS (Price, 1990), crowd-workers raise questions around the detailed scripts describing air travel planning scenarios.

TempStructure (Chen and Bunescu, 2019) and TimeExpression (Lee et al., 2014) particularly focused on addressing the temporal-related dependency. In TempStructure, human users or the simulators raise natural language questions chronologically towards a knowledge base. The facts in the knowledge base are organized in time series. Therefore, the questions in TempStructure are rich with time expressions. TimeExpression only annotate temporal mentions (text segments that describe time expressions) instead of complete questions. All the mentions are from the time expression-rich corpora.

SCONE (Long et al., 2016) and CSQA (Saha et al., 2018) use semi-automatic approaches to simulate the contextual dependency. Each interaction in SCONE is merely labeled with an initial denotation and an end denotation. The denotations in SCONE are regarded as the states that can be manipulated by the programs. Within each interaction, multiple candidate sequences of programs would be automatically generated while only the sequence of programs, which could correctly transit the initial state to the end state, would be kept and described with natural language by the crowd-workers. To create CSQA (Saha et al., 2018) dataset, the crowd-workers are asked to raise questions that can be answered from single fact tuples (e.g. relation: *CEO*, subject: *Google*, object: *Sundar Pichai*) in the knowledge graph or the complex facts which are the composition of multiple tuples. To create coherent dependency among questions, the questions that share the relations or entities are placed next to each other. And crowd-workers would manually modify the questions such that the sequence of questions would include contextual linguistic properties such as ambiguity, underspecification or coreference. It is worth mentioning that, with such method, CSQA includes the largest number of interactions until now, which is over 200k.

Multi-party Scenarios Similar to the scenario of SPARC, to obtain the answers to the questions sampled from SPIDER (Yu et al., 2018), the conversations in COSQL (Yu et al., 2019a) are conducted between two human interlocutors, who play the roles of user and system, respectively. The dialogues in the SPACEBOOK (Vlachos and Clark, 2014) are under the scenarios formed by the routing requests. One human interlocutor pretends to be a tourist walking around Edinburgh while another interlocutor plays the role of a system responding to the tourist. The conversations in EMAILDIALOGUE are between the human agent and an email assistant instead of two humans.

4.2 Context and Annotations

The contextual linguistic phenomena in the CDSP corpora is quite close to the phenomena in the corpora of tasks such as document-level machine translation, question answering, dialogue system, etc.. However, in CDSP datasets, the contextual linguistic phenomena has a tight relation with the annotations.

Iyyer et al. (2017), Vlachos and Clark (2014) defined specific components in the *MR* languages of SEQUENTIALQA and SPACEBOOK to explicitly model the context dependency. SEQUENTIALQA introduced a keyword *subsequent*. All the answers of *MR* statements after *subsequent* would only be the subset of the answers of the precedent *MR*. In the language of SPACEBOOK, to resolve the coreference problem, a special predicate *equivalent* could indicate the identical entities across questions at different turns.

The context dependency could be reflected by some properties of annotations. Yu et al. (2019b) analyzed semantic changes over turns in SPARC by calculating the overlapping percentage of tokens between the SQL annotations at different turns. In SPARC, the average overlapping percentage increases at later turns within one interaction, where the users tend to narrow down their topics with turns increasing. Both Yu et al. (2019b) and Liu et al. (2020) categorized the contextual phenomena in SPARC into fine-grained types and calculate their frequency. Yu et al. (2019b) found some SQL representations correspond to certain contextual phenomena types. For instance, in the questions of the *theme-entity*, which means the current question and precedent question are around the same entities but request for different properties, their corresponding SQL representations have the same *FROM* and *WHERE* clauses. But the SQL representations for other types may vary.

