# SCoRe: Pre-Training for Context Representation in Conversational Semantic Parsing

**Tao Yu**
Yale University
tao.yu@yale.edu

**Rui Zhang**
The Pennsylvania State University
rmz5227@psu.edu

**Oleksandr Polozov, Christopher Meek, Ahmed Hassan Awadallah**
Microsoft Research
{polozov,meek,hassanam}@microsoft.com

## Abstract

Conversational Semantic Parsing (CSP) is the task of converting a sequence of natural language queries to formal language (e.g., SQL, SPARQL) that can be executed against a structured ontology (e.g. databases, knowledge bases). To accomplish this task, a CSP system needs to model the relation between the unstructured language utterance and the structured ontology while representing the multi-turn dynamics of the dialog. Pre-trained language models (LMs) are the state-of-the-art for various natural language processing tasks. However, existing pre-trained LMs that use language modeling training objectives over free-form text have limited ability to represent natural language references to contextual structural data. In this work, we present SCoRe, a new pre-training approach for CSP tasks designed to induce representations that capture the alignment between the dialogue flow and the structural context. We demonstrate the broad applicability of SCoRe to CSP tasks by combining SCoRe with strong base systems on four different tasks (SParC, CoSQL, MWoZ, and SQA). We show that SCoRe can improve the performance over all these base systems by a significant margin and achieves state-of-the-art results on three of them.

## 1 Introduction

The goal of task-oriented dialog systems is to assist the user in completing a certain task by performing an action or retrieving relevant information (Tur & Mori, 2011). They are often built on top of a structured ontology grounded in a knowledge base, a database, or a set of API calls. This in contrast to open-domain dialog systems (also referred to as chit-chat systems) where the goal is to maximize engagement with users in open-ended conversations (Jafarpour et al., 2010; Ritter et al., 2011).

A key component of task-oriented conversational systems is Conversational Semantic Parsing (CSP), which converts each utterance in the dialog into a formal language query (e.g., SQL, SPARQL) that can be executed against the structured ontology. CSP has been extensively studied in several academic and industrial research settings such as dialog systems (e.g., dialog state tracking in MWoZ (Budzianowski et al., 2018)), interacting with physical agents (e.g., (Chai et al., 2018)), context-dependent semantic parsing (e.g., SParC (Yu et al., 2019b)), SQL-grounded state tracking (e.g., CoSQL (Yu et al., 2019a)), and sequential question answering (e.g., SQA (Iyyer et al., 2017)). These settings differ in some respect, but they share the same overall objective and key challenge: *how to jointly represent the natural language utterances and underlying structured ontology while taking into consideration the multi-turn dynamics of the dialog*.

Similar to many other natural language tasks, recent work in CSP has significantly benefited from advances in language model pre-training. However, existing general-purpose pre-trained language models, e.g. BERT (Devlin et al., 2019), are pre-trained on free-form text data using language model objectives. This limits their ability in modeling the structural context or the multi-turn dynamics of the dialogs. This presents an opportunity to improve pre-trained LMs to specifically address these limitations for CSP tasks. Recent work has demonstrated the benefits of adapting pre-trained LMs

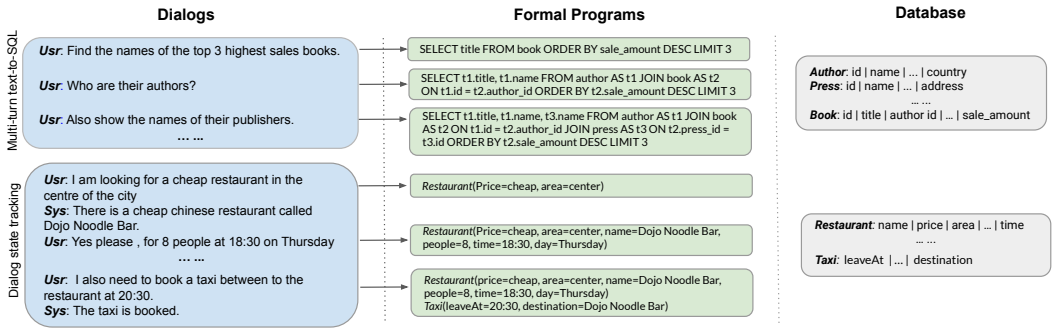


Figure 1: Examples of conversational semantic parsing tasks from SPARC and MWOZ datasets.

to specific domains (Gururangan et al., 2020) or tasks (Zhang et al., 2019b) via a second phase of pre-training. For example, open-domain dialogue language models such as DialoGPT (Zhang et al., 2020) and ConveRT (Henderson et al., 2019) are pre-trained on the Reddit data and applied to dialog response generation and retrieval tasks.

In this paper, we introduce SCORE (**S**tructured & **S**equential **Co**ntext **Re**presentation), a language model pre-training approach for CSP tasks. SCORE adapts general pre-trained LMs by introducing a second phase of pre-training using multiple objectives that capture both multi-turn dynamics and the structural contexts in a dialog. In contrast to open-domain dialogs, CSP datasets are usually much smaller due to the difficulty and expense of obtaining and labeling data (mapping natural language utterances to formal language). Unlike most prior work on contextualized LMs which are pre-trained on free text, according to the finding where questions in CSP tasks are more compositional than other free-text since they can be mapped into formal representations, we propose to train SCORE on synthesized conversational semantic parsing data with multiple training objectives that aim to ground utterances into the schema of the underlying ontology and to model the relationship between different utterances in the multi-turn conversation. In this way, SCORE can effectively inject structural and conversational inductive biases in LMs that can translate to many CSP tasks. SCORE uses an order of magnitude smaller dataset for the second stage of pre-training, does not require changes to the pre-trained model architecture, can be used as a drop-in replacement of general pre-trained LMs with any semantic parsing model, and can be used out-of-the-box in many CSP tasks.

We apply SCORE to four different CSP tasks: (1) sequential text-to-SQL (SPARC), (2) conversational text-to-SQL (COSQL), (3) dialog state tracking (MWOZ), and (4) weakly-supervised sequential question answering (SQA). The fours tasks represent different scenarios, types of ontologies, supervision signals, system responses, and domains (see Table 1 for a detailed comparison and Figure 1 for examples). We demonstrate that: (1) SCORE training objectives can effectively incorporate synthesized data, (2) a single pre-trained SCORE model can be used for several CSP tasks and can be combined with many baseline systems with different model architectures and (3) SCORE significantly improve all baseline systems and achieves new state-of-the-art results on three benchmarks (SPARC, SPARC, and MWOZ) and comparable performance to state-of-the-art results on the fourth (SQA).

