Master's Thesis

An Efficient Fine-tuning Method for Hierarchical Semantic Parsing in Task-Oriented Dialog System by Structure-aware Boosting and Grammar Constraints

DO, Truong Dinh

Supervisor NGUYEN, Minh Le

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Master Degree)

September, 2023

Abstract

Task-oriented dialogue (TOD) systems have become increasingly crucial in various domains, providing efficient and appropriate interactions with users to perform specific tasks. These systems, such as voice assistants and chatbots, rely on semantic parsing models to convert user queries into machine-understandable representations, enabling accurate interpretation and appropriate responses. While traditional semantic parsing methods have shown effectiveness in handling flat intent-slot structures, they struggle to cope with the complexities introduced by hierarchical semantic parsing (HSP). Hierarchical representations offer increased expressiveness, allowing for more nuanced interpretations of user queries, but they also pose challenges in accurately labeling and capturing nested structures. In recent years, the field of natural language processing (NLP) has witnessed significant advancements, largely driven by the remarkable success of pre-trained language models like BERT. These models have demonstrated exceptional performance across a wide range of tasks, including hierarchical semantic parsing. Among these models, the RINE model has emerged as a leader by providing state-of-the-art (SOTA) results. The distinctive strength of the RINE model lies in its innovative utilization of a recursive insertion-based mechanism. This mechanism allows the model to leverage the decoded tree from previous steps as direct input for the current step, enabling more effective and context-aware parsing. Despite the impressive achievements observed in the realm of hierarchical semantic parsing, there is still significant room for enhancing the overall performance of these systems. Previous works do not explicitly consider structured information within utterances. However, this information about sentence structures is crucial in the semantic parsing task, specifically when parsing complex utterances of hierarchical semantic parsing.

Taking these points into consideration, this thesis focuses on two key enhancements: strengthening the hierarchical structure awareness of pre-trained language models and employing dynamic pruning of unpromising decoding directions using inductive grammar. By incorporating sentence structure information, we aim to improve the performance of hierarchical semantic parsing systems. To achieve this objective, we propose the StructSP framework, which leverages hierarchical representations for their deep fine-grained structure and consists of two distinct phases: structure-aware boosting and grammar-based RINE. In the structure-aware boosting phase, our aim is to enhance the representation of structured information within pre-trained language models. We extend the vanilla mask language modeling (MLM) task by giving higher priority to logical tokens present in the linearized hierarchical representation. By doing so, we encourage the model to

focus on the essential elements of the logical structure and improve its understanding of the semantic relationships. Furthermore, we introduce a novel subtask called relative tree agreement, which allows the model to develop closer vector representations for relative trees in the same parsing process. By learning the hidden relationships between these intermediate parsing steps, our framework gains a more comprehensive understanding of the logical structure and improves its ability to capture intricate dependencies. In the second phase, grammar-based RINE, we incorporate grammar extracted from annotated data to help in the prediction of node labels. By utilizing this grammar during the parsing process, we aim to guide the model toward more accurate predictions by preventing branches in the decoding process that won't yield a solution, thus mitigating errors and improving the overall performance of the system.

To evaluate the proposed StructSP framework, comprehensive experiments were conducted on the widely used TOP and TOPv2 datasets. The performance comparison with several previous approaches on the TOP dataset demonstrates the effectiveness of the proposed approach. Our method achieves an impressive exact match (EM) score of 88.18, outperforming existing SOTA models by notable margins. Specifically, it surpasses the current SOTA model, RINE, by 0.61 EM points. The experiments on the TOPv2 dataset, which includes low-resource scenarios, further highlight the superiority of our approach. In these settings, our models outperform the RINE model by up to 1.02 EM points. The results indicate that our StructSP framework effectively leverages hierarchical semantic structured information, demonstrating its potential to enhance the accuracy and reliability of semantic parsing systems. The incorporation of grammar proves to be beneficial in most cases, guiding the decoding process and improving label predictions. However, in extremely limited training data scenarios, the performance gains from grammar utilization may not be substantial due to insufficient coverage of grammar patterns in the data. Addressing this issue requires further investigation and potential improvements in grammar extraction techniques.

In the analysis section, we conducted a systematic ablation study to evaluate the impact of different factors on the performance of our proposed model. By comparing the performance of the full-setting model, which includes both the Structure-aware boosting and Grammar-based RINE phases, with variations where specific components were selectively disabled, we identified the key elements that significantly influenced the model's performance. The results highlighted the importance of structure-focused MLM and relative tree agreement in the structure-aware boosting phase, as well as the incorporation of grammar in the grammar-based RINE phase, as these factors contributed to enhanced performance on the TOP dataset. Furthermore, we investigated the impact of the masking probability on the model's performance by varying the value of masking probabilities during training. Finally, to gain insights into the model's perfor-

mance on specific input queries, we conducted a case study comparing the outputs of the baseline RINE model with our StructSP model on the validation set of the TOP dataset. We presented some examples, showcasing instances where our model outperformed the baseline in accurately capturing the underlying semantics of the input query. While our model demonstrated improved performance in most cases, there were also examples where both models produced incorrect outputs, highlighting areas for potential improvement in handling intricate queries.

In summary, this thesis contributes to the field of hierarchical semantic parsing by proposing an innovative approach to integrate hierarchical semantic structured information into pre-trained language models. By leveraging grammar during the parsing process, our method achieves superior performance compared to existing SOTA models in task-oriented semantic parsing. This work demonstrates the potential of our StructSP framework to advance the accuracy and reliability of semantic parsing systems in various applications. Future work could focus on refining the grammar extraction process, exploring other mechanisms for deep structure awareness, and investigating techniques to improve performance in low-resource scenarios.

Acknowledgement

First of all, I would like to express my deep gratitude to my supervisor, Prof. Nguyen Le Minh, who provided wholehearted guidance and support throughout my studies at JAIST, particularly during my graduation thesis.

I would also like to thank my second supervisor, Associate Professor Kiyoaki Shirai, and my minor supervisor, Prof. Shinobu Hasegawa, for their continuous guidance and support throughout my entire research process. Additionally, I am deeply grateful to the teachers, colleagues, and fellow researchers at Nguyen's Lab. Their invaluable contributions and assistance have significantly facilitated my research journey, making it smoother and easier. The collaborative atmosphere in the lab, along with their willingness to share knowledge and offer support, has greatly enhanced my understanding of complex concepts and expanded my professional knowledge.

I extend my sincere appreciation to all the professors at JAIST for creating a nurturing and supportive academic environment. I am also grateful to the Ministry of Education, Culture, Sports, Science, and Technology for granting me with the MEXT scholarship throughout my master's program.