For the datasets, SPACEBOOK and COSQL, Yu et al. (2019a) and Vlachos and Clark (2014) label utterances with dialogue acts along with *MRs*. Different from (Yu et al., 2019a), Vlachos and Clark (2014) integrated the dialogue acts into the *MRs*. The dialogue acts can be considered as the overall functions of the utterances while different dialogue acts reflect different properties of utterances. For example, in COSQL, the unanswerable questions that can not be parsed into SQLs are labelled with dialogue acts such as *NOT_RELATED*, *CANNOT_UNDERSTAND*, or *CANNOT_ANSWER*. The ambiguous questions that need to be clarified are labelled with *AMBIGUOUS*. The following questions are then labeled with *CLARIFY*. The dialogue acts could provide additional contextual information for CDSP.

5 Challenges and Future Directions

CDSP distinct from CISP by context modelling and utilization of context information in the parsing process to complete missing information in *MRs*. Despite significant progress in recent years, there are still multiple directions worth pursuing.

Analysis of Linguistic Phenomena Benefiting from Context Yu et al. (2018) and Liu et al. (2020) analyzed the influence of different types of contextual information on CDSP methods. Despite some empirical results, it still lacks of a thorough understanding of pros and cons of each type of context in relation to the parsing task. For example, in which cases should parsers extract information from *MRs* in context instead of utterances? Apart from ellipsis and coreference resolution, are there other linguistically motivated problems in context the current parsers have not addressed yet?

Incorporating Far-side Pragmatics Current CDSP approaches fall in the scope of near-side pragmatics, in particular reference resolution, and current CDSP datasets (e.g. COSQL, SPACEBOOK) consider dialogue acts as merely the overall function of the utterances (Vlachos and Clark, 2014). However, *far-side pragmatics* focuses on what happens beyond saying, including implicatures and communicative intentions etc.. Incorporating far-side pragmatics in semantic parsing will be especially useful towards completely understanding dialogues. Thus, there is a need to create large corpora annotated with rich information about various aspects of pragmatics for both training and evaluation.

Causal Structure Discovery in Context A key challenge of context based modelling is composition complexity caused by highly varying context. The empirical results in (Liu et al., 2020) show that the SOTA models can capture well nearby context information but it is still challenging to capture long-range dependencies in context. One possible direction is to find out the underlying causal structure (Glymour et al., 2014), which should be sparse and explains well which contextual information leads to current utterances. If we can focus only on the key reasons in context that lead to changes of *MRs*, the influence from noisy information and overfitting of models is expected to decrease significantly. Another potential benefit of understanding causal structures in context is to improve robustness of parsers by ignoring non-robust features (Zhang et al., 2020; Ilyas et al., 2019).

Low-resource CDSP Since most CDSP datasets are small in terms of the number of utterances and interactions, the direction on addressing the low-resource problem in CDSP is quite promising. The meta-learning approaches, such as the MAML CDSP in (Guo et al., 2019a), could be a potential direction to address this issue. The other typical methods to solve low-resource issues, including weakly supervision, data augmentation, semi-supervised learning, self-supervised learning etc., could be further investigated in the scenarios of CDSP.

Acknowledgements

We thank Xuanli He and anonymous reviewers for their useful suggestions.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*.

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

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*, pages 178–186.

Charles Chen and Razvan Bunescu. 2019. Context-dependent semantic parsing over temporally structured data. *arXiv preprint arXiv:1905.00245*.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*.

Ahmed Elgohary, Saghar Hosseini, and Ahmed Hassan Awadallah. 2020. Speak to your parser: Interactive text-to-sql with natural language feedback. *arXiv preprint arXiv:2005.02539*.

Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

Clark Glymour, Richard Scheines, and Peter Spirtes. 2014. *Discovering causal structure: Artificial intelligence, philosophy of science, and statistical modeling*. Academic Press.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

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*, pages 2942–2951.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2019a. Coupling retrieval and meta-learning for context-dependent semantic parsing. *arXiv preprint arXiv:1906.07108*.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019b. Towards complex text-to-sql in cross-domain database with intermediate representation. *arXiv preprint arXiv:1905.08205*.

