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

Anonymous authors
Paper under double-blind review

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. Our implementation and checkpoints of the model will be available at Anonymous URL.

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 to specific domains (Gururangan et al., 2020) or tasks (Zhang et al., 2019a) 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.

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.

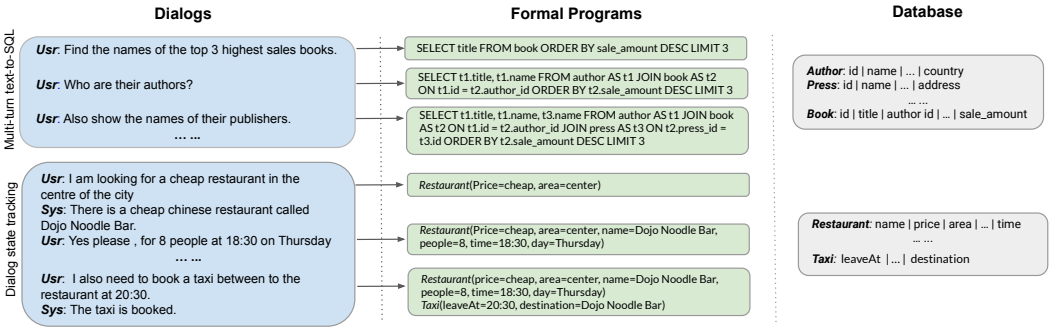

Figure 1: Examples of conversational semantic parsing tasks from SPARC and MWOZ datasets.

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 pre-training 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). As such, 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 conversational semantic parsing 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)* objective and the *Turn Contextual Switch (TCS)* objective. Furthermore, 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. Because the size of existing semantic parsing datasets is limited, we produce synthesized data for pretraining SCORE by sampling from the context-free grammar that is induced from complex text-to-SQL examples in different domains.

2.1 PRELIMINARIES

Task Definition In the CSP task, 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 variants of the problem, we 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:

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

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., 2019b; 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:

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

where $\boldsymbol{h}^c_{t,i}$ is the contextualized representation of the ith column's special token `</s>` in the contextualized input C_t and $\boldsymbol{W}_2$ is a learned parameter matrix.

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.

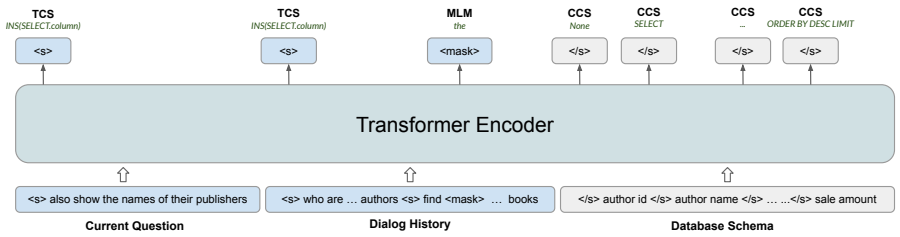

Figure 2: Pre-training of a SCoRe encoder on a SParC text-to-SQL example from Figure 1.

Based on the context-free grammar of SQL, we identify 26 possible *kinds of differences* that a conversational turn could elicit. 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:

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

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, 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 As Table 1 indicates, the total number of labeled dialog turns in CSP datasets is ca. 160,000. Following established practice on large-scale language model pre-training, we augment these datasets with *synthesized conversational data*. Importantly, incorporating synthetic data directly into the training set of a supervised task is known to be ineffective (Zhang et al., 2019b; Herzig et al., 2020b; Campagna et al., 2020; Zhong et al., 2020), which we also replicate in Section 4. Instead, we incorporate synthetic data into pre-training, forming an auxiliary dataset $\mathcal{D}_{\text{syn}}$ in addition to the naturally supervised dataset $\mathcal{D}_{\text{sup}}$. In Section 4, we show that this approach dramatically outperforms using synthetic data directly for fine-tuning. Section 2.3 details our data synthesis method.

The total pre-training loss is the sum of the three with CCS and TCS only applied to $\mathcal{D}_{\text{syn}}$ and MLM only to $\mathcal{D}_{\text{sup}}$:

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

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 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 unique tables in WikiTables (Bhagavatula et al., 2015), WikiSQL, and Spider datasets as underlying databases d, over which we generate 435k text-to-SQL conversations.

To this end, we 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

$\tilde{h} \leftarrow \emptyset$
$r_s \leftarrow$ SAMPLE(G_s)
$\tilde{u}_0, \tilde{q}_0 \leftarrow$ 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 RAND$(0, 1) < 0.2$ **then**
    $r_s \leftarrow$ SAMPLE(G_s)
    $\tilde{u}_t, \tilde{q}_t \leftarrow$ RANDASSIGNSLOTS(d, r_s)
  else
    $r_c \leftarrow$ SAMPLE(G_c)
    if CONSTRAINTCHECK$(r_c, \tilde{q}_p)$ **then**
      $\tilde{u}_t, \tilde{q}_t \leftarrow$ 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 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, 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.

MWOZ (Budzianowski et al., 2018; Eric et al., 2019) [3] is a corpus of over 100k 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 a weakly-supervised sequential question answering task over Wikipedia tables. Each question is annotated with one or more table cells representing the answer.

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.

