Controlling Industrial Robots with High-level Verbal Commands*

Dongkyu Choi[1], Shi Wei[2], Ying Siu Liang[1],
Yeo Kheng Hui[2], and Jung-jae Kim[2]

[1] Institute of High Performance Computing
Agency for Science, Technology and Research, Singapore
{choi_dongkyu,liangys}@ihpc.a-star.edu.sg
[2] Institute for Infocomm Research
Agency for Science, Technology and Research, Singapore
{shi_wei,yeokh,jjkim}@i2r.a-star.edu.sg

Abstract. Industrial robots today are still mostly pre-programmed to perform a specific task. Despite previous research in human-robot interaction in the academia, adopting such systems in industrial settings is not trivial and has rarely been done. In this paper, we introduce a robotic system that we control with high-level verbal commands, leveraging some of the latest neural approaches to language understanding and a cognitive architecture for goal-directed but reactive execution. We show that a large-scale pre-trained language model can be effectively fine-tuned for translating verbal instructions into robot tasks, better than other semantic parsing methods, and that our system is capable of handling through dialogue a variety of exceptions that happen during human-robot interaction including unknown tasks, user interruption, and changes in the world state.

Keywords: Intention translation · Semantic parsing · Human-robot interaction · Cognitive architecture

1 Introduction

Despite the recent advance in human-robot interaction, we still see industrial robots being pre-programmed to perform a specific task in a secure environment. Although new collaborative robots, or co-bots, are safer around humans and therefore used in or near human work spaces, making them to work on a new task still involves manual programming in many industry cases. We aim to change this paradigm and develop a system that enables robots to adapt to the human norm, in which workers interact with their teammates to gather sufficient information about new tasks they need to perform and inform each other about the progress. There can be multiple modalities with which teammates can communicate with one another but, in this work, we present a system that can understand natural

* This research is supported by A*STAR under its *Human-Robot Collaborative AI for Advanced Manufacturing and Engineering* (Award A18A2b0046).

verbal instructions from humans, translating the utterances into a machine-readable representation of tasks, and perform the given tasks appropriately in situations that evolve dynamically.

Previous works on the instruction-to-task translation task utilize verb frames [3], grammar-based semantic parsers [11, 19], syntactic parser-based probabilistic models [14, 12], and end-to-end neural networks [18] including large-scale pre-trained language models [5]. However, they have not explored latest techniques of *semantic parsing*, the task of identifying the semantics of a given sentence and representing it in a machine-readable representation. In this work, we adapt recent and well-known methods of semantic parsing for the translation task and show that a pre-trained model [17] outperforms the other methods of semantic parsing probably because the pre-trained model can be effectively fine-tuned with the small size of our dataset.

In the sections that follow, we first review previous research on semantic parsing, as wells as translating instructions for robots, that influenced our work. Then we explain how our system leverages latest techniques for natural language processing to translate user utterances to goals to achieve in the given situation. After that, we describe our robot's cognitive architecture that takes the translated goals and execute procedures that achieve them in a reactive manner. We then discuss some future work before we conclude.

2 Related work

2.1 Translating instructions for robot motion planning

In this section, we discuss selected previous works on translating human instruction into meaning representation of machine-readable format for the purpose of robot motion planning. [3] presented a model that translates simple instruction sentences with a single verb frame into robot motion plans, even if the instructions are incomplete, by using commonsense reasoning. However, there is a need for expressing more complex intentions than single verb frames.

To address the need, [11, 19] developed combinatory categorical grammar (CCG)-based semantic parsers, where a grammar is to understand the whole meaning of even complex sentences. CCG is a lexical grammar, where most of linguistic information required for natural language understanding are specified at the lexical level; in other words, each word is associated with its pre-defined semantics. While a CCG-based semantic parser can be learned from a large collection of meaning representations [1], such a grammar-based approach cannot be automatically adapted for a small dataset in the robotics domain.

