

# Parsing Natural Language into Propositional and First-Order Logic with Dual Reinforcement Learning

Xuantao Lu<sup>♣</sup>, Jingping Liu<sup>♣\*</sup>, Zhouhong Gu<sup>♣</sup>, Hanwen Tong<sup>♣</sup>,

Chenhao Xie<sup>★</sup>, Junyang Huang<sup>♣</sup>, Yanghua Xiao<sup>♣♡\*</sup>, Wenguang Wang<sup>◇</sup>,

<sup>♣</sup>Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University

<sup>★</sup>SenseDeal Intelligent Technology Co., Ltd., Beijing, China

<sup>♠</sup>School of Information Science and Engineering, East China University of Science and Technology

<sup>♡</sup>Fudan-Aishu Cognitive Intelligence Joint Research Center <sup>◇</sup>DataGrand Inc., Shanghai, China

{xtlu20, shawyh}@fudan.edu.cn, jingpingliu@ecust.edu.cn,

xiechenhao@senseddeal.ai

## Abstract

Semantic parsing converts natural language utterances into structured logical expressions. We consider two such formal representations: Propositional Logic (PL) and First-order Logic (FOL). The paucity of labeled data is a major challenge in this field. In previous works, dual reinforcement learning has been proposed as an approach to reduce dependence on labeled data. However, this method has the following limitations: 1) The reward needs to be set manually and is not applicable to all kinds of logical expressions. 2) The training process easily collapses when models are trained with only the reward from dual reinforcement learning. In this paper, we propose a scoring model to automatically learn a model-based reward, and an effective training strategy based on curriculum learning is further proposed to stabilize the training process. In addition to the technical contribution, a Chinese-PL/FOL dataset is constructed to compensate for the paucity of labeled data in this field. Experimental results show that the proposed method outperforms competitors on several datasets. Furthermore, by introducing PL/FOL generated by our model, the performance of existing Natural Language Inference (NLI) models is further enhanced.

## 1 Introduction

Semantic parsing is the task of mapping natural language utterances into logical expressions. As two major logical forms of text representation, Propositional Logic (PL) and First-order Logic (FOL) play an increasingly important role in a wide range of downstream tasks including Inductive Logic Programming (ILP) (Yang and Song, 2019), Question Answering (QA) (Longo and Santoro, 2020) and Interpretable Reinforcement Learning (Ma et al., 2020; Kimura et al., 2021) because they are capable

\* Corresponding Author

English-FOL
<b>Natural Language Utterance</b> Some volunteers include executives and professionals.
<b>Logical Expression</b> <code>exists x1.(_volunteer(x1) &amp; exists x2. (_executive(x2) &amp; _professional(x2) &amp; _include(x1, x2)))</code>

Table 1: Example of the English-FOL dataset. Text in **olive** denotes quantifiers, and text in **teal** denotes predicates.  $x1$  and  $x2$  denote variables, and  $\&$  denotes logical connectives.

of discovering and representing knowledge in an explicit symbolic structure that can be understood and examined by human (Evans and Grefenstette, 2018). It is worth noting that all of these tasks have a prerequisite, i.e., parsing natural language utterances into PL/FOL. The results of parsing directly affect the performance of downstream tasks. Thus, it is crucial to have a strong semantic parser for PL/FOL.

Some solutions have been proposed to parse natural language into PL/FOL. One of the most typical methods is to model the parsing task as a sequence-to-sequence (Seq2seq) generation problem, including using a character-level recurrent neural network (Levkovsky and Li, 2021) and introducing a variable alignment mechanism (Singh et al., 2020). However, these approaches have the following issues. First, a large amount of labeled data is required for these approaches to achieve good results, which inevitably suffers from the paucity of labeled data in this field. One of the solutions is to generate labeled data by templates (Levkovsky and Li, 2021), but it leads to a lack of diversity in the data which makes the model prone to overfitting the training data. Second, previous works (Singh et al., 2020; Levkovsky and Li, 2021) only consider unidirectional generation (from utterance to PL/FOL), while intuitively bidirectional generation can further enhance the performance of the models.

Regarding these issues, we propose an effective framework for parsing natural language into PL/FOL named **Dual-(m)T5**, and introduce unlabeled data to alleviate the impact of insufficient labeled data. Inspired by He et al. (2016); Cao et al. (2019), we model the learning of logical expressions and natural language generation as dual tasks. Both of the tasks are jointly trained via reinforcement learning (RL) since the training process is non-differentiable. However, we encounter the following challenges when applying dual reinforcement learning to PL/FOL: 1) The validity reward in dual reinforcement learning is rule-based (Cao et al., 2019), which results in a lot of manual attempts to get an effective one, and needs to be redesigned when the type of logical expressions changes (e.g. from lambda-calculus to FOL) since it is customized for a specific type of logical expression. 2) The effectiveness of the validity reward needs to be improved because it only considers a lexical-level matching between utterances and logical expressions, and cannot achieve deep semantic matching. 3) An effective training strategy needs to be explored since the training process easily collapses when the model is trained with only the rewards from dual reinforcement learning.

To address the first two issues mentioned above, we propose a scoring model to automatically learn a model-based validity reward that is applicable to various types of logical expressions. The scoring model aims to evaluate whether the semantics of utterances and logical expressions match. For issue 3, we propose an effective training strategy to stabilize the training process. Specifically, curriculum learning (Bengio et al., 2009) is employed to initialize the parameters of the model, aiming to get the model in a good initial state, and unlabeled data is introduced to prevent models from crashes due to insufficient labeled data. Experimental results on different datasets show that our framework effectively parses natural language into PL/FOL and consistently improves performance compared to competitors. To further demonstrate the value of PL/FOL, we take Natural Language Inference (NLI) as a downstream task, and the performance of existing NLI models is further enhanced by introducing PL/FOL generated by our model.

Our contributions are three-folds:

- We propose a dual reinforcement learning framework called **Dual-(m)T5** with a novel model-based validity reward that fully con-

siders the semantics of utterances and logical expressions and is applicable to all types of logical expressions. We further propose an effective training strategy to stabilize the training process of dual reinforcement learning.

- We release a new dataset called **Chinese-PL/FOL** that contains 1,263 Chinese-PL pairs and 1,464 Chinese-FOL pairs to compensate for the paucity of labeled data in this field.
- Experimental results show that the proposed method outperforms competitors on several datasets. Furthermore, by introducing additional logical expressions generated by **Dual-(m)T5**, the performance of existing NLI models is further enhanced.