[1] https://yale-lily.github.io/sparc
[2] https://yale-lily.github.io/cosql
[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., 2019b)	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^2SQL + 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^2SQL + BERT are from the official leaderboards.

Models	MultiWOZ 2.1
DST-reader	36.40
TRADE	46.60
DS-DST	51.21
SOM-DST	52.57
DS-picklist	53.30
TripPy	55.29
SimpleToD	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. (2020b)	**67.2**	**40.4**
Wang et al. (2019) + RoBERTa	62.8	33.2
with 10% training data	53.3	21.2
Wang et al. (2019) + SCORE	65.4	38.1
with 10% training data	57.1	26.1

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

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.

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.

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. We also compare ROBERTA and SCORE under a few-shot setting on SQA when only 10% of training data is available. As shown in Table 4, the performance gain of SCORE becomes larger.

Does SCORE improve question match accuracy on individual turns? Table 5 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 5: Detailed results on the dev set of SPARC. Q_i is the accuracy of the ith conversation question.

What is the effect of each pre-training objective? Table 6 shows an ablation study on different pre-training objectives. We find that the performance drops for COSQL and MWOZ while increases for SPARC when removing the MLM loss (CCS+TCS). One possible reason is that SPARC is created by decomposing complex questions in sequences of inner related simpler questions with more strict underlying patterns and less language diversity. 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 doesn't help but even hurt the performance on SPARC. For the other tasks, the MLM loss slightly improves the performance especially on COSQL. Finally, we test the effectiveness of TCS on SPARC and COSQL (CCS only vs. CCS + TCS), SCORE gains improvements of 1.5% on SPARC and 0.5% on COSQL.

	SPARC	COSQL	MWOZ	SQA
CCS + TCS + MLM	38.6	21.7	60.48	33.7
MLM only	37.0	20.3	59.47	34.7
CCS only	41.3	21.2	59.32	32.7
CCS + TCS	42.5	22.0	-	38.1

Table 6: The effect of SCORE pre-training objectives.

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., 2019b; Herzig et al., 2020b; 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? In addition to generalization over different *task settings*, 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.

For generalization in synthetic grammar and data, as shown in Table 2 and 4, SCORE is pre-trained on the synthesized data of the grammar induced from SPARC, and it still improves the performance on COSQL and SQA. 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.

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). The dialogue state tracking task and the conversational text-to-SQL task are in fact similar since we can view: (1) the slot-value annotations as SQL queries that only have `SELECT` and `WHERE` clauses, and (2) the pre-defined ontology as a database schema.

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., 2019a), knowledge inference (Sun et al., 2019b; Liu et al., 2019a), etc.). Perhaps closest to our work is work on adapting pre-trained LMs for open-domain chit-chat models and on tabular data representation. Pre-training for open-domain conversations 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). Recently, 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. Another line of recent work has focused on improving language model pre-training for single turn semantic parsing over tables to improve the encoding of tabular data (Yin et al., 2020; Herzig et al., 2020a). Our work follows this line of work but with a focus on conversational semantic parsing tasks by creating representations that capture the relation between the unstructured language utterance and the structured ontology while representing the multi-turn dynamics of the dialog.

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-tune 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. (2020b). We use synthesized data, in a different way, to induce better representations for 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.

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.

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*, 2020a.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020b.

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.

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.

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.

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.

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, S. Hoi, R. Socher, and Caiming Xiong. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *ArXiv*, abs/2004.06871, 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.

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*, 2019a.

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, 2019b.

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 9: Detailed results of CoSQL on the dev set. Q_i is the accuracy of the ith 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**	**38.1**	**78.3**	**65.3**	**54.9**
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 10: Detailed results of SQA on the test set. Q_i is the accuracy of the ith question in the conversation.

B SYNTHESIZED EXAMPLES & TEMPLATES

Table 11 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" SELECT COUNT(*) ORDER BY COLUMN0 OP0 VALUE0	"Find the number of football team with team hometown is not murrieta, california?" SELECT COUNT(*) WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA"
2	"Can you give me their COLUMN1?" TCS: REPLACE(SELECT.COLUMN0), DEL(SELECT.AGG)	"Can you give me their football team player?" SELECT FOOTBALL_TEAM_PLAYER WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA"
3	"How about only show those with AS0 COLUMN2?" TCS: ADD(ORDERBY_AS0.COLUMN2)	"How about only show those with the largest age?" SELECT FOOTBALL_TEAM_PLAYER WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA" ORDER BY AGE DESC LIMIT 1
4	"AS1?" TCS: REPLACE(ORDERBY_AS1.COLUMN2)	"The smallest?" SELECT FOOTBALL_TEAM_PLAYER WHERE TEAM_HOMETOWN != "MURRIETA, CALIFORNIA" ORDER BY AGE AS LIMIT 1

Table 11: 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.

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.

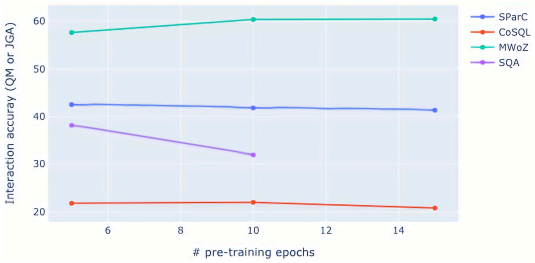

Figure 3: The effect of pre-training time.

As for the data, as shown in Table 6, 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 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).