Instead of adapting a grammar-based semantic parser, [14, 12] utilize syntactic parsers, which identify the syntactic relations among words in a given sentence but do not represent them in a domain-specific meaning representation. For instance, [14] proposed dynamic grounding graphs (DGG). DGG first performs syntactic parsing on a given instruction, producing a tree-like syntactic structure of the sentence, then maps each word phrase of the structure

to its groundings (e.g. object, location, motion, task), and finally computes cost function parameters used in optimization-based motion planner by using a probabilistic model based on the groundings and a given environment. However, these methods require a large data collection to learn such a probabilistic model that can translate syntactic information into robot motion plans ([14] used 100,000 samples, and [12] used 6,099 utterances).

[18] and other works (available at archives) present neural networks that learn to translate an instruction utterance to meaning representation without a grammar or a syntactic parser, including LSTM [9] and BERT [5]. However, they have not explored more recent and well-known methods of semantic parsing for the translation task. We thus adapt 4 recent advanced semantic parsing methods for the task and discuss their evaluation comparison results in this work.

2.2 Semantic parsing

In this section, we discuss selected recent works on neural network-based semantic parsing. [6] present a neural network model, called *Coarse2Fine*, which has two encoders for training, while using only one encoder for inference. Coarse2Fine assumes that a meaning representation can be simplified into an *intermediate* form, which can be generated automatically from the ground-truth meaning representation. They train a model that first learns to generate the intermediate form from an input sentence and then generates the actual meaning representation based on the sentence embeddings and the intermediate embeddings. At inference, the model takes as input only an input sentence.

A new approach to semantic parsing is to incorporate the 'world knowledge' of meaning representation into semantic parsing. [20] present a pre-trained language model that jointly learns representations for sentences and (semi-)structured tables and a semantic parser based on the pre-trained model, where their goal is to generate e.g. SQL queries executable on the tables. We do not follow this approach since robot world states are dynamic and keep changing, unlike tables.

Another approach is 'interactive' semantic parser, getting feedback from user and updating semantic parsing outputs accordingly. [7] generate a SQL query and its text description and, if user gives a corrective instruction upon the text description and the SQL query's execution results, update the SQL query by incorporating the corrective instruction. [21] analyze a user instruction and, if it has any ambiguous phrase, ask the user a specific question for disambiguation. The proposed work is also interactive in that it checks if the intention translated from a given instruction is valid in given conditions of robot world states and, if not valid, gives feedback to the user for them to alternate the instruction.

3 Goal Translation

In this section, we discuss automatic generation of an intention from a given instruction sentence. We describe our in-house dataset for this task and the approach we have taken.

3.1 Utterance-Goal Annotation Dataset

Our dataset consists of 141 human utterances of instruction sentences for the robot to execute a subgoal in a gearbox assembly and disassembly scenario. Each intention string is translated into a sequence of tokens (words and symbols) consisting of three parts: Intention *type* (e.g. instruct achieve, instruct maintain), symbols describing the actions to perform as a function (e.g. fasten ?casing-top ?screw), and variable bindings or conditions (e.g. casing-base ?casing-base), which are given after the keyword 'given'. The dataset contains 45 unique verbs (e.g. insert, attach, install) and 32 unique object types (e.g. gearbox, input-shaft, screw). Table 1 shows example instruction sentences and their intentions.

Human utterance	**Translated intention**
Please begin assembly of the casing base	instruct achieve (assembled ?casing-base) given (casing-base ?casing-base)
Install the input shaft	instruct achieve (installed ?input-shaft ?object) given (input-shaft ?input-shaft)
Please attach the small hub cover onto the casing top next	instruct achieve (attached ?small-hub-cover ?casing-top) given (small-hub-cover ?small-hub-cover) (casing-top ?casing-top)
To start disassembly, put the gearbox in a vertical position	instruct achieve (vertical ?gearbox) given (gearbox ?gearbox)

Table 1. Examples of translated intentions in our utterance-goal dataset

3.2 Goal Translation System

We approach it as a sequence-to-sequence task, taking an instruction sentence as input and generating an intention string as a sequence of tokens (words and symbols). One advantage of the sequence-to-sequence approach is that we can employ pre-trained language models (e.g. GPT-2 [16], T5 [17]), which are trained with large collection of English texts by self-supervised learning methods (e.g. masked language modeling) and recently led to many breakthroughs in natural language processing including semantic parsing. The intention has structure, consisting of tuples and keywords, but the elements of the structure are written in English words and can thus be targeted for generation by fine-tuning the pre-trained language models.