## 2 Overview

In this section, we formalize the problem and outline our framework.

### 2.1 Problem Definition

As shown in Table 1, given a natural language sentence  $s$ , the goal of this paper is to generate the corresponding logical expression  $e$ . For example, given a sentence “Some volunteers include executives and professionals.”, an ideal model would generate a logical expression “exists x1.(\_volunteer(x1) & exists x2. (\_executive(x2) & \_professional(x2) & \_include(x1, x2)))”.

### 2.2 Framework

The overview of our framework is shown in Figure 1. The backbone of the framework is dual reinforcement learning, which consists of two sub-modules: The prime module generates a logical expression given a natural language sentence, while the dual module produces a sentence given a logical expression. The scoring model is used to obtain the validity reward, and the reconstruction reward is used to force the generated sentence in the dual module as similar to the original sentence as possible. To stabilize the training process, models are pre-trained before dual reinforcement learning, and curriculum learning is employed to get the model in a good initial state.

## 3 Methodology

In this section, we first present the details of dual reinforcement learning in § 3.1. The scoring model

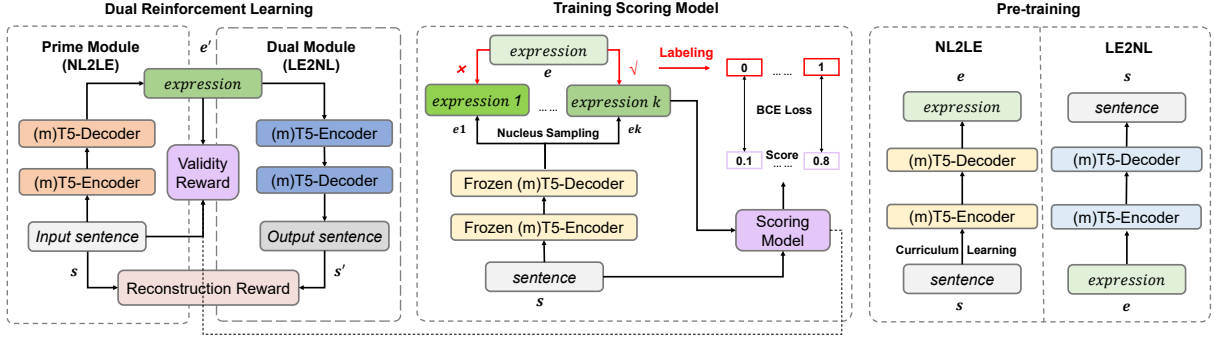


Figure 1: An overview of the framework. Dual reinforcement learning consists of NL2LE and LE2NL, and the validity reward is learned by the scoring model. The (m)T5s used in the scoring model and dual reinforcement learning are initialized by pre-training. The sentence  $s$  and expression  $e$  represent a ground truth pair.

is introduced in § 3.2, and the training strategy is provided in § 3.3.

### 3.1 Backbone: Dual Reinforcement Learning

The backbone of our framework is dual reinforcement learning which consists of two sub-modules: Natural Language to Logical Expression (NL2LE) and Logical Expression to Natural Language (LE2NL). Both of the modules adopt T5 (Raffel et al., 2019) / mT5 (Xue et al., 2021), a (multi-lingual) pre-trained text-to-text transformer as the backbone. These two modules in a closed-loop are trained by a reinforcement learning (RL) method based on policy gradient (Sutton et al., 2000). In RL, the state is denoted by the input of the prime module, i.e., sentence  $s$ . The action in the prime and dual modules is defined as the logical expression and sentence generation, respectively. The policy is denoted as the parameters of the (m)T5 models in the two modules.

**Prime Module (NL2LE)** aims to transform natural language into PL/FOL. Specifically, given a sentence  $s$ , the NL2LE model could generate  $k$  possible logical expressions  $e_1, e_2, \dots, e_k$  via nucleus sampling (Holtzman et al., 2020). Then, the scoring model scores the generated logical expressions and obtains a validity reward  $R_{val}(e_i | s)$  for each logical expression  $e_i$ . The details of the scoring model will be introduced in § 3.2.

**Dual Module (LE2NL)** is an inverse of the prime module, which aims to generate sentences given PL/FOL. Formally, the input is the logical expression  $e_i$  generated in the prime task, and the model is expected to output the original sentence  $s$ . Reconstruction reward is used to estimate the similarity between the input of the prime model and the output of the dual model. Let  $\Theta_{NL2LE}$  and  $\Theta_{LE2NL}$

denote all the parameters of NL2LE and LE2NL, respectively. The reconstruction reward is formulated as:

$$R_{rec}(s | e_i) = \log P(s | e_i; \Theta_{LE2NL}) \quad (1)$$

**Learning Algorithm** By utilizing policy gradient (Sutton et al., 2000), the stochastic gradients of  $\Theta_{NL2LE}$  and  $\Theta_{LE2NL}$  are computed as:

$$\nabla_{\Theta_{NL2LE}} \mathbb{E}[r] = \frac{1}{k} \sum_{i=1}^k r_i \cdot g_i \quad (2)$$

$$r_i = \alpha R_{val}(e_i | s) + (1 - \alpha) R_{rec}(s | e_i) \quad (3)$$

$$g_i = \nabla_{\Theta_{NL2LE}} \log P(e_i | s; \Theta_{NL2LE}) \quad (4)$$

$$\nabla_{\Theta_{LE2NL}} \mathbb{E}[r] = \frac{1 - \alpha}{k} \sum_{i=1}^k g'_i \quad (5)$$

$$g'_i = \nabla_{\Theta_{LE2NL}} \log P(s | e_i; \Theta_{LE2NL}) \quad (6)$$

where a hyper-parameter  $\alpha \in [0, 1]$  is exploited to balance between  $R_{val}$  and  $R_{rec}$ .