Lastly, I want to express my deep appreciation to my family, who have played a significant role in my upbringing and have supported me in becoming the person I am today. To my father, mother, and sisters, thank you for always encouraging and standing by me during difficult and challenging times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Semantinc Parsing in Task-oriented Dialog Systems

Task-oriented dialogue (TOD) systems have become increasingly important in various domains by providing efficient and appropriate interactions with user utterances to perform specific tasks (Bai et al., 2022; He et al., 2022; Pasupat et al., 2019). These systems, such as voice assistants and chatbots, rely on semantic parsing models to convert user queries into machine-understandable representations, enabling accurate interpretation and appropriate responses (Z. Zhang et al., 2020; Yan et al., 2017). Figure 1.1 provides an illustrative example of a semantic parser operating within a TOD system.

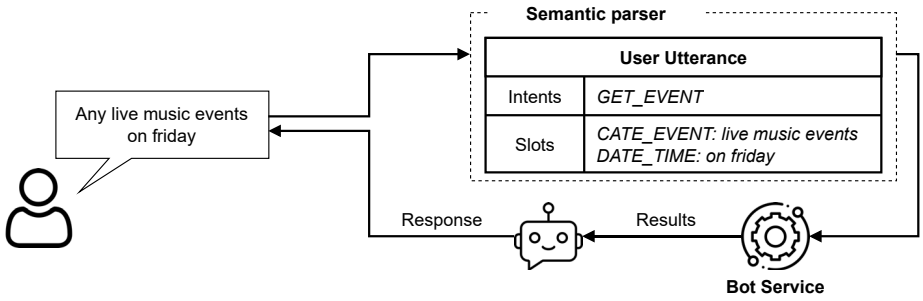
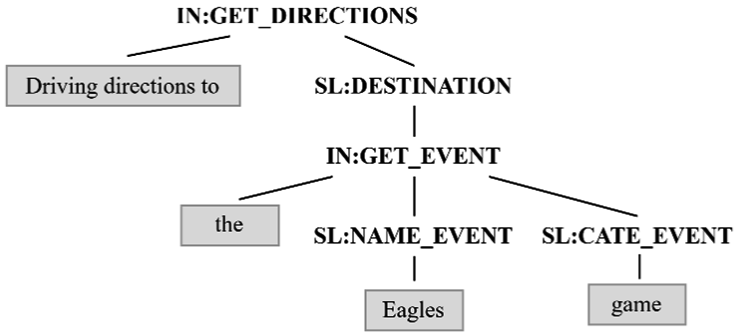


Figure 1.1: An example of the role of a semantic parser in a task-oriented dialog system.

Specifically, semantic parsing plays a crucial role in transforming user utterances into structured representations that include the identification of the user's intentions (intents) and relevant entities (slots). For instance, an intent could be la-

beled as GET_EVENT to obtain event information, accompanied by a corresponding slot such as DATE_TIME indicating the specific date and time of the event (Figure 1.1). While traditional semantic parsing primarily focused on single-level intents and flat slots (Mesnil et al., 2013; Q. Chen et al., 2019; Phuong et al., 2022), the advent of hierarchical representations has led to the emergence of hierarchical semantic parsing (HSP) (Gupta et al., 2018). HSP involves the identification of nested intents and slots within a user query, enabling the capture of complex linguistic structures present in TOD systems (Aghajanyan et al., 2020). Figure 1.2 provides an example illustrating the hierarchical representation of an utterance through a fully parsed tree.



Linearized Representation: [IN:GET_DIRECTIONS Driving directions to [SL:DESTINATION [IN:GET_EVENT the [SL:NAME_EVENT Eagles] [SL:CATE_EVENT game]]]]

Figure 1.2: An example of hierarchical representation with the input utterance *"Driving directions to the Eagles game"*. Where *"SL:"* denotes slot, *"IN:"* denotes intent.

1.2 Research Gap

While traditional semantic parsing methods have shown effectiveness in flat intent-slot structures (Mesnil et al., 2013; Q. Chen et al., 2019), they struggle to handle the complexities introduced by HSP. Hierarchical representations offer increased expressiveness, allowing for more nuanced interpretations of user queries, but they also pose challenges in accurately labeling and capturing nested structures (Mansimov & Zhang, 2022).

In recent years, the field of natural language processing (NLP) has witnessed significant advancements, largely driven by the remarkable success of pre-trained language models like BERT (Devlin et al., 2019). These models have demonstrated exceptional performance across a variety of tasks, including hierarchical

semantic parsing (Ziai, 2019; Herzig & Berant, 2021; Rubin & Berant, 2021). Among these models, RINE model (Mansimov & Zhang, 2022) has emerged as a leader by providing state-of-the-art (SOTA) results. The distinctive strength of the RINE model lies in its innovative utilization of a *recursive insertion-based* mechanism. This mechanism allows the model to leverage the decoded tree from previous steps as direct input for the current step, enabling more effective and context-aware parsing. Despite the impressive achievements observed in the realm of hierarchical semantic parsing, there is still significant room for enhancing the overall performance of these systems.

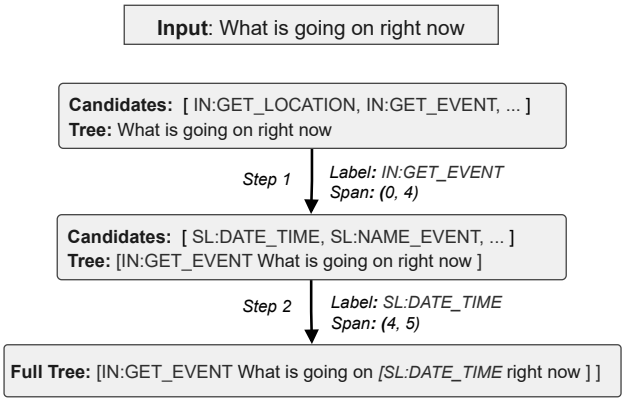


Figure 1.3: Example of the parsing process in our framework. Each tree is a linearized representation, where *IN:*, *SL:* represent the intent and slot, respectively. Logical tokens are highlighted in red.

One promising approach is to adapt the pre-trained language models to the specific characteristics of the semantic parsing task. These language models, although powerful in their general language understanding capabilities, lack explicit knowledge of the logical structure present inside sentences (Bender et al., 2021). Therefore, incorporating sentence structure information is crucial for accurate and effective semantic parsing. In addition, task-oriented dialogue systems typically adhere to task- or domain-specific grammars, where certain intents consistently serve as parent nodes for specific slots. For instance, in the navigation domain, the slot `SL:DESTINATION` usually follows by an intent `SL:GET_DIRECTIONS`. Surprisingly, existing methods largely overlook this valuable information (Mansimov & Zhang, 2022).

1.3 Objectives

Taking these points into consideration, this thesis focuses on two key enhancements: strengthening the hierarchical structure awareness of pre-trained language

models and employing dynamic pruning of unpromising decoding directions using inductive grammar. By incorporating the logical structure of sentences and leveraging task-specific grammar, we aim to improve the performance of hierarchical semantic parsing systems. To achieve this objective, we propose the StructSP framework, which leverages hierarchical representations for their deep fine-grained structure (Y. Zhao et al., 2022; Desai & Aly, 2021; Louvan & Magnini, 2020) and consists of two distinct phases: **structure-aware boosting** and **grammar-based RINE**.