The intention generation can be considered as a task of semantic parsing, in that it aims at generating the meaning representation of a given sentence from the viewpoint of motion planning. Therefore, we adapt several methods of semantic parsing for the task, including employing LSTM, the pre-trained language models and other methods (e.g. Coarse2Fine [6]). Technically, we used two layers of bi-LSTM and GloVe word embeddings [15] as inputs to the LSTM model (learning rate: 1e-3, number of epochs: 60, dropout rate: 0.1). We used the following hyper-parameters to fine-tune of the two pre-trained models: GPT-2 (learning rate: 5e-5, number of epochs: 150, dropout rate: 0.1) and T5 (learning rate: 3e-4, number of epochs: 50, dropout rate: 0.1).

For Coarse2Fine, it requires the output of semantic parsing to be of tree structure. We thus slightly modified the intention string as follows: 1) adding brackets to surround the whole intention string and 2) moving the "repeat all" keyword before the actions so that the keyword is the root of the sub-tree of actions. Table 2 shows examples of these modifications. We used the following hyper-parameters to train the Coarse2Fine parser[3]: learning rate 5e-3, number of epochs 100, dropout rate 0.5. Batch size is set as 16 for all the 4 models.

Changes	**Original intention**	**Changed intention**
Add brackets	instruct achieve (attached ?small_hub_cover ?casing_top) given (small_hub_cover ?small_hub_cover) (casing_top ?casing_top)	(instruct_achieve (attached (?small_hub_cover ?casing_top)) (given (small_hub_cover ?small_hub_cover) (casing_top ?casing_top)))
Move "repeat all" to the front	instruct achieve (checked ?bolt) given (bolt ?bolt) repeat all	(instruct_achieve (repeat-all ((checked ?bolt) (given (bolt ?bolt)))))

Table 2. Examples of changes made in intentions in coarse2fine

4 Reactive Execution for Translated Goals

The translated goals are used in our robot to execute procedures that achieve them in a reactive manner. We developed our robotic system in the context of a cognitive architecture, Icarus [4], that provides an infrastructure for cognitively-inspired intelligent capabilities on our robot. In this section, we first review Icarus briefly and then describe how the architecture processes the translated goals from the dialogue system, which employs the semantic parsers introduced in the next section, for reactive execution. Figure 1 illustrates the workflow of the system.

4.1 ICARUS Review

Research on cognitive architectures is inspired by psychological evidence for many aspects of human mind. They share a certain set of commitments they make about representation, memory, and processes that work over them. One such architecture, Icarus, assumes relational representation of knowledge, distinguishes long-term and short-term memories, and operates in recognize-act cycles as other architectures do. But Icarus also features a unique combination of its explicit commitment to hierarchical knowledge structures, the distinction of concepts, procedures, and goals, and its goal reasoning and teleoreactive execution.