### 3.2 Scoring Model

The scoring model is used to evaluate whether the semantics of the generated logical expression is consistent with the semantics of the input sentence and calculate the validity reward  $R_{val}$ . Formally, given a  $\langle s, e \rangle$  pair, the scoring model outputs  $P(e | s) \in [0, 1]$ , which represents the correlation between  $s$  and  $e$ . Intuitively, we take  $\{\langle s_i, e_i \rangle\}$  pairs from the supervised dataset  $\mathcal{L}$  as positive samples  $\mathcal{P}$  to train such a scoring model. The challenge is that off-the-shelf negative samples are not available. Negative sampling, i.e., sampling  $\{\langle s_i, e_j \rangle\}_{i \neq j}$  pairs from  $\mathcal{L}$  as negative samples is an optional solution, but the quality of negative samples obtained in this way are not challenging

for the scoring model. To get enough hard negative samples to train the scoring model, we design the following approach:

First, an NL2LE model is pre-trained with the supervised dataset  $\mathcal{L}$  (details in § 3.3.1). After that, given a sentence  $s$ , the NL2LE model could generate  $k$  possible logical expressions  $\bar{e}_1, \bar{e}_2, \dots, \bar{e}_k$  via nucleus sampling (Holtzman et al., 2020). We denote  $\{\langle s, \bar{e}_i \rangle\}$  as negative samples  $\mathcal{N}$ , where the logical expression  $\bar{e}_i$  is not equal to ground truth  $e$ . Since the NL2LE model has been pre-trained,  $\bar{e}_i$  will be similar to the ground truth  $e$ . These hard negative samples will challenge the scoring model and enable it to learn the effects of small differences in logical expressions. For each  $\langle s_i, e_i \rangle \in \mathcal{P} \cup \mathcal{N}$ , we take the last layer hidden states of the NL2LE model’s encoder  $h_{i1}^{enc}, \dots, h_{in}^{enc}$  and decoder  $h_{i1}^{dec}, \dots, h_{im}^{dec}$  as the feature of  $s_i$  and  $e_i$  respectively. Then, the scoring model is defined as follows:

$$\bar{h}_i^{enc} = \frac{1}{n} \sum_{j=1}^n h_{ij}^{enc}, \quad \bar{h}_i^{dec} = \frac{1}{m} \sum_{j=1}^m h_{ij}^{dec} \quad (7)$$

$$u_i = \bar{h}_i^{enc} \cdot W_1 + b_1, \quad v_i = \bar{h}_i^{dec} \cdot W_2 + b_2 \quad (8)$$

$$P(e_i | s_i) = \sigma([u_i; v_i; |u_i - v_i|] \cdot W_3 + b_3) \quad (9)$$

where  $W_{1|2|3}$  and  $b_{1|2|3}$  are trainable parameters.  $[\cdot; \cdot]$  is the concatenation operation, and  $\sigma$  represents sigmoid function. The training loss  $L$  of the scoring model is binary cross-entropy (BCE) loss between the model’s output  $P(e_i | s_i)$  and labels,

$$L = - \frac{1}{|\mathcal{P} \cup \mathcal{N}|} \left( \sum_{\langle s_i, e_i \rangle \in \mathcal{P}} \log P(e_i | s_i) + \sum_{\langle s_i, e_i \rangle \in \mathcal{N}} (1 - \log P(e_i | s_i)) \right) \quad (10)$$

Note that the parameters of the NL2LE model are fixed, and only the scoring model is updated during the backpropagation. Finally, we take  $P(e_i | s)$  as the validity reward  $R_{val}(e_i | s)$ .

### 3.3 Training Strategy

In this section, we will introduce the training strategy of our framework. The entire training process consists of two stages: pre-training and dual reinforcement learning. For pre-training, we explore how to integrate curriculum learning into the training phase in § 3.3.1. For dual reinforcement learning, we explore how to construct and introduce

unlabeled data in § 3.3.2, and make the training stable in § 3.3.3.

#### 3.3.1 Pre-training with Curriculum Learning

The pre-training of the NL2LE model aims to maximize the likelihood  $p(e|s)$  for each  $\langle s, e \rangle$  pair from the supervised dataset  $\mathcal{L}$ . According to the learning principle of human beings in the cognitive process, we should start with simple samples and gradually consider more complex samples. To this end, we employ curriculum learning (Bengio et al., 2009) to determine the training order. Here, we take the length of logic expressions as an indicator of the training order, i.e., the longer the logical expression, the more difficult it is. We first sort the training samples according to the length of the logical expressions. At each training step  $t$ , a batch of training samples is obtained from the top  $f(t)$  portions of the entire sorted training samples. Following Platanios et al. (2019),  $f(t)$  is defined as:

$$f(t) = \min \left( 1, \sqrt{\frac{t(1-c_0^2)}{T} + c_0^2} \right) \quad (11)$$

where  $c_0$  represents the models start training using the  $c_0\%$  easiest training samples, and  $T$  represents the duration of curriculum learning.<sup>1</sup>

#### 3.3.2 Introducing Unlabeled Data for Dual Reinforcement Learning

Since the training process of dual reinforcement learning only leverages natural language utterance and does not need the corresponding logical expression, in addition to using utterances from the supervised dataset  $\mathcal{L}$ , we further improve the performance of the models by introducing unlabeled utterances  $\mathcal{U}$ .

Different from the previous work (Cao et al., 2019) where unlabeled data is constructed by manually defined rules, we leverage off-the-shelf paraphrase generation models (see in Appendix.D) which generate synonymous sentences from existing utterances in the supervised dataset  $\mathcal{L}$ . According to our observation, since the paraphrase generation models are trained on paraphrase generation datasets that are different from the datasets we use, the generated synonymous sentence is not particularly similar to the original one, and the logical expressions corresponding to the two sentences are different in most cases. Thus, it is reasonable

<sup>1</sup>In practice, curriculum learning has no effect on LE2NL, so we only apply it to NL2LE.



	# PL	# FOL	TOTAL
<b>Training</b>	871	1,037	1,908
<b>Validation</b>	128	145	273
<b>Test</b>	264	282	546
<b>Total</b>	1,263	1,464	2,727

Table 2: Statistics of the Chinese-PL/FOL dataset.

to treat the generated sentences as unlabeled data. The experiments in § 5 also demonstrate the effectiveness of this method.

### 3.3.3 Stable Dual Reinforcement Learning

In practice, we find that the training process easily collapses when the models are trained with only the rewards from dual reinforcement learning. To keep the training stable and prevent the models from crashing, we adopt the following method:

**Introducing Supervisor** We pre-train both of the models with the supervised dataset  $\mathcal{L}$  before dual reinforcement learning starts (the pre-training of NL2LE refers to § 3.3.1). Moreover, after each update according to Eq.(2) and Eq.(5), the models are trained with the labeled data again, i.e., both of the models are trained with dual reinforcement learning and supervised learning alternately.