## 2 APPROACH

The key challenge of CSP is to capture the relationship between the natural language utterance and the structured ontology in the multi-turn dialog dynamics. To this end, we inject structural and conversational inductive biases in SCORE by introducing two objective functions: *Column Contextual Semantics (CCS)* and the *Turn Contextual Switch (TCS)*. Because the size of existing semantic parsing datasets is limited, we produce synthesized data for pretraining SCORE by sampling from the context-free grammar induced from complex text-to-SQL examples in different domains. Moreover, to prevent SCORE from overfitting to the linguistic pattern of our synthesized data, we use the *Masked Language Modeling (MLM)* objective on human-generated utterances as regularization.

### 2.1 PRELIMINARIES

**Task Definition** In CSP, at each turn $t$, we aim to produce a formal representation $q_t$ given the current utterance $u_t$, the interaction history $h_t = [u_1, u_2, \ldots, u_{t-1}]$, and the schema $c$ (table and column names, slots, etc.) of the target database (ontology) $d$. To cover different problem variants, we

| Dataset | Structured Ontology | Annotation (Supervision) | Cross Domain | System Response | # Dialogs | # Turns |
|---|---|---|---|---|---|---|
| SPARC | database | SQL (supervised) | ✓ | ✗ | 4,298 | 12,726 |
| COSQL | database | SQL (supervised) | ✓ | ✓ | 3,007 | 15,598 |
| MWOZ | domain ontology | slot-value (supervised) | ✗ | ✓ | 8,438 | 113,556 |
| SQA | table | denotation (weakly-supervised) | ✓ | ✗ | 6,066 | 17,553 |

Table 1: Comparison of CSP datasets. Examples from two of the datasets are shown in Figure 1. Cross-domain means the train and test sets have different domains, so MWOZ is not cross-domain.

consider four popular CSP tasks shown in Table 1: SPARC (sequential text-to-SQL), COSQL (conversational text-to-SQL), MWOZ (dialogue state tracking), and SQA (weakly supervised sequential question answering). They have different target formal language and structured ontology:

- For the **utterance** $\boldsymbol{u}$, it is the user question for SPARC and SQA, while for COSQL and MWOZ, $u$ is the combination of a user query and a system response.
- For the **database** $\boldsymbol{d}$, SPARC and COSQL use multi-table databases; for MWOZ, the pre-defined ontology $d$ can also be viewed as a database; for SQA, $d$ is a single table.
- For the **formal representation** $\boldsymbol{q}$, it is the SQL query for SPARC and COSQL; in MWOZ it is the slot-value pairs that can be viewed as simple SQL queries consisting of `SELECT` and `WHERE` clauses; and for SQA, $q$ is the latent program.

**Base Architecture** The base architecture of SCORE takes as input a single turn of a CSP dialog $\langle u_t, h_t \rangle$ jointly with the underlying database schema $c$. Given this *contextualized conversational input* $C_t = \langle u_t, h_t, c \rangle$, SCORE encodes it into *contextualized conversation representations* $\vec{\boldsymbol{S}}_t$ for each token in $C_t$. The encoder architecture follows ROBERTA (Liu et al., 2019b). It is then followed by a linear layer and normalized (Ba et al., 2016) to produce final representations $\vec{\boldsymbol{h}}_t$ for each token:

$$C_t = \langle u_t, h_t, c \rangle,\ \vec{\boldsymbol{S}}_t = \text{ROBERTA}(C_t),\ \boldsymbol{h}_{t,i} = \text{LayerNorm}(\text{GELU}(\boldsymbol{W_1}\boldsymbol{S}_{t,i}))\ \forall\, \boldsymbol{S}_{t,i} \in \vec{\boldsymbol{S}}_t, \quad (1)$$

where GELU is an activation by Hendrycks & Gimpel (2016) and $\boldsymbol{W_1}$ is a learned parameter matrix.

To build $C_t$, we first concatenate current utterances $u_t$ and dialog history $h_t$ separated by a special token `<s>`, as this simple strategy has been shown effective in state-of-the-art CSP systems (Zhang et al., 2019c; Wu et al., 2019; Liu et al., 2020; Heck et al., 2020). To incorporate the database schema, we follow Hwang et al. (2019) to concatenate all column names as a single sequence. Column names are separated by the special token `</s>` and prefixed by their corresponding table name.

### 2.2 SCORE PRE-TRAINING

SCORE addresses the challenges of CSP by *pre-training a task-oriented language model contextualized by the conversational flow and the underlying ontology*. In pre-training, the SCORE model is self-supervised by two novel objectives in addition to the established Masked Language Modeling (MLM) objective. These objectives facilitate the accurate representation of the conversational flow between dialog turns and how this flow maps to the desired columns in the ontology.

**Column Contextual Semantics** The first challenge of CSP is capturing the alignment between the natural language utterance and the underlying database schema. To address it, we optimize the SCORE model with the auxiliary objective of *Column Contextual Semantics (CCS)*. For each column in the database schema $c$, CCS targets the *operations* that should be performed on this column in a given conversational turn. Specifically, each formal representation $q$ is decomposed into operations on columns and tables, e.g. `GROUP BY` and `HAVING` for SQL queries, or `WHERE` for the slot-value pairs. In this way, our data covers 148 column operations. We use the encoding of the special token `</s>` right before each column or table name to predict its corresponding operations, and then compute the CCS loss:

$$\mathcal{L}_{\text{CCS}}(C_t) = \sum\nolimits_{i \in c} \text{CrossEntropy}_{148}(\text{LayerNorm}(\boldsymbol{W}_2\, \boldsymbol{h}^c_{t,i}), \text{CCS}(q_t)) \quad (2)$$

where $\boldsymbol{h}^c_{t,i}$ is the contextualized representation of the $i^{\text{th}}$ column's special token `</s>` in the contextualized input $C_t$, $\text{CCS}(q_t)$ returns the column operation label for the current formal representation $q_t$, $\text{CrossEntropy}_{148}$ computes the 148-way cross-entropy between the column operation prediction and label, and $\boldsymbol{W}_2$ is a learned parameter matrix.

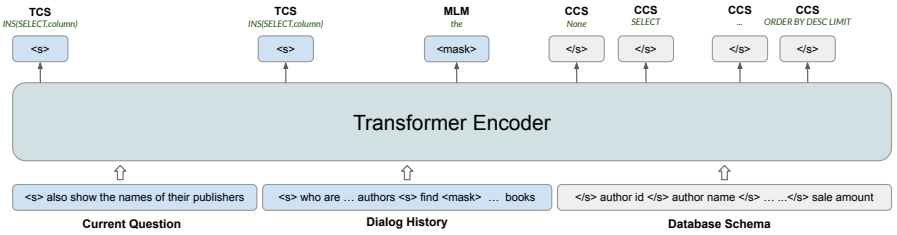


Figure 2: Pre-training of a SCORE encoder on a SPARC text-to-SQL example from Figure 1.

**Turn Contextual Switch** The second challenge of CSP is capturing the conversational context flow and how it is grounded into the formal representations. The TCS objective aims to capture this grounding of context flow. To this end, it targets predicting *the difference in formal representations between dialog turns* based on the natural language utterance.