[3] https://github.com/donglixp/coarse2fine

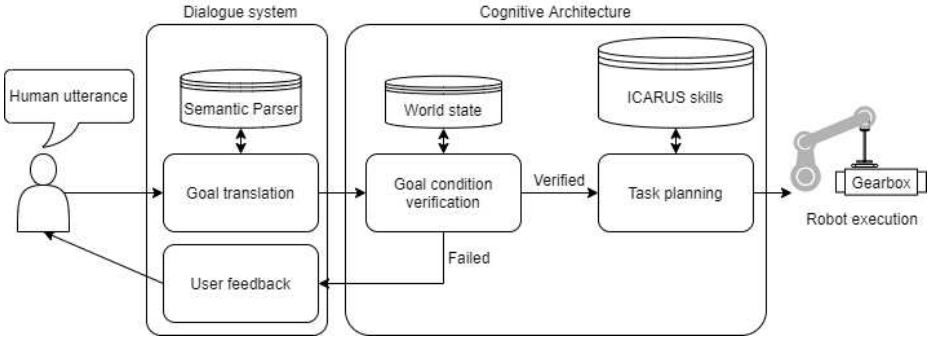

Fig. 1. Overview of the goal translation pipeline.

```
((holding ?hand ?obj)
 :elements ((hand ?hand *status ?gripper))
 :tests ((not (= ?gripper 'open))
         (not (= ?gripper 'closed))))
((output-insertion-complete ?hand ?output ?case)
 :elements ((output ?output)
            (inserted ?output ?case)
            (hand-empty ?hand)
            (in-high-pose ?hand)))
```

Table 3. Sample concepts of the ICARUS cognitive architecture

ICARUS uses *concepts* as its vocabulary to describe relations that hold true in the world. Table 3 shows some examples in the industrial manipulation domain we use. Concepts resemble Horn clauses [10] that include a head, a list of matching conditions, and optional tests against matched variables. The first concept, `holding`, matches against a manipulator hand and checks its status attribute to see if the hand is holding any object. Notice that this concept definition uses only perceptual matching against objects and their attributes, making this a *primitive* concept. The second concept, however, is a non-primitive one, and we can see that it refers to other concepts, `inserted`, `hand-empty`, and `in-high-pose`, in addition to perceptual matching against an `output` object. In this manner, ICARUS's concepts form a hierarchy of relations.

To describe procedures that achieve certain situations in the world, the architecture uses *skills*. Table 4 shows some sample skills from our industrial manipulation domain. We can consider skills to be a hierarchical version of STRIPS operators [8] with a head, a list of matching conditions, a list of direct actions or sub-skills, and a set of effects. The first skill, `insert-object`, is a primitive skill, in that it only refers to actions that can be executed directly in the world. It requires three objects (a `hand`, an `object`, and a `case`) and two concepts (`holding` and `in-insertable-pose`) as its precondition and, upon a successful completion, achieves the concept, `inserted`, in the world. In contrast, the second skill

```
((insert-object ?hand ?object ?target)
 :elements ((hand ?hand)
            (object ?object)
            (case ?target)
            (holding ?hand ?object)
            (in-insertable-pose ?hand ?object ?target))
 :actions  ((*move-in-z-until-contact ?hand))
 :effects  ((inserted ?object ?target)))
((insert-object ?hand ?object ?target)
 :elements ((hand ?hand)
            (object ?object)
            (case ?target))
 :subskills ((move-to-insertable-pose ?hand ?object ?target)
            (insert-object ?hand ?object ?target))
 :effects  ((inserted ?object ?target)))
```

Table 4. Sample skills of the ICARUS cognitive architecture

is a non-primitive skill, which refers to other skills, `move-to-insertable-pose` and `insert-object` (the first example), in order to achieve its effects.

Using concepts and skills like the ones we have seen so far, the ICARUS architecture is able to infer the current situation of the world based on the sensory input and make decisions to execute a certain skill at each given time. ICARUS's execution of skills is governed by its goals, which are general descriptions of desired situations written as concept instances with their associated relevance conditions. For example, the architecture will decide to execute two different skills even under two exactly same situations, given two different top-level goals. Due to the limited space available here, we will refer curious readers to our previous work for more detailed review of ICARUS and various processes it employs. We will discuss its goal-oriented but reactive execution in the next section, while we describe how the architecture uses the translated goals from the dialogue system for execution.

4.2 Reactive Execution of Translated Goals

When the user generates utterances, our system's goal translation module takes and translates them into a format our cognitive architecture can understand. First, our system parses the translations into the description of goals and the conditions that should be met. ICARUS compiles these into a top-level goal, which its execution module can readily take and process. During this process, it also looks up the words used in the goals and conditions against its linguistic domain knowledge base to disambiguate the meaning. For example, the system replaces the verbs, `place`, `put`, and `install` with `install`, which ICARUS's skills are written with. Then it checks whether the translated goal exists in its concept definitions and verifies that the conditions hold in the current world state. If either fails, the user is prompted to modify their instruction by using

more specific descriptions. Given a valid translation of goals and conditions, ICARUS generates a task plan to achieve the given goal. The plan is a set of skill instances at multiple levels of abstraction, grounded at the robot's low-level actions. During action execution, the architecture consistently checks for new user utterances, as well as changes in situation. This enables the user to interrupt the execution anytime by defining a new goal. Table 5 shows an example dialogue between human user and robot, where the user interrupts robot.