**Reward Baseline** To cope with high variance in reward signals, we generate  $k$  intermediate outputs as mentioned in § 3.1 and re-define reward signals by introducing a reward baseline to stabilize the training process. Here, we take the average of rewards within samples per input as the reward baseline. Thus, the final validity reward  $R'_{val}$  and reconstruction reward  $R'_{rec}$  are as follows:

$$R'_{val}(e_i | s) = R_{val}(e_i | s) - \frac{1}{k} \sum_{i=1}^k R_{val}(e_i | s)$$

$$R'_{rec}(s | e_i) = R_{rec}(s | e_i) - \frac{1}{k} \sum_{i=1}^k R_{rec}(s | e_i)$$

## 4 Dataset Collection

To compensate for the paucity of labeled data in this field and verify the effectiveness of our proposed framework, we construct a dataset containing natural language and PL/FOL pairs. In the previous work (Levkovskiy and Li, 2021), the authors define templates first and then obtain samples by filling slots. However, the resulting dataset is limited by the lack of diversity of templates. Crowdsourcing is another option but is not applicable for this task, since professional knowledge about PL/FOL

is required. Therefore, we use expert annotation to ensure the quality and diversity of the dataset.

The annotation team consists of 8 Chinese graduate students who are familiar with PL/FOL. If the annotators are required to construct data without any reference, this will introduce inevitable troubles and labeling errors, since PL/FOL is not intuitive to humans. Aiming to reduce nontrivial human labor and ensure the quality of the dataset, the data collection process consists of the following steps: We first obtain PL/FOL exercise sets and exam papers that require students to convert natural language into PL/FOL from Baidu Wenku<sup>2</sup>. Then, the annotators are asked to organize these exercises in a uniform format. Each sample consists of three parts: natural language sentence  $s$ , symbolic definition  $d$ , and logical expression  $e$  (see in Table 7). This is slightly different from the English-FOL dataset where the symbolic definition is not included. We believe that the introduction of symbolic definition is beneficial because it helps to reach agreement among annotators. After that, the annotators are encouraged to rewrite existing data to obtain more challenging data, i.e., some samples have only slight differences in utterances, but their corresponding logical expressions are totally different.

In this way, we obtain a total of 2,727 samples consisting of 1,263 PL and 1,464 FOL with the corresponding utterances and symbolic definitions. To establish human performance and conduct consistency assessments, we ask an additional 3 undergraduate and 2 graduate students who have acquired basic knowledge of PL/FOL to provide logical expressions given natural language sentences and symbolic definitions from the entire test set. The detailed statistics of the dataset are shown in Table 2.

## 5 Experiments

In the experimental section, we investigate the following research questions: 1) How is the overall performance of Dual-(m)T5 in comparison to competitors? 2) How does Dual-(m)T5 perform on other types of logical expressions? 3) What is the optimal ratio of unlabeled data to labeled data? 4) Are all the components in Dual-(m)T5 necessary? 5) Which samples are not yet well processed by the model? 6) Can the logical expressions generated by Dual-(m)T5 help downstream tasks?

<sup>2</sup>One of the largest online platforms for sharing documents.

## 5.1 Setup

**Datasets.** 1) **English-FOL** (Levkovskyi and Li, 2021) is generated by pre-defined templates and contains natural language utterances paired with FOL. We follow the training/validation/test splits as Levkovskyi and Li (2021). 2) **Chinese-PL/FOL** The details of our dataset have been introduced in § 4. Since symbol definition only appears in this dataset, we concatenate it with the original input, i.e., natural language sentence in the prime module and logical expression in the dual module. Due to the small amount of training data, PL and FOL are trained together. 3) **ATIS** (Dahl et al., 1994) consists of queries about flight information and logical expressions in lambda-calculus syntax. For fairness in model comparison, we keep the same preprocessing settings as (Dong and Lapata, 2018; Cao et al., 2019).

**Baselines.** To verify the effectiveness of our approach on English-FOL and Chinese-PL/FOL, we reproduce several strong baselines as there are not many existing works on these two datasets. For ATIS, we directly compare our method with state-of-the-art works. Refer to Appendix.C for details.

**Metrics.** We follow the previous work (Levkovskyi and Li, 2021) and take Exact Match (EM) as the evaluation metric.

## 5.2 Overall Results

We compare our method with competitors on different datasets in Table 3. From the results, we conclude that: **1)** Our models outperform the competitors on both of the datasets, and is not affected by language, which shows the effectiveness and robustness of our models. **2)** Even without additional unlabeled data, our models outperform the competitors only with the labeled data, which indicates that our approach is also available in scenarios without unlabeled data. **3)** By introducing the unlabeled data, the performance of the models is further improved, and the improvement on Chinese-PL/FOL is more obvious than that on English-FOL as the amount of the labeled data in English-FOL is enough for the training and the performance is hardly improved by using the unlabeled data, while the unlabeled data can be used to compensate for the paucity of the labeled data on Chinese-PL/FOL.

## 5.3 Generalization on Lambda-Calculus

To verify that our framework is still effective on other types of logical expressions, we compare our

method with previous works on ATIS without using unlabeled data in fairness. From the results shown in Table 4, we see that our method has better performance over the previous works on ATIS, which demonstrates the generality of our approach to other types of logical expressions. This is mainly attributed to the model-based validity reward that is not limited to the specific form of the logical expression, and our framework is suitable for any kind of logical expression without modification.

## 5.4 Experiments on Semi-supervised Setting

To investigate whether unlabeled data benefits the framework and the optimal ratio of unlabeled data to labeled data, we keep a part of the training set as fully labeled data and leave the rest as unlabeled data where only utterances are used. We change the ratio of unlabeled data to labeled data, and the results on English-FOL are shown in Table 5. The results show that the performance of the models does not improve constantly when the amount of unlabeled data is increased. We conclude that a proper ratio of unlabeled data is crucial and it is related to the number of parameters in the model. A model with more parameters tends to perform better with more unlabeled data. On the contrary, when a model with few parameters is trained with a large amount of unlabeled data by dual reinforcement learning, it may converge to a wrong equilibrium state to adapt to the unlabeled data and forget what has been learned from the labeled data, which leads to poor performance.