Based on the context-free grammar of SQL, we identify 26 possible *turn difference operations* that a conversational turn could elicit. They encode changes between different turns of user queries (the system response is not involved here) since we assume that most turn contextual shifts are from the user. For example, `INS(WHERE)` indicates inserting a new `WHERE` condition and `DEL(SELECT.agg)` indicates removing an aggregate operation from a `SELECT` statement (e.g. when an utterance *"Show all the ages instead."* elicits a change `SELECT MAX(age) ...` $\rightarrow$ `SELECT age ...`). We use the encoding of the special token `</s>` right before each turn to predict the context switch label between this turn and the previous history:

$$\mathcal{L}_{\text{TCS}}(C_t) = \text{CrossEntropy}_{26}(\text{LayerNorm}(\boldsymbol{W}_3 \boldsymbol{H}_t^s), \text{TCS}(q_t, q_{t-1})) \tag{3}$$

where $\boldsymbol{H}_t^s \in \mathbb{R}^{(t-1)\times d}$ is the contextualized representation of all previous turns in $C_t$ with hidden dimension $d$, $\text{TCS}(q_t, q_{t-1})$ returns the turn difference operations from $q_{t-1}$ to $q_t$, and $\boldsymbol{W}_3$ is a learned parameter matrix. We don't use this objective to pre-train SCORE for MWOZ because the context switch label between turns is relatively simple in MWOZ (only `select` and `where` changes).

**Masked Language Modeling** As in prior work on large-scale language models (Devlin et al., 2019), we use the *Masked Language Modeling (MLM)* objective to facilitate contextual representation learning for natural language utterances. Importantly for regularization, we only apply this loss on *in-domain human-annotated* natural language data. Namely, it includes utterances in SPARC, COSQL, and SQA as well as nine task-oriented dialog datasets processed by Wu et al. (2020) for MWOZ (see data statistics in Figure 4). Formally, the MLM loss is given by:

$$\mathcal{L}_{\text{MLM}}(C_t) = \sum_m \text{CrossEntropy}_{\text{Vocab}}(\text{LayerNorm}(\boldsymbol{W}_4 \boldsymbol{h}_t^m)) \tag{4}$$

where $\boldsymbol{h}_t^m$ are the contextualized representations of the masked 15% of tokens in $C_t$, and $\boldsymbol{W}_4$ is a learned parameter matrix.

**Pre-Training Setup and Steps** To summarize the pre-training steps, we first collect a dataset $\mathcal{D}_{\text{nat}}$ of combined human-annotated natural language questions (without labels) from existing CSP tasks (as mentioned above), and create a large synthesized conversational data $\mathcal{D}_{\text{syn}}$ that is generated by a grammar induced from a small set of SPARC annotated examples (See 2.3). After that, we incorporate both two datasets in pre-training. More specifically, synthetic and natural examples are randomly sampled during pre-training. The total pre-training loss is the sum of the three objectives with CCS and TCS only applied to $\mathcal{D}_{\text{syn}}$ and MLM only to $\mathcal{D}_{\text{nat}}$:

$$\mathcal{L} = \sum\nolimits_{C_t \in \mathcal{D}_{\text{syn}}} (\mathcal{L}_{\text{CCS}}(C_t) + \mathcal{L}_{\text{TCS}}(C_t)) + \sum\nolimits_{C_t \in \mathcal{D}_{\text{nat}}} \mathcal{L}_{\text{MLM}}(C_t) \tag{5}$$

Figure 2 shows an overview of SCORE pre-training on an example SPARC dialogue from Figure 1. We report additional implementation details for pre-training SCORE in Section 3.3 and Appendix C.

### 2.3 DATA SYNTHESIS

We re-use the synthetic dataset of 120k synthetic task-oriented dialogues for MWOZ, introduced by Campagna et al. (2020). In this work, we introduce a complementary procedure to synthesize data for conversational text-to-SQL dialogues. We use about 400k tables in WIKITABLES (Bhagavatula et al., 2015) (after filtering and cleaning), WikiSQL, and Spider datasets as underlying databases $d$, and then synthesize about one dialog for each table. Finally, we synthesize 435k text-to-SQL conversations in total. Table 12 in Appendix B shows an example of the synthesized question-SQL pairs and their corresponding templates in our grammar.

To this end, we use only 500 dev examples from SPARC to induce two utterance-SQL generation grammars: (1) a single-turn context-free grammar $G_s$ for generating context-independent question-SQL pairs, and (2) a follow-up context-free grammar $G_c$ for follow-up question-SQL pairs. The single-turn grammar $G_s$ contains a list of synchronous question-SQL templates where typed slots (`COLUMN0`, `OP0`, `VALUE0`, ...) represent mentions of tables, columns, values, and SQL operations. The follow-up grammar $G_c$ contains context switch labels and lists of follow-up question templates. For example, if the context switch label is `INS(SELECT.column0)`, the corresponding question could be *"How about show `column0` too?"*. To ensure generalization, we only induce the grammars from the SPARC training set. Appendix B shows examples of the grammar rules and synthesized utterances.

**Algorithm 1** Data synthesis algorithm

$\tilde{h} \leftarrow \emptyset$
$r_s \leftarrow \text{SAMPLE}(G_s)$
$\tilde{u}_0, \tilde{q}_0 \leftarrow \text{RANDASSIGNSLOTS}(d, r_s)$
$\tilde{h} += (\tilde{u}_0, \tilde{q}_0)$
$\tilde{u}_p, \tilde{q}_p \leftarrow \tilde{u}_0, \tilde{q}_0$
**for** $t \leftarrow 1$ to $T$ **do**
  **if** $\text{RAND}(0, 1) < 0.2$ **then**
    $r_s \leftarrow \text{SAMPLE}(G_s)$
    $\tilde{u}_t, \tilde{q}_t \leftarrow \text{RANDASSIGNSLOTS}(d, r_s)$
  **else**
    $r_c \leftarrow \text{SAMPLE}(G_c)$
    **if** $\text{CONSTRAINTCHECK}(r_c, \tilde{q}_p)$ **then**
      $\tilde{u}_t, \tilde{q}_t \leftarrow \text{EDITASSIGN}(\tilde{q}_p, r_c)$
  $\tilde{h} += (\tilde{u}_t, \tilde{q}_t, r_c)$
  $\tilde{u}_p, \tilde{q}_p \leftarrow \tilde{u}_t, \tilde{q}_t$
**return** $\tilde{h}$

The data synthesis procedure using the two grammars is shown in Algorithm 1. Given a database $d$ and a sampled single-turn question-SQL template, the function RANDASSIGNSLOTS samples values (column names, cell values, and SQL operations) for typed slots in the template and returns the first synthesized question $\tilde{u}_0$ and the corresponding SQL query $\tilde{q}_0$. To generate $T$ follow-up question-SQL pairs, the function $\text{CONSTRAINTCHECK}(r_c, \tilde{q}_p)$ checks if the previous query $\tilde{q}_p$ satisfies constraints of the sampled template $r_c$ (e.g. contains its mentioned nonterminal). Finally, $\text{EDITASSIGN}(\tilde{q}_p, r_c)$ edits the previous SQL $\tilde{q}_p$ to generate the current follow-up SQL label $\tilde{q}_t$ and samples values for typed slots in the template to generate the corresponding follow-up question $\tilde{u}_t$.