In the **structure-aware boosting phase**, our aim is to enhance the representation of structured information within pre-trained language models. We extend the vanilla mask language modeling (MLM) task (Devlin et al., 2019) by giving higher priority to logical tokens present in the linearized hierarchical representation. By doing so, we encourage the model to focus on the essential elements of the logical structure and improve its understanding of the semantic relationships. Furthermore, we introduce a novel subtask called *relative tree agreement*, which allows the model to develop closer vector representations for relative trees in the same parsing process. By learning the hidden relationships between these intermediate parsing steps, our framework gains a more comprehensive understanding of the logical structure and improves its ability to capture intricate dependencies.

In the second phase, **grammar-based RINE**, we incorporate grammar extracted from annotated data to help in the prediction of node labels. By utilizing this grammar during the parsing process, we aim to guide the model toward more accurate predictions by preventing branches in the decoding process that won't yield a solution, thus mitigating errors and improving the overall performance of the system.

Through comprehensive experiments and evaluations, we seek to demonstrate the superiority of the StructSP framework compared to existing models in task-oriented semantic parsing on the widely used TOP and TOPv2 datasets. By effectively integrating hierarchical semantic structured information into a pre-trained language model and utilizing grammar contains in the decoding step, our framework aims to achieve remarkable results, showcasing its potential to advance the accuracy and performance of semantic parsing systems in various applications. In summary, this work presents three key contributions:

- Introduction of an effective fine-tuning approach to integrating hierarchical semantic structured information into pre-trained language models.
- Introduction of grammar integration mechanism in the decoding parsing process to avoid label predictions that won't yield a solution.
- Demonstration of the superior performance of the StructSP framework compared to existing models in task-oriented semantic parsing using the TOP

and TOPv2 datasets.

1.4 Thesis Outline

This thesis comprises five chapters, Chapter 1 is the introduction. The outline of the remaining chapters is as follows:

- In the first section of Chapter 2 a comprehensive review of relevant literature and previous works pertaining to the fundamental concepts and methods explored in this thesis are described. It focuses on topics including: hierarchical semantic parsing, pre-trained language model adaptation, and grammar-constrained neural network models. The second section provides background knowledge of this thesis.

- Chapter 3 presents a detailed explanation of the proposed method, which consists of two distinct fine-tuning phases: Structure-aware Boosting and Grammar-based RINE. Each step of the method is elucidated to provide a clear understanding of its implementation.

- Chapter 4 describes the experimental settings, evaluation methods, and primary results obtained from the conducted experiments.

- Chapter 5 involves a thorough analysis aimed at gaining deeper insights into the proposed method's performance and behavior.

- Chapter 6 serves as the concluding chapter, summarizing the main findings and conclusions drawn from the research. It also offers suggestions and directions for potential future improvements.

Chapter 2

Related Works and Background Knowledge

2.1 Related Works

2.1.1 Hierarchical Semantic Parsing

Task-oriented parsing (TOP) (Gupta et al., 2018) and its variant TOPv2 (X. Chen et al., 2020) have emerged as prominent benchmarks for evaluating the performance of task-oriented semantic parsing models. These datasets have been instrumental in introducing a hierarchical representation that emphasizes the significance of nested sub-logic composition in task-oriented dialog systems. The hierarchical structure allows for a more fine-grained understanding of the user intentions and enables the modeling of complex dependencies between different components. As a result, numerous approaches have been proposed to tackle the semantic parsing task on these datasets, aiming to improve the effectiveness and accuracy of parsing models.

Einolghozati et al. (2019) proposed and demonstrated the effectiveness of three different improvements to the semantic parsing model for task-oriented dialog, including contextualized embeddings, ensembling, and pairwise re-ranking, which collectively address various errors in hierarchical representation. Aghajanyan et al. (2020) proposed a novel semantic representation, known as the "decoupled representation," for task-oriented conversational systems. This representation overcomes the constraints of the structured hierarchical representation by integrating session-based properties like co-reference resolution and context carryover, allowing for a more comprehensive understanding of queries. Furthermore, the authors employed sequence-to-sequence (S2S) models to accurately predict this new representation data. Rongali et al. (2020) introduced a unified architecture utilizing S2S and Pointer Generator Network, which offers a flexible and

precise approach to process both simple and complex queries in virtual assistant systems. The noteworthy aspect of this architecture is its ability to handle queries without imposing any limitations on the semantic parse schema, allowing for improved flexibility and accuracy in parsing. Zhu et al. (2020) proposed a non-autoregressive parser based on the Insertion Transformer (Stern et al., 2019), which addresses the challenges of slow inference time and poor performance commonly encountered in semantic parsing. Additionally, Babu et al. (2021) developed a non-autoregressive approach for semantic parsing using an efficient S2S model architecture, addressing the limitations of higher compute requirements and latency associated with traditional autoregressive models. W. Zhao et al. (2022) explored naturalized semantic parsing for TOP by introducing a general reduction of hierarchical representation to abstractive question answering, which addresses limitations associated with canonical paraphrasing. Mansimov and Zhang (2022) introduced a novel approach called Recursive INsertion-based Encoder (RINE) model. In general, this approach employs an encoder network to construct the semantic parse tree step by step, predicting non-terminal labels along with their positions in the linearized tree. During the generation process, the model recursively inserts the predicted non-terminal labels at the predicted positions until the tree construction is complete.

However, despite these advancements, the hierarchical structure information has often been disregarded in parsing processes. Therefore, our research focuses on leveraging this hierarchical structure to continue pre-training language models and utilizing an inductive grammar derived from annotated data to guide node label predictions, aiming to enhance the performance and accuracy of semantic parsing models.

2.1.2 Pre-trained Language Model Adaptation

In recent years, the adaptation of pre-trained language models has emerged as a powerful technique for enhancing the performance of natural language processing tasks across various domains. Notably, researchers have explored the application of pre-trained language models in tasks such as summarization (J. Zhang et al., 2020), knowledge inference (Sun et al., 2019; W. Liu et al., 2020), and biomedical natural language processing (Lee et al., 2020; Gu et al., 2021) by adapting pre-trained language models and achieving remarkable results.

When it comes to semantic parsing, Yu, Zhang, et al. (2020) introduced SCORE, a pre-training approach for Conversational Semantic Parsing (CSP) tasks. SCORE aims to capture the alignment between dialogue flow and structural context, addressing the limitations of existing pre-trained language models in representing natural language references to contextual structural data. In addition, GraPPa (Yu, Wu, et al., 2020) was introduced as a pre-training approach specifically designed

for table semantic parsing. This approach leverages a synchronous context-free grammar (SCFG) to generate synthetic question-SQL pairs over high-quality tables. The model is pre-trained on this synthetic data using a novel text-schema linking objective that predicts the syntactic role of a table field in the SQL for each question-SQL pair. Moreover, the utilization of Abstract Meaning Representation (AMR) as a source of explicit semantic knowledge for pre-training language models in the field of dialogue comprehension is explored by Bai et al. (2022). The authors proposed a semantic-based pre-training framework that extends the standard pre-training framework by incorporating three tasks related to core semantic units, semantic relations, and overall semantic representation based on AMR graphs.