## 5.5 Ablation Study

To evaluate the effectiveness of each component in Dual-(m)T5, we perform an ablation analysis on English-FOL. From the results shown in Table 6, we conclude that: **1)** Dual reinforcement learning without the validity reward even gets worse results than the T5 baseline, which indicates that the validity reward is critical and indispensable in dual reinforcement learning. **2)** Curriculum learning improves EM but not significantly. Therefore, the value of curriculum learning is mainly to stabilize the training process rather than improve the performance. **3)** The model-based validity reward in dual reinforcement learning has certain advantages over the rule-based validity reward (Cao et al., 2019), which indicates the effectiveness of our approach.

Method	English-FOL	Chinese-PL/FOL		
		PL	FOL	TOTAL
Human Performance	-	87.94	79.92	84.07
ATT (Luong et al., 2015)	65.70	39.39	36.88	38.10
GPT-2 (Radford et al., 2019)	85.32	56.04	61.74	58.97
GPT-2-large (Radford et al., 2019)	90.03	-	-	-
Text2log (Levkovskiy and Li, 2021)	89.54	-	-	-
(m)T5-small (Raffel et al., 2019; Xue et al., 2021)	89.95	64.02	58.87	61.35
(m)T5-base (Raffel et al., 2019; Xue et al., 2021)	91.30	70.08	61.35	65.57
Dual-(m)T5-small ( <b>Ours</b> )	90.98	64.77±0.47	61.70±0.58	63.19±0.52
+ generated unlabeled data	91.06	70.83±0.78	63.83±0.75	67.22±0.75
Dual-(m)T5-base ( <b>Ours</b> )	92.65	70.45±0.31	63.12±0.29	66.67±0.15
+ generated unlabeled data	<b>92.83</b>	<b>75.00±0.94</b>	<b>68.44±0.73</b>	<b>71.61±0.83</b>

Table 3: EM on the test set of English-FOL and Chinese-PL/FOL. (Dual-)T5 and (Dual-)mT5 are used on English-FOL and Chinese-PL/FOL, respectively. Generated unlabeled data represents the unlabeled data obtained by paraphrase generation models in § 3.3.2.

Method	EM
TISP (Zhao and Huang, 2015)	84.2
Seq2tree (Dong and Lapata, 2016)	84.6
ASN+SUPATT (Rabinovich et al., 2017)	85.9
Tranx (Yin and Neubig, 2018)	86.2
Coarse2fine (Dong and Lapata, 2018)	87.7
Transformer (Ge et al., 2019)	87.7
ATTPTR + Dual (Cao et al., 2019)	88.6
TreeGen (Sun et al., 2020)	89.1
Dual-T5-base ( <b>Ours</b> )	<b>89.5</b>

Table 4: EM on the test set of ATIS.

Method	Labeled:Unlabeled	EM
Dual-T5-small	-	78.32
+ unlabeled data	<b>1:1</b>	<b>79.34</b>
+ unlabeled data	1:2	79.27
+ unlabeled data	1:3	78.37
+ unlabeled data	1:4	77.93
Dual-T5-base	-	87.01
+ unlabeled data	1:1	88.94
+ unlabeled data	1:2	89.22
+ unlabeled data	<b>1:3</b>	<b>89.34</b>
+ unlabeled data	1:4	88.76

Table 5: EM on the test set of English-FOL. It fixes the number of labeled samples (20% of the training set) and varies the ratio of unlabeled data to labeled data.

## 5.6 Error Analysis

For error analysis, we present three typical bad cases of Chinese-PL/FOL in Table 7. In Case 1, the model inverts cause and effect, indicating that the ability of causal reasoning needs to be enhanced. Case 2 requires the model to have the capability of coreference resolution, while Dual-(m)T5 does not have this ability yet. Case 3 shows that some errors are due to one utterance may correspond to multiple correct logical expressions, while there is only one annotated ground truth<sup>3</sup>. Such a problem

<sup>3</sup>In Case3, the ground truth is closer to the meaning of the utterance than the prediction, but the latter is an equivalent representation of the ground truth.

Method	EM
T5-small	89.95
w/ curriculum	90.09
w/ dual (w/o $R_e^{val}$ , w/ $R_s^{rec}$ )	89.23
w/ dual (w/ rule-based $R_e^{val}$ , w/ $R_s^{rec}$ )	90.44
w/ dual (w/ model-based $R_e^{val}$ , w/ $R_s^{rec}$ )	<b>90.88</b>
T5-base	91.30
w/ curriculum	91.41
w/ dual (w/o $R_e^{val}$ , w/ $R_s^{rec}$ )	90.87
w/ dual (w/ rule-based $R_e^{val}$ , w/ $R_s^{rec}$ )	91.96
w/ dual (w/ model-based $R_e^{val}$ , w/ $R_s^{rec}$ )	<b>92.51</b>

Table 6: Ablations on the test set of English-FOL.

exists in all three datasets and has been ignored by previous works. To compensate for this, we perform manual statistics on the incorrect prediction results of Dual-mT5-base on the test set of Chinese-PL/FOL. About 5.2% of errors are due to the above. We think this may be solved by defining a series of equivalence transformation rules. We leave this for future work.

## 5.7 Improvements to Downstream Tasks

To further demonstrate the value of the generated logical expressions, we take Natural Language Inference (NLI) as a downstream task. NLI involves reading a pair of sentences and judging the relationship between their meanings, such as entailment, neutral and contradiction. We explore if additional logical expressions improve the model’s performance on the NLI task. We conduct experiments on the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015). The backbone model is BERT (Devlin et al., 2019) which concatenates two sentences ( $s1$  and  $s2$ ) with special tokens as input, and uses the representation of  $[CLS]$  token for text classification. In practice, we concatenate the logical expressions ( $e1$  and  $e2$ )

Chinese-PL/FOL	
<b>Natural Language Utterance</b>	Xiao Ming lives alone because he is not married.
<b>Symbolic Definition</b>	$A(x)$ : $x$ lives alone; $B(x)$ : $x$ is married; $a$ : Xiao Ming
<b>Ground Truth</b>	$\neg B(a) \rightarrow A(a)$
<b>Prediction</b>	$A(a) \rightarrow \neg B(a)$
<b>Natural Language Utterance</b>	The dark night has given me a pair of black eyes, but I use them to find the light.
<b>Symbolic Definition</b>	$A(x, y)$ : $x$ give me $y$ ; $B(x, y)$ : i use $x$ to find $y$ ; $a$ : dark night; $b$ : black eyes; $c$ : light
<b>Ground Truth</b>	$A(a, b) \wedge B(b, c)$
<b>Prediction</b>	$A(a, b) \wedge B(a, c)$
<b>Natural Language Utterance</b>	Not everyone who can play football can play basketball.
<b>Symbolic Definition</b>	$A(x)$ : $x$ can play football; $B(x)$ : $x$ can play basketball; $x$ : person
<b>Ground Truth</b>	$\neg \forall x(A(x) \rightarrow B(x))$
<b>Prediction</b>	$\exists x(A(x) \wedge (\neg B(x)))$

Table 7: Case study of Dual-(m)T5-base on Chinese-PL/FOL. Text in red and brown represents the difference between the prediction and ground truth.

corresponding to the sentences ( $s_1$  and  $s_2$ ) on the input, i.e.,  $[CLS] s_1 e_1 [SEP] s_2 e_2 [SEP]$ . We vary the amount of training data from 10k to 100k and the results are shown in Fig.2.