## 3 EXPERIMENT SETTINGS

### 3.1 DATASETS AND EVALUATION METRICS

We evaluate SCORE on four popular CSP tasks: SPARC (sequential text-to-SQL), COSQL (conversational text-to-SQL), MWOZ (dialogue state tracking), and SQA (sequential question answering), summarized in Table 1.

**SPARC** (Yu et al., 2019b) [1] is a large collection of sequences of inter-related context-dependent question-SQL pairs. It contains 4.3K questions sequences and 12k+ questions. **COSQL** (Yu et al., 2019a) [2] is a large conversational text-to-SQL corpus, with 3k dialogues, collected under the Wizard-of-Oz (WOZ) setting. We focus on the SQL-grounded dialogue state tracking task which maps user intents into SQL queries if possible given the interaction history. Both SPARC and COSQL cover 200 complex DBs spanning 138 domains.

[1] `https://yale-lily.github.io/sparc`
[2] `https://yale-lily.github.io/cosql`

**MWOZ** (Budzianowski et al., 2018; Eric et al., 2019) [3] is a corpus of over 10k human-human written task-oriented dialogs created through a WOZ crowdsourcing setting. We focus on the belief state tracking task in MWOZ which maps multi-turn user utterances to slot-value annotations.

**SQA** (Iyyer et al., 2017) [4] is constructed from a subset of WikiTableQuestions (Pasupat & Liang, 2015) by decomposing highly compositional questions into a sequence of simple questions. The task is weakly-supervised because each resulting decomposed question is only annotated with answers as one or more table cells, while the logic program is latent. It has 6,066 question sequences with 17,553 questions in total on 982 unique open-domain tables from Wikipedia.

We adopt the official metrics defined for each of the tasks. For SPARC and COSQL, we report question match accuracy (QM): the exact set match accuracy (Yu et al., 2018b) over SQL templates and interaction match accuracy (IM): the ratio of interactions for which all questions are predicted correctly. For MWOZ, we report joint goal accuracy (JGA) which is similar to the IM accuracy used in SPARC and COSQL. Finally, for SQA, we report denotation QM and IM accuracies.

### 3.2 BASE MODELS AND OTHER BASELINES

For SPARC and COSQL, we use RAT-SQL (Wang et al., 2020) as our base model. Since it is originally developed for single-turn text-to-SQL, we extend it to a multi-turn setting by concatenating current utterances and dialog history (see Section 2.2). Note that RAT-SQL alone, without SCORE, achieves better or comparable results to state-of-the-art models developed for SPARC and COSQL.

For MWOZ, we employ Trippy (Heck et al., 2020). It achieves state-of-the-art performance on MWOZ and uses $\text{BERT}_{\text{base}}$ to encode user and system utterances and dialog history. We report higher results (around 2%) for Trippy than reported by Heck et al. (2020) since we train it for more epochs (25 vs. 10). To show the improvement of SCORE is not tied to specific base systems, we also experiment with another strong base model SOM-DST (Kim et al., 2020) for MWOZ and follow the same experimental details to train it.

For SQA, we use the weakly-supervised semantic parser proposed by Wang et al. (2019). The model first generates an abstract program given an input question and then instantiates it by searching for alignments between slots in the abstract program and question spans. As it is originally developed for single-turn questions, we extend it to the multi-turn setting in the same way as RAT-SQL.

We report additional implementation details for all base models in Appendix C. In addition to reporting results for all base models with SCORE, we also report original base models results (with BERT and/or ROBERTA) and several other state-of-the-art baselines for each task.

### 3.3 DATASET USAGE IN PRE-TRAINING

In our experiments and ablation study, we train several versions of SCORE with different objectives and datasets: (1) SCORE (MLM): pre-trained on annotated natural questions using MLM. (2) SCORE (CCS+TCS): pre-trained on only synthesized data, which achieves the best results on SParC, CoSQL, and SQA. (3) SCORE (CCS+TCS+MLM): pre-trained on the synthesized data using CCS+TCS and annotated natural questions using MLM.

Furthermore, note that the synthesized data is generated using grammar induced by about 500 examples from only SPARC. Therefore, no COSQL or SQA data are seen in any pre-training steps. For MWOZ, Campagna et al. (2020) study only the dev examples to induce the data synthesis grammar.

## 4 RESULTS AND ANALYSIS

**Overall Results** The results of SPARC and COSQL, MWOZ, and SQA are in Table 2, 3, and 4 respectively. We run each main experiment three times with different random seeds and report the mean. Overall, SCORE gains significant improvements over BERT and ROBERTA on all tasks, achieving state-of-the-art performances on SPARC, COSQL, and MWOZ.

[3] https://github.com/budzianowski/multiwoz

[4] http://aka.ms/sqa

| | SPARC | | | | COSQL | | | |
|---|---|---|---|---|---|---|---|---|
| | Dev | | Test | | Dev | | Test | |
| Models | QM | IM | QM | IM | QM | IM | QM | IM |
| SyntaxSQL (Yu et al., 2018a) | 18.5 | 4.3 | 20.2 | 5.2 | - | - | 14.2 | 2.2 |
| GAZP + BERT (Zhong et al., 2020) | 48.9 | 29.7 | 45.9 | 23.5 | 42.0 | 12.3 | 39.7 | 12.8 |
| EditSQL + BERT (Zhang et al., 2019c) | 47.2 | 29.5 | 47.9 | 25.3 | 39.9 | 12.3 | 40.8 | 13.7 |
| IGSQL + BERT | 50.7 | 32.5 | 51.2 | 29.5 | 44.1 | 15.8 | 42.5 | 15.0 |
| $R^2$SQL + BERT | - | - | 55.8 | 30.8 | - | - | 46.8 | 17.0 |
| RAT-SQL + BERT (Wang et al., 2019) | 56.8 | 33.4 | - | - | 48.4 | 19.1 | - | - |
| + ROBERTA | 58.2 | 36.7 | - | - | 50.1 | 19.3 | - | - |
| + SCORE | **62.2** | **42.5** | **62.4** | **38.1** | **52.1** | **22.0** | **51.6** | **21.2** |

Table 2: The SPARC and COSQL accuracy over all questions (QM) and all interactions (IM). The scores of IGSQL + BERT and $R^2$SQL + BERT are from the official leaderboards.