In contrast to existing methods such as SCORE and Grappa, which rely on generating or augmenting large artificial datasets to learn structure information (Yu, Zhang, et al., 2020; Yu, Wu, et al., 2020), our approach focuses on strengthening the hierarchical structure information. Instead of relying on extensive computational resources, we continue the training process of the model using only annotated data for the fine-tuning task. This strategy enables the model to effectively handle the challenges of hierarchical semantic parsing while requiring fewer computational resources. By adopting this approach, we aim to enhance the model's performance and ensure efficient utilization of available computational capacity.

2.1.3 Grammar-Constrained Neural Network Models

The integration of grammar constraints with deep neural networks has received significant attention in the research community. In a previous work, Yin and Neubig (2017) proposed a neural architecture that incorporates a grammar model to explicitly capture the target syntax of a general-purpose programming language. By encoding the underlying syntax as prior knowledge, this grammar model facilitates the generation of code that adheres to syntactic correctness. Similarly, Xiao et al. (2016) emphasized the integration of grammatical constraints into the RNN-based sequential predictor. This integration allows the predictor to follow the syntactic rules and structures specified by the grammar when generating logical forms. By incorporating these constraints, the model ensures that the generated sequences maintain grammatical accuracy and coherence in line with the underlying grammar. In recent research by Baranowski and Hochgeschwender (2021), the significance of grammar in semantic parsing tasks, particularly in programming or query languages like SQL, was highlighted. By leveraging a context-free grammar, which formalizes the target meaning representations, syntactical constraints can be enforced during the prediction of logical forms. To enforce these grammar constraints, an LR parser (Knuth, 1965), was employed to ensure the va-

lidity of the generated sequences during the decoding phase. The incorporation of grammar constraints in the parsing process guarantees that the produced outputs adhere to the grammatical rules of the target language.

Our approach distinguishes itself from prior research by going beyond using grammar solely for valid label predictions. Instead, we exploit grammar as an additional source of structured information during training by incorporating it into a conditional loss function. This novel aspect enhances the capabilities of our approach and sets it apart from previous works.

2.2 Background Knowledge

2.2.1 Hierarchical Representation

When coming to the design of semantic representation, there are various considerations that need to be taken into account. One of the main challenges is finding the right balance between the expressiveness of the representation and the ease of annotation, parsing, and execution. This trade-off is crucial in order to create effective semantic parsing systems.

In the context of semantic parsing in task-oriented dialog systems, many existing annotation schemes (Mesnil et al., 2013; Zettlemoyer & Collins, 2012) have leaned towards either non-recursive intent or slot tagging. An example of this can be seen in the ATIS dataset (B. Liu & Lane, 2016), where given one utterance, the annotations focus on categorizing one intent and tagging specific slots. These approaches have their advantages, such as simplicity and straightforwardness in the annotation process. However, they also have limitations in terms of capturing more complex and nuanced semantic structures.

Gupta et al. (2018) introduced a hierarchical representation that similarity to a constituency syntax tree, where words serve as terminals. In this structure, non-terminals can either be intents or slots, and the root node is always an intent. By organizing the representation in a hierarchical manner, it becomes possible to capture the relationships and dependencies between different components of a given utterance or query. This allows for a more nuanced and comprehensive understanding of the user's input.

Formally, given the utterance $X = [x_1, x_2, ..., x_m]$ with m tokens. The goal of hierarchical representation is to predict the semantic parse tree $\mathcal{P}_{gold}$ (*e.g.* Figure. 2.1). In this tree, each leaf node represents a token span $x_{i:j} = [x_i, x_{i+1}, ..., x_{j-1}]$ from X. Each non-terminal node has a label l. This label can be either an intent (starting with `"IN:"`) or a slot (starting with `"SL:"`).

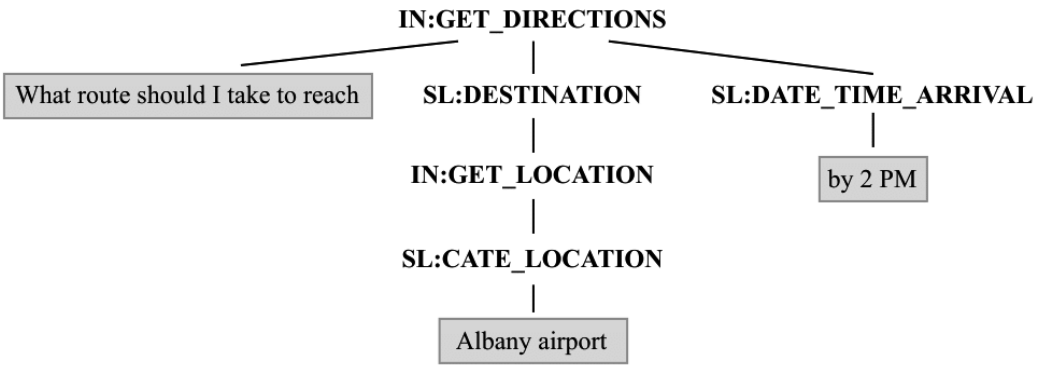


Figure 2.1: An example of hierarchical representation with the input utterance *"What route should I take to reach Albany airport by 2 PM"*.

2.2.2 Recursive Insertion-based Method for Hierarchical Semantic Parsing

The recursive insertion-based method, as outlined in the study by Mansimov and Zhang (2022), involves a recursive and incremental process of constructing sub-parsed trees. This method can be described as the step-by-step generation of a sequence of sub-parsed trees denoted as $\mathcal{P} = [\mathcal{P}_0, \mathcal{P}_1, ..., \mathcal{P}_{gold}]$. The initial sub-parsed tree $\mathcal{P}_0$ is identical to the input utterance X, while each subsequent sub-parsed tree is generated based on the output of the previous step.

An alternative perspective to understanding this method is to view the target tree $\mathcal{P}_{gold}$ as the result of a series of incremental insertions of elements from a set $\mathcal{S}$ = $\{(l_0, s_0, e_0)$,..., (l_i, s_i, e_i) ,..., $(l_T, s_T, e_T)\}$ into the original utterance X. Here, each element (l_i, s_i, e_i) in $\mathcal{S}$ consists of an intent/slot label l_i, a start position s_i, and an end position e_i. The label l_i represents a specific semantic category and covers the token span (s_i, $e_i - 1$) within the partially constructed tree $\mathcal{P}_i$. By successively inserting the elements from $\mathcal{S}$ into X in the specified positions, we arrive at the desired target tree $\mathcal{P}_{gold}$.