As shown in the results, the model has better performance by introducing logical expressions when the amount of training data is not large. As the amount of training data increases, the role of logical expressions gradually decreases. We conclude that the generated logical expression explicitly represents the logical information contained in the sentences and is suitable for the NLI task in low-resource scenarios as a supplementary.

## 6 Related Works

**Parsing Natural Language into PL/FOL** Logic expressions are commonly written in standardized mathematical notation, and learning this notation typically requires many years of experience. Barker-Plummer et al. (2009) study why students find translating natural language sentences into FOL hard and systematically categorize the problems encountered by students. Bansal (2015) proposes a rule-based framework that leverages the Part-of-speech structure of natural language sentences. Limited to the manually defined rules and

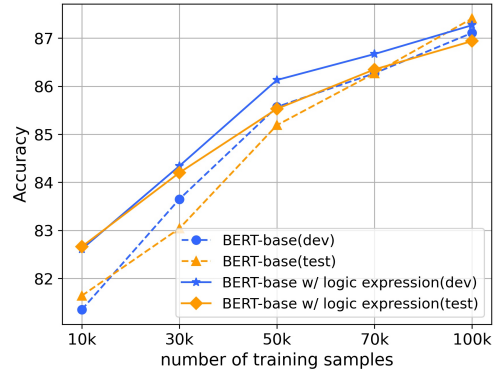


Figure 2: Accuracy on the dev and test set of the SNLI dataset.

a small amount of experimental data, the system only works under a specific setting. With the development of deep learning, neural approaches alleviate the need for manually defining lexicons. Singh et al. (2020) examine the capability of neural models on parsing FOL from natural language sentences. They propose to disentangle the representations of different token categories while generating FOL and use category prediction as an auxiliary task. Unfortunately, they do not release the dataset they construct. Levkovskiy and Li (2021) release a dataset containing English-FOL sentence pairs and set up a baseline encoder-decoder model, but the dataset is not challenging for it is generated by templates, and vanilla models obtain high scores.

**Dual Learning** Dual learning is first proposed to improve neural machine translation (NMT) (He et al., 2016). The author makes full use of monolingual corpus to improve the effectiveness of the model through dual learning. Xia et al. (2017) introduce a probabilistic duality term to serve as a data-dependent regularizer to better guide the dual supervised learning. Since then, the idea of dual learning has been applied in various tasks, such as Question Answering/Generation (Tang et al., 2017), Open-domain Information Extraction/Narration (Sun et al., 2018), Semantic Parsing with lambda calculus (Cao et al., 2019, 2020), and Emotion-Controllable Response Generation (Shen and Feng, 2020).

## 7 Conclusion

In this paper, we introduce **Dual-(m)T5**, an effective dual reinforcement learning framework for parsing natural language into PL/FOL. A novel reward mechanism is proposed to avoid manually defining the validity reward in RL. An effective training strategy is further proposed to stabilize the



training process. Experimental results show that the proposed method outperforms competitors on several datasets. By introducing logical expressions, we further enhance the existing NLI model. In addition to the technical contribution, a new dataset called **Chinese-PL/FOL** is constructed to aid further research in this field.

## Acknowledgement

We thank the anonymous reviewers for their constructive suggestions. This work was supported by National Key Research and Development Project (No.2020AAA0109302), Shanghai Science and Technology Innovation Action Plan (No.19511120400), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103).

## References

- Naman Bansal. 2015. *Translating Natural Language Propositions to First Order Logic*. Ph.D. thesis, INDIAN INSTITUTE OF TECHNOLOGY KANPUR.
- Dave Barker-Plummer, Richard Cox, and Robert Dale. 2009. *Dimensions of difficulty in translating natural language into first order logic*. *International Working Group on Educational Data Mining*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. *Curriculum learning*. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.
- Patrick Blackburn and Johannes Bos. 2005. *Representation and inference for natural language: A first course in computational semantics*. Center for the Study of Language and Information Amsterdam.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. *A large annotated corpus for learning natural language inference*. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.
- Ruisheng Cao, Su Zhu, Chen Liu, Jieyu Li, and Kai Yu. 2019. *Semantic parsing with dual learning*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 51–64, Florence, Italy. Association for Computational Linguistics.
- Ruisheng Cao, Su Zhu, Chenyu Yang, Chen Liu, Rao Ma, Yanbin Zhao, Lu Chen, and Kai Yu. 2020. *Unsupervised dual paraphrasing for two-stage semantic parsing*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6806–6817, Online. Association for Computational Linguistics.
- Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. *Expanding the scope of the ATIS task: The ATIS-3 corpus*. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Li Dong and Mirella Lapata. 2016. *Language to logical form with neural attention*. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.
- Li Dong and Mirella Lapata. 2018. *Coarse-to-fine decoding for neural semantic parsing*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia. Association for Computational Linguistics.
- Richard Evans and Edward Grefenstette. 2018. *Learning explanatory rules from noisy data*. *Journal of Artificial Intelligence Research*, 61:1–64.
- Donglai Ge, Junhui Li, and Muhua Zhu. 2019. *A transformer-based semantic parser for nlpc-2019 shared task 2*. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer.
- Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. *Dual learning for machine translation*. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 820–828.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. *The curious case of neural text degeneration*. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Daiki Kimura, Masaki Ono, Subhajt Chaudhury, Ryosuke Kohita, Akifumi Wachi, Don Joven Agravante, Michiaki Tatsubori, Asim Munawar, and Alexander Gray. 2021. *Neuro-symbolic reinforcement learning with first-order logic*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3505–3511.