| Models | MWOZ 2.1 |
|---|---|
| DST-reader (Gao et al., 2019) | 36.40 |
| TRADE (Wu et al., 2019) | 46.60 |
| DS-DST (Zhang et al., 2019a) | 51.21 |
| SOM-DST (Kim et al., 2020) | 52.57 |
| DS-picklist (Zhang et al., 2019a) | 53.30 |
| TripPy (Heck et al., 2020) | 55.29 |
| SimpleToD (Hosseini-Asl et al., 2020) | 55.72 |
| TripPy (ours) | 58.37 |
| + SCORE | **60.48** |

Table 3: Joint goal accuracies (JGA) on MWOZ 2.1 test set. All models use a BERT-like encoder/GPT.

| | SQA | |
|---|---|---|
| Models | QM | IM |
| Pasupat & Liang (2015) | 33.2 | 7.7 |
| Neelakantan et al. (2017) | 40.2 | 11.8 |
| Iyyer et al. (2017) | 44.7 | 12.8 |
| Sun et al. (2019a) | 45.6 | 13.2 |
| Müller et al. (2019) | 55.1 | 28.1 |
| Herzig et al. (2020) | **67.2** | **40.4** |
| Wang et al. (2019) + RoBERTa | 62.8 | 33.2 |
| Wang et al. (2019) + SCORE | **65.4** | **38.5** |

Table 4: Question (QM) and interaction (IM) accuracy on the SQA test set.

For SPARC and COSQL in Table 2, compared with ROBERTA, SCORE boosts the performance by 4.0% QM / 5.8% IM on SPARC, and 2.0% QM / 2.7% IM on COSQL. This demonstrates the effectiveness of SCORE on contextual semantic parsing tasks. In addition, on MWOZ dialog state tracking task in Table 3, TripPy achieves 60.5% JGA by replacing BERT with SCORE, outperforming the prior state-of-the-art (Hosseini-Asl et al., 2020) by 4.8%. This indicates that dialog state tracking also benefits from SCORE. Finally, SCORE also achieves higher performance than ROBERTA on weakly supervised sequential question answering SQA task. As Table 4 shows, SCORE improves QM by 2.6% and IM by 4.9% over ROBERTA with Wang et al. (2019) as the base model. This demonstrates that the enhanced ability of semantic parsing and context modeling in SCORE is transferable to denotation-based CSP tasks.

**What is the effect of each pre-training objective?** Table 5 shows an ablation study on different pre-training objectives. We find that the best SCORE results are achieved by pre-training on only synthesized data (CCS+TCS) without any natural questions (MLM) on SPARC, COSQL, and SQA but not on MWOZ. By adding MLM to CCS+TCS (CCS+TCS vs. CCS+TCS+MLM), MLM actually hurts the performance (-3.9% on SParC, -0.3% on CoSQL, and -4.4% on SQA) while increases for MWOZ. One possible reason is that questions in MWOZ are more diverse in language but less compositional while semantic compositionality and turn changes are more important in the other three CSP tasks. Also, the synthesized data used to pre-train SCORE for SPARC and COSQL is generated by the grammar induced by SPARC, which might overfit to SPARC. In addition, SCORE pre-trained with only MLM loss improves the performance ( 1.0%) but not as large as CCS+TCS (+5.5% on SPARC, +1.7% on COSQL, and +3.4% on SQA). Finally, we test the effectiveness of TCS on SPARC, COSQL, and SQA by adding TCS to CCS (CCS only vs.

| Learning Objective | SPARC | COSQL | MWOZ | SQA |
|---|---|---|---|---|
| MLM only | 37.0(+0.3) | 20.3(+1.0) | 59.47(+1.10) | 34.7(+1.5) |
| CCS only | 41.3(+4.6) | 21.2(+1.9) | 59.32(+0.95) | 32.7(-0.5) |
| CCS+TCS | **42.5(+5.8)** | **22.0(+2.7)** | - | **38.5(+5.3)** |
| CCS+TCS+MLM | 38.6(+1.9) | 21.7(+2.4) | **60.48(+2.11)** | 33.7(+0.5) |

Table 5: The effect of SCORE pre-training objectives. Improvements are shown in the parentheses.

CCS+TCS), SCORE gains improvements of 1.2% on SPARC and 0.8% on COSQL, and 4.4% on SQA.

**Does SCORE improve question match accuracy on individual turns?** Table 6 shows detailed results of SCORE's question accuracy for individual conversation turns on the SPARC dev set. SCORE provides a significant improvement for every conversation turn except the first (in which the task is more similar to single-turn semantic parsing). COSQL and SQA exhibit similar behavior and are presented in Appendix A.

| | QM | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|
| RAT-SQL + BERT | 56.8 | **71.1** | 53.6 | 47.8 | 31.8 |
| +RoBERTa | 58.2 | 68.7 | 58.5 | 48.9 | 35.2 |
| + SCORE | **62.2** | 70.6 | **63.5** | **52.6** | **45.5** |

Table 6: Detailed results on the dev set of SPARC. $Q_i$ is the accuracy of the $i^{\text{th}}$ conversation question.

**What if we use the synthesized data to simply augment the training data?** To answer this, we compare the results of the base models trained with or without the synthesized data on COSQL and MWOZ. As shown in Table 7, the extra synthetic data does not significantly improve the performance, indicating that directly augmenting the synthetic data to the training set is not effective. The similar findings are reported in many recent work (Zhang et al., 2019c; Herzig et al., 2020; Campagna et al., 2020; Zhong et al., 2020). In contrast, pre-training on the synthesized data with our objectives improves the performance on the downstream tasks.

| | COSQL | MWOZ |
|---|---|---|
| no syn | 48.4 | 58.37 |
| with syn | 48.6 | 58.45 |

Table 7: Effect of synthetic data as training data augmentation.

**How general is SCORE and its synthetic grammar?** For generalization in task settings, we have shown that the pre-training strategy of SCORE can improve the performance over different CSP tasks including semantic parsing (SPARC and COSQL), dialog state tracking (MWOZ), and weakly supervised table question answering (SQA). In addition, we demonstrate the effectiveness of SCORE on different *base models*. To this end, we experiment with a different base model SOM-DST for MWOZ. As shown in Table 8, SCORE can still improve the performance with a different base model on MWOZ (SOM-DST+BERT vs. SOM-DST+SCORE on syn. MWOZ).

| | MWOZ |
|---|---|
| SOM-DST + BERT | 52.57 |
| + SCORE on syn. text-to-SQL | 53.57 |
| + SCORE on syn. MWOZ | 54.61 |

Table 8: Performance of SCORE pre-trained on different synthesized data on MWOZ.