A concrete example of the semantic tree generation process using the recursive insertion-based method is shown in Figure 2.2. In the first iteration, the label `IN:GET_DIRECTIONS` is inserted at the beginning of the utterance, spanning positions 0 to 7 in the sentence *"What is the shortest way home ?"*. This insertion operation yields the intermediate tree representation *"`[IN:GET_DIRECTIONS` What is the shortest way home ?]"*. Subsequently, this modified tree is fed back into the model, which generates the tuple (`SL:DESTINATION`, 6, 8) as output. The updated tree, now including the `SL:DESTINATION` label *"`[IN:GET_DIRECTIONS` What is the shortest way `[SL:DESTINATION` home] ?]"*, is then used as input for the model, producing the tuple (`IN:GET_LOCATION_HOME`,

7, 9). Finally, to indicate the completion of the generation process, the model predicts a special end-of-prediction (EOP) label.

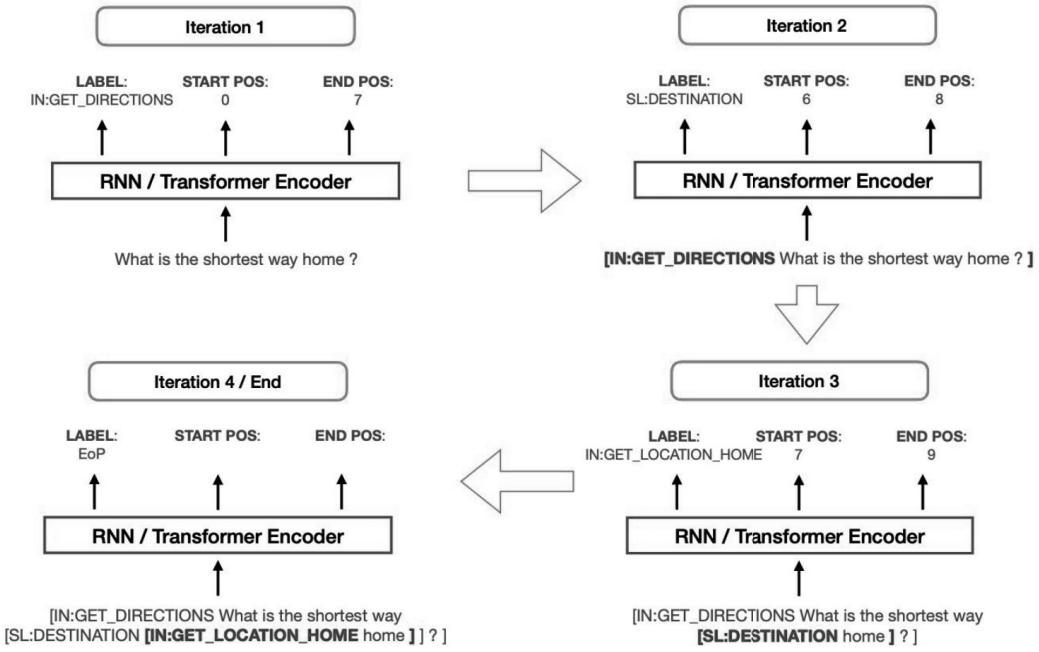


Figure 2.2: An overview of top-down generation of the recursive insertion-based method for hierarchical semantic parsing. The inserted labels at each step are highlighted in red.

The recursive insertion-based method incorporates an encoder component, which can utilize various architectures such as an RNN (Elman, 1990), Transformer (Vaswani et al., 2017), or any other suitable model. The encoder plays a crucial role in processing the input utterance and extracting meaningful representations that can be used for subsequent steps in the parsing process. The choice of encoder architecture depends on the specific requirements of the task and the characteristics of the data being parsed.

Chapter 3

Proposed Method

3.1 Overview

Figure 3.5 provides an overview of our StructSP framework, which consists of two fine-tuning phases: **Structure-aware Boosting** and **Grammar-based RINE**. In the first phase, we aim to enhance the structured information within hierarchical representations, enabling pre-trained language models to adapt more effectively to the semantic parsing task. To achieve this, we propose two novel sub-tasks: Structure-focused Masked Language Modeling (MLM) (detailed in section 3.2.1) and Relative Tree Agreement (described in section 3.2.2). These sub-tasks focus on strengthening the understanding of structured information and improving the ability to capture hierarchical dependencies within the language models.

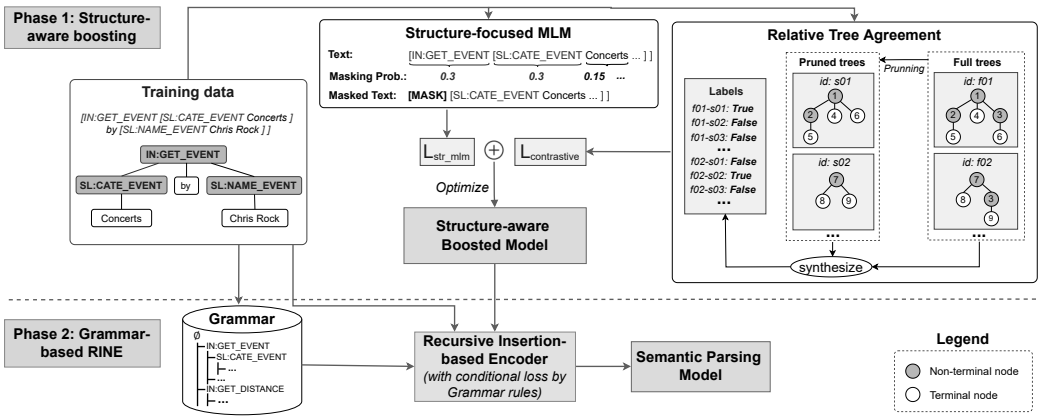


Figure 3.1: System architecture of StructSP framework. Where the *structure-aware boosted model* is the output model from the first phase and is used as the backbone encoder for the second *Grammar-based RINE* phase.

In the second phase, called Grammar-based RINE, we employ a recursive

insertion-based approach (Mansimov & Zhang, 2022) enhanced by grammar constraints to address the problem of the hierarchical semantic parsing task. We synthesize a grammar from the annotated training data, resulting in an inductive grammar with the root represented by the special label $\emptyset$. Utilizing the structure-aware model obtained from the previous phase as the backbone encoder, we introduce a constrained objective function using the synthesized grammar. This enables the model to focus on crucial information for promising label predictions. During the inference phase, we further utilize the grammar to guide the decoding process and prevent unpromising decoding directions, improving the overall accuracy and effectiveness of the framework.

3.2 Structure-aware Boosting

The structure-aware boosting phase consists of two sub-tasks: *Structure-focused MLM* and *Relative Tree Agreement*, which we will discuss in detail in this section.

3.2.1 Structure-focused MLM

Before delving into the proposed approach of structure-focused MLM, we first recap the concept of vanilla MLM (Devlin et al., 2019).