Kelvin Guu, Panupong Pasupat, Evan Zheran Liu, and Percy Liang. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *arXiv preprint arXiv:1704.07926*.

Xuanli He, Quan Hung Tran, and Gholamreza Haffari. 2019. A pointer network architecture for context-dependent semantic parsing. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 94–99.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. *arXiv preprint arXiv:1803.02400*.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. *arXiv preprint arXiv:1704.08760*.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2018. Mapping language to code in programmatic context. *arXiv preprint arXiv:1808.09588*.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.

Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765. Association for Computational Linguistics.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D Forbus, and Ni Lao. 2016. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *arXiv preprint arXiv:1611.00020*.

Percy Liang. 2016. Learning executable semantic parsers for natural language understanding. *Communications of the ACM*, 59(9):68–76.

Qian Liu, Bei Chen, Jiaqi Guo, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. How far are we from effective context modeling? an exploratory study on semantic parsing in context. *arXiv preprint arXiv:2002.00652*.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. *arXiv preprint arXiv:1606.05378*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Constituency Parsing. 2009. Speech and language processing.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. 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 1470–1480, Beijing, China, July. Association for Computational Linguistics.

Patti J Price. 1990. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

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

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. *arXiv preprint arXiv:1906.01603*.

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. *arXiv preprint arXiv:1910.05069*.

Shashank Srivastava, Amos Azaria, and Tom M Mitchell. 2017. Parsing natural language conversations using contextual cues. In *IJCAI*, pages 4089–4095.

Alane Suhr and Yoav Artzi. 2018. Situated mapping of sequential instructions to actions with single-step reward observation. *arXiv preprint arXiv:1805.10209*.

Alane Suhr, Srinivasan Iyer, and Yoav Artzi. 2018. Learning to map context-dependent sentences to executable formal queries. *arXiv preprint arXiv:1804.06868*.

Yibo Sun, Duyu Tang, Jingjing Xu, Nan Duan, Xiaocheng Feng, Bing Qin, Ting Liu, and Ming Zhou. 2019. Knowledge-aware conversational semantic parsing over web tables. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 827–839. Springer.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

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

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700.

Andreas Vlachos and Stephen Clark. 2014. A new corpus and imitation learning framework for context-dependent semantic parsing. *Transactions of the Association for Computational Linguistics*, 2:547–560.

Ziyu Yao, Xiujun Li, Jianfeng Gao, Brian Sadler, and Huan Sun. 2019a. Interactive semantic parsing for if-then recipes via hierarchical reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2547–2554.

Ziyu Yao, Yu Su, Huan Sun, and Wen-tau Yih. 2019b. Model-based interactive semantic parsing: A unified framework and a text-to-sql case study. *arXiv preprint arXiv:1910.05389*.

Pengcheng Yin and Graham Neubig. 2018. Tranx: A transition-based neural abstract syntax parser for semantic parsing and code generation. *arXiv preprint arXiv:1810.02720*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.

Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019a. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. *arXiv preprint arXiv:1909.05378*.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, et al. 2019b. Sparc: Cross-domain semantic parsing in context. *arXiv preprint arXiv:1906.02285*.

Luke S Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 976–984. Association for Computational Linguistics.

Luke S Zettlemoyer and Michael Collins. 2012. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*.

Weinan Zhang, Yiming Cui, Yifa Wang, Qingfu Zhu, Lingzhi Li, Lianqiang Zhou, and Ting Liu. 2018. Context-sensitive generation of open-domain conversational responses. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2437–2447.

Rui Zhang, Tao Yu, He Yang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. Editing-based sql query generation for cross-domain context-dependent questions. *arXiv preprint arXiv:1909.00786*.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Kai Zhao and Liang Huang. 2014. Type-driven incremental semantic parsing with polymorphism. *arXiv preprint arXiv:1411.5379*.

Q. Zhu, X. Ma, and X. Li. 2019. Statistical learning for semantic parsing: A survey. *Big Data Mining and Analytics*, 2(4):217–239.