To demonstrate the generalization in synthetic grammar and data, as shown in Table 2 and 4, SCORE (TCS+CCS) is pre-trained on the synthesized data of the grammar induced from SPARC *only*, and it still improves the performance on COSQL (+2.7%) and SQA (+4.9%) where *no* any CoSQL and SQA annotated data is seen in any pre-training steps. Moreover, in Table 8 we show that SCORE pre-trained on the text-to-SQL synthesized data could also surprisingly improve the performance on MWOZ. We expect that higher performance could be achieved with SCORE pre-trained on task-specific synthesized data. Finally, our pre-training approach can be applied to *any* existing LMs including larger seq2seq LMs (e.g., BART (Lewis et al., 2020), T5 (Raffel et al., 2020)).

**Can SCORE deliver more value when in-domain data is limited (e.g., in a low-resource setting)?** We want to answer this question similar to experiments other investigations of LMs as few-shot learners (Wu et al., 2020; Brown et al., 2020; Schick & Schütze, 2020). To this end, we compare ROBERTA and SCORE under a few-shot setting on SQA when only 10% of training data is available. We choose SQA because its annotation is most different from the synthetic text-to-SQL dataset we use for pretraining. Table 9 demonstrates that SCORE delivers even larger improvements compared to the ROBERTA baseline when only 10% training data is available (3.8% vs 2.6%).

| | QM | IM |
|---|---|---|
| RoBERTa | 53.3 | 21.2 |
| SCORE | **57.1** | **26.1** |

Table 9: Performance of SCORE on 10% training data of SQA.

## 5 RELATED WORK

**Conversational Semantic Parsing** Conversational semantic parsing is one of the most important research topics in conversational AI and has been studied in different settings including task-oriented dialogues, question answering, and text-to-SQL. Task-oriented dialog systems (Henderson et al.,

2014; Wen et al., 2016; Mrkšić et al., 2017; Budzianowski et al., 2018) aim to help users accomplish a specific task (e.g. flight booking) and often pre-define slot templates grounded in a domain-specific ontology. In comparison, several other datasets were recently introduced for cross-domain conversational text-to-SQL tasks (SPARC and COSQL (Yu et al., 2019a;b)) and sequential questions answers over tables (Iyyer et al., 2017). While the previous work has achieved significant progress in different datasets separately, to the best of our knowledge, we are the first to study four different CSP tasks together (sequential text-to-SQL, conversational text-to-SQL, dialog state tracking, and weakly-supervised sequential question answering) by addressing the shared key challenge of learning representations in pre-trained language models that capture the alignment between the dialogue flow and the structural context.

**Conversational Language Model Pre-training** Several recent efforts have demonstrated the value of adapting pre-trained LMs to specific tasks using different pre-training objectives, e.g., summarization (Zhang et al., 2019b), knowledge inference (Sun et al., 2019b; Liu et al., 2019a), etc. Closest to our work is adapting pre-trained LMs for open-domain chit-chat models and for tabular data representation. The former focuses on improving response generation on open-ended dialogues by adding a pre-training step on open-domain conversations data, such as Reddit data (Zhang et al., 2020; Henderson et al., 2019). For example, Wu et al. (2020) introduced ToD-BERT, a pre-trained language model combining 9 high-quality human-human task-oriented dialogue datasets to conduct language model and response selection pre-training. However, they use language modeling training objectives over free-form text and therefore have limited ability to represent structural data. The latter has focused on improving language model pre-training for encoding tabular data (Yin et al., 2020; Herzig et al., 2020), but they focus on the single turn semantic parsing setting. Our approach is different from previous work because we address the challenge of conversational semantic parsing tasks by learning pretrained representation for both the multi-turn dynamics of the dialog and the relation between the unstructured language utterance and the structured ontology. Furthermore, our pre-training approach is much more data-efficient than prior LM pre-training work and saves a lot of time and computing resources (Appendix D for more details). Our pre-training step can be done within only one day using 8 V100 GPUs.

**Using Synthesized Data for Semantic Parsing** Synthesized data has been frequently used in semantic parsing to alleviate the challenge of labeled data scarcity. For example, Wang et al. (2015) proposed a method for training semantic parsers in new domains by generating logical forms and canonical utterances and then paraphrasing the canonical utterances via crowd-sourcing. Similar approaches were used to train semantic parsers in other domains and settings (Zhong et al., 2017; Su et al., 2017; Cheng et al., 2018; Shah et al., 2018). Another line of work has proposed using synthesized data to adapt single turn semantic parsing models to new domains (Jia & Liang, 2016; Yoo et al., 2018; Campagna et al., 2019) and task-oriented dialogues (Campagna et al., 2020). However, they reported that combining synthetic data and the supervised data does not yield significant improvements, consistent with results by Herzig et al. (2020). By contrast, we introduce a new data synthesize procedure for conversational text-to-SQL dialogues and use it in a different way by pretraining language models to induce better representations for many CSP tasks. Our synthesized data can be easily generated without human involvement and the pre-trained models add value to different tasks simultaneously.

## 6 CONCLUSION

We presented SCORE a new pre-training approach for conversational semantic parsing. The training objectives of SCORE aim to induce natural language representations that capture the multi-turn dynamics, compositional semantic of the target language, and the references to the structural ontology appearing in the dialog. SCORE can be used with many semantic parsing models as a drop-in replacement for general pretrained LMs. We demonstrated SCORE effectiveness by using it as a feature representation encoder with strong baseline models for a wide range of CSP tasks. In particular, our empirical results on four different CSP tasks demonstrated that SCORE can be used to significantly improve the performance of existing strong baseline models by simply replacing an existing pre-trained LM with our SCORE pre-trained model. Furthermore, we are able to achieve state-of-the-art results on three of these tasks. We hope SCORE will encourage further exploration of the benefits and limitations of pre-training approaches for CSP systems.

## REFERENCES


Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

Chandra Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel: Entity linking in web tables. In *International Semantic Web Conference*, 2015.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*, 2018.

Giovanni Campagna, Silei Xu, Mehrad Moradshahi, Richard Socher, and Monica S. Lam. Genie: A generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 394–410. Association for Computing Machinery, 2019.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2020.

Joyce Y Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. Language to action: Towards interactive task learning with physical agents. In *IJCAI*, pp. 2–9, 2018.

Jianpeng Cheng, Siva Reddy, and Mirella Lapata. Building a neural semantic parser from a domain ontology. *ArXiv*, abs/1812.10037, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*, 2019.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. Dialog state tracking: A neural reading comprehension approach. In *SIGDial*, 2019.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020. Association for Computational Linguistics.

M. Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, M. Moresi, and Milica Gavsi'c. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *SIGdial*, 2020.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. The second dialog state tracking challenge. In *SIGDIAL Conference*, 2014.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkvsi'c, P. Su, Tsung-Hsien, and Ivan Vulic. Convert: Efficient and accurate conversational representations from transformers. *ArXiv*, abs/1911.03688, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ArXiv*, abs/1610.02136, 2016.

Jonathan Herzig, P. Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. In *ACL*, 2020.

Ehsan Hosseini-Asl, B. McCann, Chien-Sheng Wu, Semih Yavuz, and R. Socher. A simple language model for task-oriented dialogue. *ArXiv*, abs/2005.00796, 2020.