Vanilla MLM: In the field of pre-trained language models, the masked language modeling technique is a crucial component for enhancing the proficiency of language understanding. The main objective of this task is to train the language model by predicting masked tokens based on the contextual information provided by the input sequence. Mathematically, given an input sequence $\mathcal{X} = [x_1, x_2, ..., x_m]$, a masking distribution $T = [t_1, t_2, ..., t_m]$ is applied, where t_i represents the probability of masking token x_i. In the vanilla MLM approach proposed by Devlin et al. (2019), all t_i values in T are set equally to 15%. Let $\mathcal{D}$ denote the set of tokens selected for masking, e.g., $\mathcal{D} = \{x_1\}$, and let $\mathcal{X}'$ represent the masked sequence $\mathcal{X}' = [\texttt{[MASK]}, x_2, ..., x_m]$. The objective function for vanilla MLM is defined as follows:

$$\mathcal{L}_{mlm} = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{V}|} x_{ij} \log p(x_{ij} \mid \mathcal{X}') \tag{3.1}$$

Here, $\mathcal{L}_{mlm}$ denotes the specific loss function employed for the vanilla MLM task. The vocabulary set $\mathcal{V}$ encompasses all possible tokens that the language model can generate. The conditional probability $p(x_{ij} \mid \mathcal{X}')$ measures the likelihood that the

masked token x_i is predicted as the j-th token in the vocabulary, given the context $\mathcal{X}'$. By minimizing the average negative log-likelihood across all masked tokens, the model learns to accurately predict the original tokens, leveraging the contextual information provided by the surrounding tokens. The vanilla MLM approach has proven to be effective (Lee et al., 2020; Nguyen & Nguyen, 2021). However, when applying the masked language model to the task of semantic parsing, there is room for further improvement by leveraging the characteristics of this specific task.

Structure-focused MLM: In the linearized hierarchical representation, we encounter two types of tokens: normal tokens and logical tokens (Figure 3.2). Normal tokens correspond to the natural language text of the utterance, while logical tokens encode the structural information of the semantic representation. The logical tokens can be further categorized into two subtypes: label tokens and bracket tokens. Label tokens represent specific label types, such as intents or slots, for instance `SL:NAME_EVENT`, while bracket tokens indicate the corresponding spans of labels. When we input the linearized hierarchical representation into a masked language model, the correct prediction of label tokens demonstrates the model's understanding of label types, while the correct prediction of bracket tokens demonstrates the model's comprehension of label spans. Therefore, masking logical tokens is an effective approach to help the model learn the structured information within the hierarchical representation (Bai et al., 2022). However, vanilla MLM treats all tokens equally, without considering whether they are logical or normal tokens.

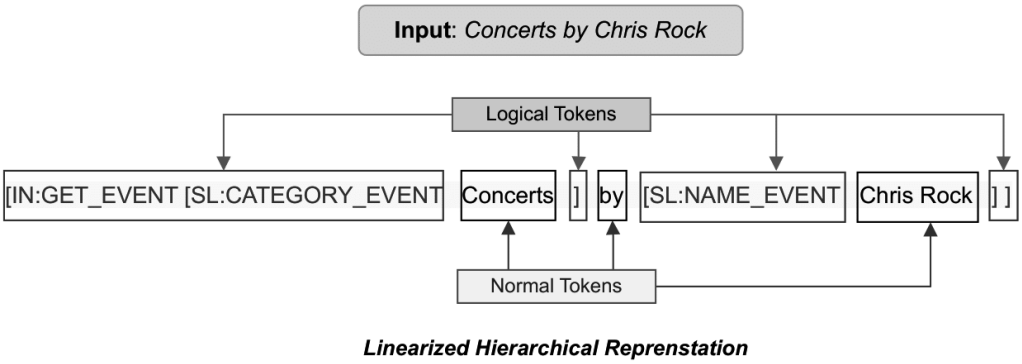


Figure 3.2: An example of linearized hierarchical representation. Logical tokens are highlighted in red, and normal tokens are in black.

In the structure-focused MLM approach, we assign a higher masking probability to logical tokens, aiming to prioritize the hierarchical structure information

within language models. This mechanism allows the models to acquire knowledge from the semantic structure of utterances while retaining contextual understanding based on normal masked tokens (Yu, Wu, et al., 2020). Specifically, we introduce a new masking distribution $T' = [t'_1, t'_2, ..., t'_m]$, where $t'_i = \alpha$ if token x_i corresponds to a logical token. The choice of a suitable value for α is crucial, and we conduct an experiment to determine its optimal value (see section 5.2). During training, we use the structure-focused MLM loss function ($\mathcal{L}_{str\_mlm}$) to compute the loss for masked tokens, following a similar approach to the original MLM. Using the values of T', we randomly select a set of tokens $\mathcal{D}'$ for masking, and the model is trained to predict the masked tokens based on the contextual information provided by the unmasked tokens. The objective is to minimize the negative log-likelihood of the predicted tokens, as shown in the following equation:

$$L_{str\_mlm} = -\frac{1}{|\mathcal{D}'|} \sum_{i=1}^{|\mathcal{D}'|} \sum_{j=1}^{|\mathcal{V}|} x_{ij} \log p(x_{ij} \mid \mathcal{X}') \tag{3.2}$$

Similar to the masking strategy employed in RoBERTa (Y. Liu et al., 2019a), we perform token masking during each training iteration rather than in the preprocessing step. This approach ensures that the model learns to handle the presence of masked tokens dynamically.

3.2.2 Relative Tree Agreement

In the tree format of hierarchical representation, the structure is organized such that each leaf node corresponds to a token span, while non-terminal nodes represent labels indicating intents or slots. We construct a *non-terminal list* by performing a breadth-first search traversal of the parsed tree. The non-terminal list consists of all the non-terminal nodes encountered during the traversal. Figure 3.3 provides an example of such a non-terminal list. This list will then be used in generating positive training samples for the relative tree agreement subtask.

The parsed tree that includes all the non-terminal nodes is referred to as the *full tree* or *gold tree*. In the subsequent steps, for each non-terminal node in the non-terminal list, we create a *pruned tree* by removing all the non-terminal nodes located to the right of that particular node in the list from the full tree. Consequently, a full tree can have multiple pruned trees. These pruned trees and the full tree are considered *relative* to each other. An example of pruned trees and a full tree is presented in Figure 3.4. To construct positive training samples, we randomly select one of the pruned trees, denoted as $\mathcal{P}_{pruned}$, and combine it with the full tree, denoted as $\mathcal{P}_{full}$, resulting in a positive training sample represented as ($\mathcal{P}_{pruned}$, $\mathcal{P}_{full}$).

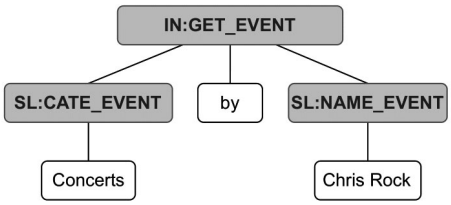


Figure 3.3: An example of non-terminal list according to the hierarchical representation of utterance *"Concerts by Chris Rock"*.