- Oleksii Levkovskiy and Wei Li. 2021. Generating predicate logic expressions from natural language. In *SoutheastCon 2021*, pages 1–8. IEEE.
- Carmelo Fabio Longo and Corrado Santoro. 2020. **Adcaspar: Abductive-deductive cognitive architecture based on natural language and first order logic reasoning**. In *NL4AI@ AI\* IA*, pages 73–86.
- Ilya Loshchilov and Frank Hutter. 2018. **Fixing weight decay regularization in adam**.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. **Effective approaches to attention-based neural machine translation**. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.
- Zhihao Ma, Yuzheng Zhuang, Paul Weng, Dong Li, Kun Shao, Wulong Liu, Hankz Hankui Zhuo, and HAO Jianye. 2020. Interpretable reinforcement learning with neural symbolic logic.
- Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. **Competence-based curriculum learning for neural machine translation**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.
- Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. **Abstract syntax networks for code generation and semantic parsing**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. **Exploring the limits of transfer learning with a unified text-to-text transformer**. *arXiv preprint arXiv:1910.10683*.
- Lei Shen and Yang Feng. 2020. **CDL: Curriculum dual learning for emotion-controllable response generation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 556–566, Online. Association for Computational Linguistics.
- Hrituraj Singh, Milan Aggrawal, and Balaji Krishnamurthy. 2020. **Exploring neural models for parsing natural language into first-order logic**. *arXiv preprint arXiv:2002.06544*.
- Jianlin Su. 2021. **Roformer-sim: Integrating retrieval and generation into roformer**. Technical report.
- Mingming Sun, Xu Li, and Ping Li. 2018. **Logician and orator: Learning from the duality between language and knowledge in open domain**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2119–2130, Brussels, Belgium. Association for Computational Linguistics.
- Zeyu Sun, Qihao Zhu, Yingfei Xiong, Yican Sun, Lili Mou, and Lu Zhang. 2020. **Treegen: A tree-based transformer architecture for code generation**. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34.
- Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. **Policy gradient methods for reinforcement learning with function approximation**. In *Advances in neural information processing systems*, pages 1057–1063.
- Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. **Question answering and question generation as dual tasks**. *arXiv preprint arXiv:1706.02027*.
- Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. **Dual supervised learning**. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3789–3798. PMLR.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. **mT5: A massively multilingual pre-trained text-to-text transformer**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Yuan Yang and Le Song. 2019. **Learn to explain efficiently via neural logic inductive learning**. *arXiv preprint arXiv:1910.02481*.
- Pengcheng Yin and Graham Neubig. 2018. **TRANX: A transition-based neural abstract syntax parser for semantic parsing and code generation**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. **Pegasus: Pre-training with extracted gap-sentences for abstractive summarization**. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Kai Zhao and Liang Huang. 2015. **Type-driven incremental semantic parsing with polymorphism**. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

## A Algorithm

---

### Algorithm 1 Training Scoring Model

---

**Input:** Supervised dataset  $\mathcal{L} = \{\langle s, e \rangle\}$ ; number of nucleus sampling  $k$ ; Fine-tuned NL2LE model

**Output:** scoring model

```
1:  $\mathcal{P} \leftarrow \{\}, \mathcal{N} \leftarrow \{\}$ 
2: for all  $\langle s, e \rangle \in \mathcal{L}$  do
3:    $\mathcal{P} \leftarrow \mathcal{P} \cup \{\langle s, e \rangle\}$ 
4:   Given  $s$ , fine-tuned NL2LE model generates  $k$  logical expressions  $\{e_i\}$  via nucleus sampling
5:   for all  $e_i \in \{e_i\}$  do
6:     if  $e_i \neq e$  then
7:        $\mathcal{N} \leftarrow \mathcal{N} \cup \{\langle s, e_i \rangle\}$ 
8:     end if
9:   end for
10: end for
11: repeat
12:   Update scoring model w.r.t. Eq.(10)
13: until scoring model converges
```

---

## B PL and FOL

FOL represents entities and actions in natural language through quantified variables and consists of predicates which take variables as arguments and attach semantics to variables (Blackburn and Bos, 2005), while PL is a relatively simple logical expression and does not deal with quantified variables. Formally, a predicate  $P(v_1; v_2; \dots; v_n)$  in PL/FOL is an  $n$ -ary function of variables  $v_i$  that are combined through logical connectives: *logical and* ( $\wedge$ ), *logical or* ( $\vee$ ), *logical not* ( $\neg$ ), *logical implication* ( $\rightarrow$ ), *logical equivalent* ( $\leftrightarrow$ ). What’s more, there are two types of quantifiers for FOL: *universal* ( $\forall$ ) which specifies that sub-formula within its scope is true for all instances of the variable and *existential* ( $\exists$ ) which asserts existence of at least one instance represented by a variable under which the sub-formula holds true.

## C Baselines

### English-FOL and Chinese-PL/FOL

- ATT (Luong et al., 2015). ATT represents attention-based Seq2seq model.
- Text2log (Levkovskyi and Li, 2021). Text2log is the latest work on converting natural language to FOL. This approach is only appli-

---

### Algorithm 2 Full Training Process

---

**Input:** Supervised dataset  $\mathcal{L} = \{\langle s, e \rangle\}$ ; Unsupervised dataset  $\mathcal{U} = \{s'\}$ ; number of nucleus sampling  $k$ ; hyper parameters  $\alpha$  and  $\beta$ ; curriculum training batches  $T$

**Output:** NL2LE model

```
1: // Pre-train NL2LE and LE2NL models
2: Fine-tune NL2LE model with  $\langle s, e \rangle$  from  $\mathcal{L}$  and curriculum learning based on Eq.(11)
3: Fine-tune LE2NL model with  $\langle e, s \rangle$  from  $\mathcal{L}$ 
4: repeat
5:   // Dual reinforcement learning
6:   Get mini-batch  $\{s\}$  from  $\mathcal{L} \cup \mathcal{U}$ 
7:   for all  $s \in \{s\}$  do
8:     NL2LE model generates  $k$  logical expressions  $\{e_i\}$  for  $s$  via nucleus sampling
9:     for all  $e_i \in \{e_i\}$  do
10:      Obtain validity and reconstruction reward for  $e_i$ 
11:    end for
12:  end for
13:  Update  $\Theta_{NL2LE}$  and  $\Theta_{LE2NL}$  w.r.t. Eq.(2) and Eq.(5) respectively
14:  // Supervisor Guidance
15:  Get mini-batch  $\{\langle s, e \rangle\}$  from  $\mathcal{L}$ 
16:  Fine-tune NL2LE model with  $\{\langle s, e \rangle\}$ 
17:  Fine-tune LE2NL model with  $\{\langle e, s \rangle\}$ 
18: until NL2LE model converges
```

---

cable to English-FOL since a character-level recurrent neural network is leveraged.