Wonseok Hwang, Jinyeung Yim, Seunghyun Park, and Minjoon Seo. A comprehensive exploration on wikisql with table-aware word contextualization. *ArXiv*, abs/1902.01069, 2019.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 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)*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Sina Jafarpour, Christopher JC Burges, and Alan Ritter. Filter, rank, and transfer the knowledge: Learning to chat. *Advances in Ranking*, 10:2329–9290, 2010.

Robin Jia and Percy Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

Sung-Dong Kim, Sohee Yang, Gyuwan Kim, and S. Lee. Efficient dialogue state tracking by selectively overwriting memory. In *ACL*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

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

Q. Liu, B. Chen, Jiaqi Guo, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. How far are we from effective context modeling ? an exploratory study on semantic parsing in context. *ArXiv*, abs/2002.00652, 2020.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph, 2019a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019b.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1777–1788. Association for Computational Linguistics, 2017.

Thomas Müller, Francesco Piccinno, Massimo Nicosia, Peter Shaw, and Yasemin Altun. Answering conversational questions on structured data without logical forms. In *EMNLP/IJCNLP*, 2019.

Arvind Neelakantan, Quoc V. Le, M. Abadi, A. McCallum, and Dario Amodei. Learning a natural language interface with neural programmer. *ArXiv*, abs/1611.08945, 2017.

Panupong Pasupat and Percy Liang. 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 1470–1480, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 583–593, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Ohad Rubin and Jonathan Berant. Smbop: Semi-autoregressive bottom-up semantic parsing. *arXiv preprint arXiv:2010.12412*, 2020.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*, 2020.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. Building a conversational agent overnight with dialogue self-play, 2018.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? *arXiv preprint arXiv:2010.12725*, 2020.

Yu Su, Ahmed Hassan Awadallah, Madian Khabsa, P. Pantel, M. Gamon, and Mark J. Encarnación. Building natural language interfaces to web apis. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.

Yibo Sun, Duyu Tang, Nan Duan, Jingjing Xu, X. Feng, and B. Qin. Knowledge-aware conversational semantic parsing over web tables. In *NLPCC*, 2019a.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration, 2019b.

Gokhan Tur and Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

Bailin Wang, Ivan Titov, and Mirella Lapata. Learning semantic parsers from denotations with latent structured alignments and abstract programs. In *Proceedings of EMNLP*, 2019.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7567–7578, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.677.

Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. 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)*, pp. 1332–1342, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1129.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745.

Kang Min Yoo, Youhyun Shin, and Sang goo Lee. Data augmentation for spoken language understanding via joint variational generation, 2018.

Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir Radev. Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task. In *Proceedings of EMNLP*. Association for Computational Linguistics, 2018a.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *EMNLP*, 2018b.

Tao Yu, Rui Zhang, He Yang 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. 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 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2019a.

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

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*, 2019a.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*, 2019b.

Rui Zhang, Tao Yu, He Yang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. Editing-based sql query generation for cross-domain context-dependent questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2019c.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*, 2020.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

Victor Zhong, M. Lewis, Sida I. Wang, and Luke Zettlemoyer. Grounded adaptation for zero-shot executable semantic parsing. *The 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

# A DETAILED RESULTS

| | QM | IM | Q1 | Q2 | Q3 | Q4 | Q5 |
|---|---|---|---|---|---|---|---|
| RAT-SQL + BERT | 48.4 | 19.1 | 54.6 | 48.4 | **47.5** | 43.9 | 31.0 |
| +RoBERTa | 50.1 | 19.3 | 59.7 | 50.9 | 46.3 | 46.5 | **32.4** |
| + SCORE | **52.1** | **22.0** | **60.8** | **53.0** | **47.5** | **49.1** | **32.4** |

Table 10: Detailed results of COSQL on the dev set. $Q_i$ is the accuracy of the $i^{\text{th}}$ question in the conversation.

| | QM | IM | Q1 | Q2 | Q3 |
|---|---|---|---|---|---|
| Wang et al. (2019) | 51.0 | 22.0 | 68.3 | 48.0 | 38.5 |
| +RoBERTa | 62.8 | 33.2 | 77.2 | 61.7 | 52.1 |
| +SCORE | **65.4** | **38.5** | **78.4** | **65.3** | **55.1** |
| Few-Shot (10% training data) | | | | | |
| Wang et al. (2019) | | | | | |
| +RoBERTa | 53.3 | 21.2 | 71.0 | 52.5 | 36.6 |
| +SCORE | **57.1** | **26.7** | **74.6** | **56.7** | **40.7** |

Table 11: Detailed results of SQA on the test set. $Q_i$ is the accuracy of the $i^{\text{th}}$ question in the conversation.

# B SYNTHESIZED EXAMPLES & TEMPLATES

Table 12 shows an example of the synthesized question-SQL pairs and their corresponding templates in our grammars.

| Turn # | Question-SQL Template | Synthesized Question-SQL |
|---|---|---|
| 1 | "Find the number of TABLE0 with COLUMN0 OP0 VALUE0"<br>SELECT COUNT(*) ORDER BY COLUMN0 OP0 VALUE0 | "Find the number of football team with team hometown is not murrieta, california?"<br>SELECT COUNT(*) WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA" |
| 2 | "Can you give me their COLUMN1?"<br>TCS: REPLACE(SELECT.COLUMN0), DEL(SELECT.AGG) | "Can you give me their football team player?"<br>SELECT FOOTBALL_TEAM_PLAYER WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA" |
| 3 | "How about only show those with AS0 COLUMN2?"<br>TCS: ADD(ORDERBY_AS0.COLUMN2) | "How about only show those with the largest age?"<br>SELECT FOOTBALL_TEAM_PLAYER WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA" ORDER BY AGE DESC LIMIT 1 |
| 4 | "AS1?"<br>TCS: REPLACE(ORDERBY_AS1.COLUMN2) | "The smallest?"<br>SELECT FOOTBALL_TEAM_PLAYER WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA" ORDER BY AGE AS LIMIT 1 |

Table 12: An example of synthetic conversational text-to-SQL data.

# C IMPLEMENTATION DETAILS

## C.1 SCORE

For pre-training SCORE on synthesized text-to-SQL data, we use ROBERTA $_{\text{large}}$ and pre-train it with batch size 12, gradient accumulation step 2, and maximum length 248. We use a learning rate $1e$-5 and gradually reduce the learning rate without a warm-up period using Adam (Kingma & Ba, 2014) with epsilon $1e$-8. $\text{BERT}_{\text{base}}$ is used in pre-training SCORE on synthesized MWOZ data because it contains longer conversations. We set the maximum length to 512 and batch size 24. All SCORE are pre-trained for 30 epochs, which usually take less than half a day on 8 V100 GPUs.