The objective of this task is to ensure that the representations of relative trees are closely aligned in the embedding space. This alignment facilitates the effective utilization of recursive insertion-based parsing. To achieve this objective, we begin by encoding the pruned tree $\mathcal{P}_{pruned}$ and the full tree $\mathcal{P}_{full}$ to obtain their corresponding hidden states, denoted as h_{pruned} and h_{full}, respectively.

$$\begin{aligned} h_{full} &= \text{RoBERTa}(\mathcal{P}_{full}) \\ h_{pruned} &= \text{RoBERTa}(\mathcal{P}_{pruned}) \end{aligned} \tag{3.3}$$

To train the model for this task, we leverage the power of contrastive learning techniques (Frosst et al., 2019; Gao et al., 2021; Bai et al., 2022; Luo et al., 2022). The primary objective of contrastive learning is to bring similar instances closer together in the embedding space, minimizing their distances, while simultaneously pushing dissimilar instances apart by maximizing their distances. In our case, at the i-th position of the batch B, the positive pair consists of $h^{(i)}_{full}$ and $h^{(i)}_{pruned}$, representing the hidden states of the full tree and the pruned tree, respectively. The negative pairs are formed by combining $h^{(i)}_{full}$ with the hidden states of the other samples in the batch. After obtaining the hidden state of each tree, we can compute the similarity between each positive or negative training pair, which measures how well the representations of the two trees align in the embedding space. To encourage closer alignment between the pruned tree and the full tree in the same parsing process, we employ a contrastive loss function, denoted as $\mathcal{L}_{contrastive}$, defined as follows:

$$\mathcal{L}_{contrastive} = -\log \frac{\exp(\text{sim}(h^{(i)}_{full}, h^{(i)}_{pruned})/\tau)}{\sum_{j \in B} \exp(\text{sim}(h^{(i)}_{full}, h^{(j)}_{pruned})/\tau)} \tag{3.4}$$

In this equation, $sim(h^{(i)}_{full},\ h^{(i)}_{pruned})$ represents the similarity score between the i-th pair of hidden states based on cosine distance, τ is a temperature parameter (Gao et al., 2021), and B denotes the batch of training examples.

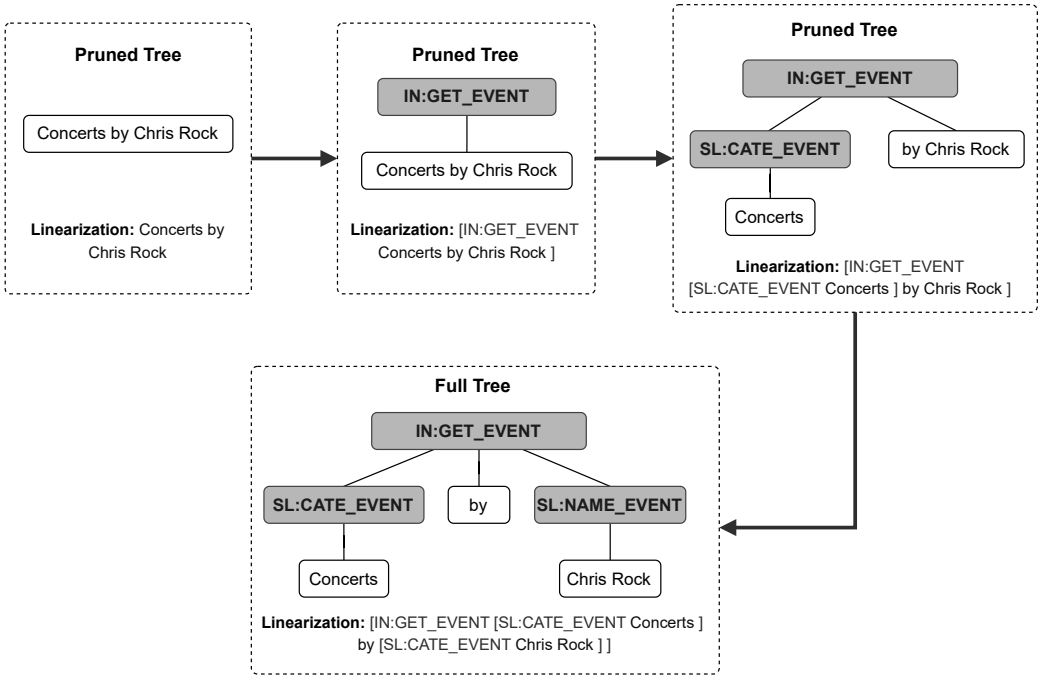


Figure 3.4: An example of pruned trees and the full tree in the parsing process of utterance *"Concerts by Chris Rock"*.

3.2.3 Objective Function

We integrate the losses from the two above sub-tasks to formulate the final objective functions for training our model.

$$\mathcal{L}_{sab} = \mathcal{L}_{contrastive} + \lambda * \mathcal{L}_{str\_mlm} \tag{3.5}$$

In this equation, the hyperparameter λ is used to balance the contributions of the two sub-tasks. By adjusting the value of λ, we can control the relative importance of the contrastive learning and structure-based MLM objectives during training. We follow the approach of previous works (Lee et al., 2020; Nandy et al., 2021; Bai et al., 2022) by initializing our model with the weights of a pre-trained language model (Y. Liu et al., 2019a). This initialization helps our model leverage prior knowledge stored in the language model.

3.3 Grammar-based RINE

In the grammar-based RINE phase, we utilize the recursive-insertion based approach (Mansimov & Zhang, 2022) enhanced by grammar constraints. The backbone encoder used in this phase is the structure-aware model from the previous

phase, which has learned to capture and encode hierarchical representations. The parsing process in the recursive-insertion based approach can be represented as an incremental generation of sub-parsed trees, where the output of the previous step serves as the input for the current step. Table 3.1 illustrates an example of the parsing process.

Table 3.1: Example chain of incremental trees in parsing process using Recursive Insertion-based Encoder.

Tree	**Linearized representation**
$\mathcal{P}_0$	Concerts by Chris Rock
$\mathcal{P}_1$	[IN:GET_EVENT Concerts by Chris Rock]
$\mathcal{P}_2$	[IN:GET_EVENT [SL:CATE_EVENT Concerts] by Chris Rock]
$\mathcal{P}_3$	[IN:GET_EVENT [SL:CATE_EVENT Concerts] by [SL:NAME_EVENT Chris Rock]]

3.4 Grammar Constraints

It has been observed that the relationships between nodes in hierarchical representations typically adhere to a grammar. This knowledge is crucial in hierarchical semantic parsing as it can be leveraged to correct decoding steps and prevent unreliable label predictions. For instance, in the case of a parent node labeled as `IN:GET_EVENT`, its child nodes should correspond to slots containing event information, such as `SL:NAME_EVENT` or `SL:CATE_EVENT`. Recognizing the importance of grammar in improving parsing performance, we introduce grammar integration into our model (section 3.4.1).