Human:	Assemble the gearbox.
Robot:	I do not recognize that goal. Please give more detailed instruction.
Human:	Insert the input subassembly into the casing base.
Robot:	I will work on inserting input subassembly into case.
	(Robot starts to move to pick up the object.)
Human:	Actually, insert the output subassembly into the casing base first.
Robot:	I will work on inserting output subassembly into case.
	(Robot works to pick up the new object.)

Table 5. Example dialogue between human user and robot, with a user interruption

5 Evaluation of Goal Translation Methods

We evaluated the four methods of goal translation against our utterance-goal annotation dataset, randomly splitting the dataset into 90% for training and 10% for testing and reporting the average performance of the methods across 5 random splits. Table 6 shows the evaluation results of the methods against the dataset. The *accuracy* measure indicates if the whole string of generated intention is correct or not. As a strict measure, the accuracies of the methods are low due to the small size of the dataset. We also introduce an *F1-score* that measures how many correct concepts the generated intentions contain. The two measures indicate how easily a user can select or write the correct intention based on the top results of a method.

Method	Accuracy	F1
LSTM	28.0%	68.0%
Coarse2Fine	34.7%	75.5%
GPT-2	32.0%	76.1%
T5	**46.7%**	**84.1%**

Table 6. Evaluation results of semantic parsing methods for intention generation. 'F1' indicates F1-score.

The evaluation results summarized in Table 6 show that pre-trained models (e.g. GPT-2, T5) outperform non-pre-trained models (e.g. LSTM) even though

the LSTM model utilizes pre-trained word embeddings (GloVe). This can be possibly due to the small size of the dataset, which is not enough to optimize randomly initialized parameters of the LSTM model. T5 is reported to show better performance also on other NLP applications than GPT-2 [17].

Coarse2Fine achieves significantly higher performance than LSTM, but significantly lower performance than T5. Coarse2Fine is better than LSTM probably because Coarse2Fine adds a sketch layer between the LSTM encoder and the decoder in order to first learn an intermediate representation of intention and then to generate the intention string, and applies parent feeding.

The pre-trained model T5 is better than Coarse2Fine, while GPT-2 is slightly worse in terms of accuracy but slightly better in terms of F1 score than Coarse2Fine. The higher accuracy yet lower F1 score of Coarse2Fine might be explained by the sketch layer, which guides Coarse2Fine to first form the whole sketch of intention before generating the full string of intention. The mixed results of comparison between the pre-trained models and the LSTM-based sophisticated semantic parsing method may result from differences between the two pre-trained models such that T5 is pre-trained with much bigger data (7 TB) than GPT-2 (40 GB), and that T5 is an encoder-decoder model, while GPT-2 has a decoder only. But, we cannot conclude that the bigger pre-trained language model shows the higher performance on the intention generation task, since we have not compared with other pre-trained language models, and leave it as a future work to understand why a certain pre-trained model is better than others for the goal translation task.

6 Future Work

The current work provides a good foundation for interactive robotic systems. But we need many additional capabilities to build a robot that humans feel more natural to work with across different domains. For instance, the system needs to have the capacity to translate a broader range of utterances, potentially using common sense knowledge to understand a variety of ways to say same things. We plan to address variations in verbs and nouns, leveraging SenticNet [2], a common sense knowledge base. The knowledge base describes the meaning of such words using their corresponding primitive words, and this enables us to replace new words with known synonyms that the Icarus architecture knows how to act upon.