We experimented with SCoRe pre-trained for 5, 10, and 30 epochs and found that most of the best downstream performances occur when base systems incorporate with SCoRe pre-trained for less than 10 epochs. Our implementation is based on the Transformers library (Wolf et al., 2019).

### C.2 Base Models

**RAT-SQL:** For a fair comparison, all RAT-SQL experiments are trained for 40k steps. We adopt the same hyperparameters as Shaw et al. (2018) except for learning rates. We find that learning rates of $1e$-4 and $1e$-5 for RAT and BERT respectively produce more stable results.

**TripPy:** We use the same hyperparameters for training TripPy on MWoZ as in (Heck et al., 2020) except we train it for 25 epochs (as opposed to 10 epochs as reported in (Heck et al., 2020)). When we train TripPy for 25 epochs, we get a new result that is higher (around 2%) than the one reported in (Heck et al., 2020). Similarly, when we train TripPy with SCoRe, we train it for 25 epochs.

**SOM-DST:** We use the same hyperparameters from Kim et al. (2020) for all SOM-DST experiments on MWoZ.

**Wang et al. (2019):** We use the same hyperparameters from Wang et al. (2019) for SQA experiments. Note that Herzig et al. (2020) outperform Wang et al. (2019) on SQA because (1) they don't generate logic forms but select table cells and applying aggregation operators. Wang et al. (2019) generate latent programs, yet the grammar of the latent program can only cover 87% questions. (2) They reduce the search space by reusing the previous question answer. We choose Wang et al. (2019) as our base model because generating symbolic programs has many practical advantages (even at a cost of around 1% accuracy drop), such as showing interpretable reasoning steps, enabling formal reasoning, and operationalization without GPU/TPU accelerators.

## D Pre-Training Cost

We test the performance of SCoRe with respect to the number of pre-training epochs. Figure 3 shows that the best performance of the downstream tasks is usually achieved in early epochs, more specifically 5 for SParC and CoSQL and 15 for MWoZ. Longer pre-training time does not improve or even hurts the performance. One possible reason is that longer pre-training makes SCoRe overfit to the synthesized data whose utterances are unnatural.

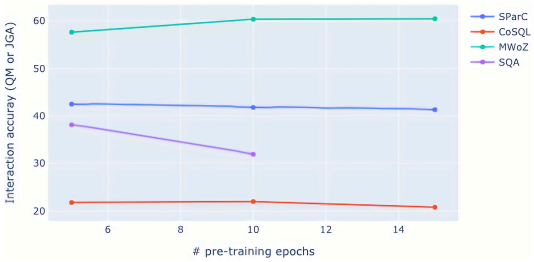


Figure 3: The effect of pre-training time.

As for the data, as shown in Table 5, even if SCoRe is pre-trained with only a relatively small amount of synthesized data (without the MLM loss), most of the tasks can achieve much higher performances. With a relatively smaller training corpus and shorter training time compared to other pre-trained language models, SCoRe is efficient in time and data.

## E Additional Results

**Effect of TCS** We ran the TCS only experiment on SParC, and will add TCS only results (including for other tasks) to Table 5 in the final version. SCoRe (TCS only) outperforms RoBERTa by 2.4% so far (note: training is still going on) on SParC (39.1% vs. 36.7%). Also, as discussed in Section 4, we also provide a secondary evidence by testing the effectiveness of TCS on SParC, CoSQL, and SQA by adding TCS to CCS (CCS only vs. CCS+TCS), SCoRe (with TCS) gains improvements of 1.2% on SParC and 0.8% on CoSQL, and 4.4% on SQA.

**Incorporating Additional Examples Used in Synthetic Grammar Induction** As we mentioned in Section 2.3, we used about 500 examples from SParC to induce the grammar for data synthesis in pre-training. For a fair comparison, we also report the results of incorporating the additional SParC examples in CoSQL and SQA. More specifically, we directly concatenate the additional SParC

examples to CoSQL training set, and train RAT-SQL+RoBERTa on it, which slightly improves the performance (19.6% vs. 19.3%) but not as large as SCoRe (22.0% vs. 19.3%).' Also, because SQA is weakly-supervised sequential question answering, which differs from SParC, we first fine-tune RoBERTa on the additional SParC examples using CCS, and then apply it to SQA. In this way, the RoBERTa trained with additional SParC examples achieves a similar performance as the original one (62.7% vs 62.8%).

**Performance Comparison with ToD-BERT** ToD-BERT is pre-trained on human-annotated questions with both MLM and response contrastive objectives. To compare TOD-BERT with SCoRe, we ran experiments of RAT-SQL + ToD-BERT on SParC. SCoRe (62.2%) outperforms ToD-BERT (54.6%) by 7.6%.

**Comparison with Finetuning Larger Language Models** Based on our experiments and other published results, we didn't find existing larger LMs (BART (Lewis et al., 2020), T5 (Raffel et al., 2020), GPT-2 (Radford et al., 2019)) outperform custom models + BERT on CSP tasks. Our evidence is based on Spider (Yu et al., 2018b), which is the single-turn version of SParC and CoSQL. For T5, Shaw et al. (2020) applied T5 as seq2seq to Spider, and compared with RAT-SQL + BERT-Large, T5-Base performs much worse (57.1% vs. 69.6%), and T5-3B improves only 0.3, but it is 6 times larger. Moreover, for Bart, we have performed experiments on Spider and we found that BART cannot outperform custom models + BERT: RAT-SQL + BERT 69.7%, RAT-SQL + BART encoder 67.8%, BART encoder + decoder (406M, as a seq2seq task) 62.4%. In Rubin & Berant (2020), BART didn't outperform BERT either. As for GPT-2, Wu et al. (2020) and Hosseini-Asl et al. (2020) found it does not outperform BERT on MWoZ.

## F Task-Oriented Dialogue Datasets

| Name | # Dialogue | # Utterance | Avg. Turn | # Domain |
|---|---|---|---|---|
| MetaLWOZ (Lee et al., 2019) | 37,884 | 432,036 | 11.4 | 47 |
| Schema (Rastogi et al., 2019) | 22,825 | 463,284 | 20.3 | 17 |
| Taskmaster (Byrne et al., 2019) | 13,215 | 303,066 | 22.9 | 6 |
| MWOZ (Budzianowski et al., 2018) | 10,420 | 71,410 | 6.9 | 7 |
| MSR-E2E (Li et al., 2018) | 10,087 | 74,686 | 7.4 | 3 |
| SMD (Eric and Manning, 2017) | 3,031 | 15,928 | 5.3 | 3 |
| Frames (Asri et al., 2017) | 1,369 | 19,986 | 14.6 | 3 |
| WOZ (Mrkšić et al., 2016) | 1,200 | 5,012 | 4.2 | 1 |
| CamRest676 (Wen et al., 2016) | 676 | 2,744 | 4.1 | 1 |

Figure 4: Data statistics of human-annotated task-oriented dialogue datasets used in Wu et al. (2020).