- GPT-2 (Radford et al., 2019). GPT-2 is a huge transformer-based model trained on massive datasets and achieves state-of-the-art results on several language modeling datasets in a zero-shot setting when it is proposed. We experiment with different sizes of GPT. Since there is no GPT-large for Chinese, it is vacant in the experiment.
- T5 (Raffel et al., 2019) / mT5 (Xue et al., 2021). T5 refers to the “Text-to-Text Transfer Transformer” which converts several NLP tasks to Text-to-Text task, and mT5 is a multi-lingual version of T5.

### ATIS

- TISP (Zhao and Huang, 2015) An incremental semantic parser that is guided by subtyping and polymorphism.

- Seq2tree (Dong and Lapata, 2016) Seq2tree is a method based on an attention-enhanced encoder-decoder model.
- ASN+SUPATT (Rabinovich et al., 2017) This work introduces abstract syntax networks, a modeling framework for code generation and semantic parsing.
- Tranx (Yin and Neubig, 2018) Tranx uses a transition system based on the abstract syntax description language for the target meaning representations.
- Coarse2fine (Dong and Lapata, 2018) Coarse2fine generates meaning sketches first and then predicts missing details to obtain full meaning representations.
- Transformer (Ge et al., 2019) Transformer is a deep learning model that adopts the mechanism of self-attention and is originally proposed for machine translation.
- ATTPTR + Dual (Cao et al., 2019) It is the first work to propose the use of dual learning for semantic parsing, and is the basis of our work.
- TreeGen (Sun et al., 2020) TreeGen uses the attention mechanism of Transformer to alleviate the long-dependency problem and introduces an Abstract Syntax Tree (AST) reader to combine grammar rules and the AST structure.

## D Implementation Details

We use Pytorch<sup>4</sup> library for implementing an auto-differentiable graph of our computations. For **pre-training**, (m)T5-small/base are trained with an AdamW optimizer (Loshchilov and Hutter, 2018) initialized with a learning rate of 1e-3/1e-4 with a decay rate of 1e-3/1e-2 respectively. For **dual reinforcement learning**, models are trained with an AdamW optimizer initialized with a learning rate of 1e-5 with a decay rate of 1e-3 for (m)T5-small/base. The batch size is fixed to 8, and the max input and output sentence length are set to 128. Since the size of the test set for the Chinese-PL/FOL dataset is small, we repeat each experiment using 3 different random seeds and report the median number and standard deviation to avoid small sample

<sup>4</sup><https://pytorch.org>

instability in the results obtained. Training runs until the performance on validation set does not improve. We use PEGASUS (Zhang et al., 2020) fine-tuned for paraphrasing<sup>5</sup> for english paraphrasing, and RoFormer-Sim (Su, 2021)<sup>6</sup> for chinese paraphrasing. For each natural language utterance in the datasets, we generate one synonymous sentence as unlabeled data.<sup>7</sup> Our models run on a computer with Intel(R) Xeon(R) Gold 6230R CPU, 4 GeForce RTX 3090, 64GB of RAM, and Ubuntu 20.04.

## E Error Analysis on English-FOL and ATIS

The bad cases of English-FOL and ATIS are presented in Table 8. We select two typical bad cases from two datasets respectively. From case 1 of English-FOL, we can find that there are some labeling errors in the dataset. The prediction is completely correct while the ground truth is wrong. The error in case 2 is due to the fact that the English-FOL dataset does not clearly indicate what the predicate is, which leads to a slight difference between the prediction and the ground truth.<sup>8</sup> For ATIS, the model makes a small mistake in case 1, which indicates that the model is not good enough to handle the details. The error in case 2 is also caused by the predicate not explicitly indicating. If the predicate can be unified, we believe that the model can answer correctly.

<sup>5</sup>[https://huggingface.co/tuner007/pegasus\\_paraphrase](https://huggingface.co/tuner007/pegasus_paraphrase)

<sup>6</sup><https://github.com/ZhuiyiTechnology/roformer-sim>

<sup>7</sup>Generating top-k ( $k \geq 1$ ) synonymous sentences for each natural language utterance leads to poor performance because the unlabeled data is too similar to each other.

<sup>8</sup>In contrast, our dataset clearly points out what the predicate is, effectively solving this problem.



English-FOL
<p><b>Natural Language Utterance</b> All people are welcome.</p> <p><b>Ground Truth</b> exists x1.(<b>_all</b>(x1) &amp; <b>_people</b>(x1) &amp; <b>_welcome</b>(x1))</p> <p><b>Prediction</b> all x1.(<b>_people</b>(x1) → <b>_welcome</b>(x1))</p>
<p><b>Natural Language Utterance</b> Every database accepts parentheses.</p> <p><b>Ground Truth</b> all x1.(<b>_database</b>(x1) → exists x2.(<b>_parenthesis</b>(x2) &amp; <b>_accept</b>(x1,x2)))</p> <p><b>Prediction</b> all x1.(<b>_database</b>(x1) → exists x2.(<b>_parenthes</b>(x2) &amp; <b>_accept</b>(x1,x2)))</p>
ATIS
<p><b>Natural Language Utterance</b> What city does al0 fly out of?</p> <p><b>Ground Truth</b> lambda \$0 e ( and ( city \$0 ) ( exists \$1 ( and ( flight \$1 ) ( airline \$1 al0 ) ( <b>from</b> \$1 \$0 ) ) ) )</p> <p><b>Prediction</b> lambda \$0 e ( and ( city \$0 ) ( exists \$1 ( and ( flight \$1 ) ( airline \$1 al0 ) ( <b>to</b> \$1 \$0 ) ) ) )</p>
<p><b>Natural Language Utterance</b> List the st0 airport.</p> <p><b>Ground Truth</b> lambda \$0 e ( and ( airport \$0 ) ( <b>loc:t</b> \$0 st0 ) )</p> <p><b>Prediction</b> lambda \$0 e ( and ( airport \$0 ) ( <b>located</b> at \$0 st0 ) )</p>

Table 8: Case study of Dual-(m)T5-base on English-FOL and ATIS. Text in red and brown represents the difference between the prediction and ground truth.