In addition, we plan to extend our system with an interactive learning capability. This will allow human users to teach new concepts and skills verbally, eliminating the need for manual encoding of such knowledge. Interactive learning can occur when the user specifically asks for it, or when the system encounters words in user utterance that do not currently map to internal goals. Some previous work on cognitive architectures [13] support this functionality in various ways, and we will use those as our reference.

7 Conclusions

We leverage recent and well-known semantic parsing models and adapted them to the intention translation task by fine-tuning the models with the small-sized dataset of our target domain. The intention translation is used in combination with a cognitive architecture, ICARUS [4], to allow the human operator to issue high-level verbal commands to an industrial robot. Using the translated goal and cognitive architecture, the robot generates and executes a task plan. Finally, we evaluated four semantic parsing methods on our small-sized utterance-goal dataset and showed that the pre-trained model T5 [17] outperforms the other methods. Future work will address investigating why some pre-trained models perform better than others on this task.

References

1. Artzi, Y., Das, D., Petrov, S.: Learning compact lexicons for CCG semantic parsing. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 1273–1283 (2014)
2. Cambria, E., Poria, S., Hazarika, D., Kwok, K.: SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1795–1802 (2018)
3. Chen, H., Tan, H., Kuntz, A., Bansal, M., Alterovitz, R.: Enabling robots to understand incomplete natural language instructions using commonsense reasoning. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 1963–1969 (2020)
4. Choi, D., Langley, P.: Evolution of the ICARUS cognitive architecture. Cognitive Systems Research **48**, 25–38 (2018)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–418 (2019)
6. Dong, L., Lapata, M.: Coarse-to-fine decoding for neural semantic parsing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. pp. 731–742 (2018)
7. Elgohary, A., Hosseini, S., Awadallah, A.H.: Speak to your parser: Interactive text-to-SQL with natural language feedback. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2065–2077 (2020)
8. Fikes, R., Nilsson, N.: STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence **2**, 189–208 (1971)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
10. Horn, A.: On sentences which are true of direct unions of algebras. Journal of Symbolic Logic **16**, 14–21 (1951)
11. Jia, Y., She, L., Cheng, Y., Bao, J., Chai, J.Y., Xi, N.: Program robots manufacturing tasks by natural language instructions. In: Proceedings of the IEEE International Conference on Automation Science and Engineering. pp. 633–638 (2016)

12. Kuo, Y.L., Katz, B., Barbu, A.: Deep compositional robotic planners that follow natural language commands. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 4906–4912 (2020)
13. Laird, J.E., Gluck, K., Anderson, J., Forbus, K.D., Jenkins, O.C., Lebiere, C., Salvucci, D., Scheutz, M., Thomaz, A., Trafton, G., Wray, R.E., Mohan, S., Kirk, J.R.: Interactive task learning. IEEE Intelligent Systems **32**(4), 6–21 (2017)
14. Park, J.S., Jia, B., Bansal, M., Manocha, D.: Efficient generation of motion plans from attribute-based natural language instructions using dynamic constraint mapping. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 6964–6971 (2019)
15. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 1532–1543 (2014)
16. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**, 9 (2019)
17. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020)
18. Venkatesh, S.G., Biswas, A., Upadrashta, R., Srinivasan, V., Talukdar, P., Amrutur, B.: Spatial reasoning from natural language instructions for robot manipulation. In: Proceedings of the IEEE International Conference on Robotics and Automation (2021)
19. Wächter, M., Ovchinnikova, E., Wittenbeck, V., Kaiser, P., Szedmak, S., Mustafa, W., Kraft, D., Krüger, N., Piater, J., Asfour, T.: Integrating multi-purpose natural language understanding, robot's memory, and symbolic planning for task execution in humanoid robots. Robotics and Autonomous Systems **99**, 148–165 (2018)
20. Yin, P., Neubig, G., Yih, W.T., Riedel, S.: 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 (2020)
21. Zeng, J., Victoria, X., Caiming, L., Richard, X., Lyu, M.R., King, I., Hoi, S.C.H.: Photon: A robust cross-domain text-to-SQL system. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 204–214 (2020)