To incorporate grammar constraints, we synthesize a grammar $\mathcal{G} = \{A \rightarrow B \mid A, B \text{ are non-terminal nodes } \}$ from annotated training data, as illustrated in Figure 3.5. This synthesized grammar captures the constraints and dependencies observed in the training data. For example, one of the constraints in the grammar could be `IN:GET_EVENT` $\rightarrow$ `IN:NAME_EVENT`. By employing this grammar in our method, at each parsing step, the model only needs to consider a candidate set $\mathcal{C} = \mathcal{G}(U)$ instead of evaluating all possible label types. Here, U represents the parent node of the label that is currently being predicted.

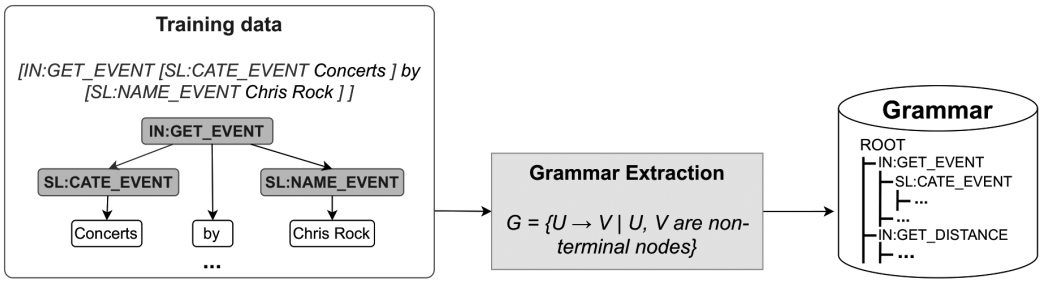


Figure 3.5: Process of using annotated training data to synthesize the grammar.

3.4.1 Modeling

Training: Without loss of generality, we consider the i^{th} step of the parsing process. At this point, the input consists of the linearized representation of the current parsed tree $\mathcal{P}_i = [x_1, x_2, ..., x_n]$ and the parent node U. The objective of the model is to predict a label from the label set $L = [l_1, l_2, ..., l_{|L|}]$ and its corresponding span (s, e), where both s and e are within the range $[0, n)$.

During the training phase, we begin by inputting the parsed tree $\mathcal{P}_i$ into the encoder model to acquire its hidden states. Following previous works (Mansimov & Zhang, 2022), we utilize the vector representation of the hidden state corresponding to the `[CLS]` token to compute the probability of the node label. Additionally, we utilize the hidden states from the last two transformer encoder layers to calculate the probabilities of the start and end positions (s, e).

$$p^{nodeLb} = \text{softmax}(W_{lb} h^{(t)}_{[CLS]} + b_{lb}) \tag{3.6}$$

$$p^{start}_k = \text{softmax}(W_s h^{(t)}_{w_k} + b_s) \tag{3.7}$$

$$p^{end}_k = \text{softmax}(W_e h^{(t-1)}_{w_k} + b_e) \tag{3.8}$$

In these equations, W_{lb}, W_s, and W_e are weight matrices, b_{lb}, b_s, and b_e are bias, $h^{(t)}_{[CLS]}$ represents the hidden state of the `[CLS]` token in the t^{th} encoder layer, and $h^{(t)}_{w_k}$ and $h^{(t-1)}_{w_k}$ represent the hidden states corresponding to the k^{th} word in the linearized representation of the parsed tree in the t^{th} and $(t-1)^{th}$ encoder layers, respectively. Especially, we integrate a grammar-based penalty into the loss function to ignore unpromising node label predictions. The penalty is defined as follows:

$$s^{penalty} = \begin{cases} 0 & \text{if} \quad nodeLb \in \mathcal{C} \\ +\infty & \text{otherwise} \end{cases} \tag{3.9}$$

Here, $\mathcal{C}$ denotes the canditates that is obtained by the grammar $\mathcal{G}$ and parent node U ($\mathcal{C} = \mathcal{G}(U)$). The final loss functions are computed using the cross-entropy loss as follows:

$$\mathcal{L}_{nodeLb} = \text{CE}(g^{nodeLb}, p^{nodeLb}) + s^{penalty} \tag{3.10}$$

$$\mathcal{L}_{start} = \sum_{k} \text{CE}(g_k^{start}, p_k^{start}) \tag{3.11}$$

$$\mathcal{L}_{end} = \sum_{k} \text{CE}(g_k^{end}, p_k^{end}) \tag{3.12}$$

$$\mathcal{L}_{final} = \mathcal{L}_{nodeLb} + \mathcal{L}_{start} + \mathcal{L}_{end} \tag{3.13}$$

Where, g^{nodeLb} represents the ground-truth node label, g_k^{start} and g_k^{end} are the ground-truth start and end positions. The loss function $\mathcal{L}_{nodeLb}$ combines the cross-entropy loss between the predicted node label p^{nodeLb} and the ground truth, with the grammar-based penalty $s^{penalty}$. Similarly, $\mathcal{L}_{start}$ and $\mathcal{L}_{end}$ compute the cross-entropy losses between the predicted start and end positions p_k^{start} and p_k^{end}, respectively, and their corresponding ground truth.

Inference: During the inference step, we employ the recursive-insertion based approach to generate the output parsed tree. However, a challenge arises in this step due to the lack of information about the current parent node U, which is required to generate the candidate set $\mathcal{C}$. To tackle this problem, we adopt a straightforward strategy: we first predict the label span and then use the predicted label span to identify the current parent node. For example, let's consider the beginning of the third step of the parsing process in Table 3.1, with the input $\mathcal{P}_2$. At this moment, we do not have any information about the parent node, which could be either `IN:GET_EVENT` or `IN:CATE_EVENT`. To determine the parent node, we run the span prediction and obtain the span *"Chris Rock"*. Based on the position of this span, we can identify that the parent node should be `IN:GET_EVENT`. Utilizing this information about the parent node, we generate the candidate set $\mathcal{C}$ based on the grammar $\mathcal{G}$. Next, we proceed with the model to obtain the current label predictions. We then prune all label predictions that do not belong to the candidate set $\mathcal{C}$ by setting the probability of these label predictions to zero, i.e., $c \notin \mathcal{C}$. This pruning step ensures that only labels consistent with the grammar constraints are considered.

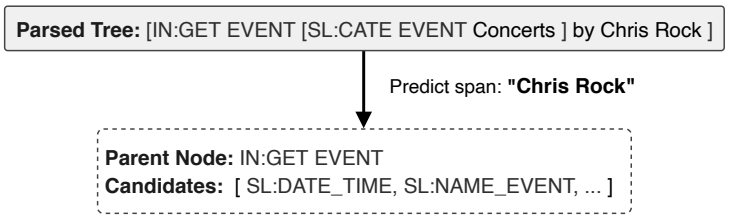


Figure 3.6: Using the span prediction to determine the parent node.

Chapter 4

Experiment

4.1 Datasets and Evaluation Metric

To evaluate the performance of our StructSP model, we conducted experiments on the TOP dataset[1] (Gupta et al., 2018) and its variant, TOPv2[2] (X. Chen et al., 2020). These datasets provide a diverse range of semantic parsing queries across different domains. Additionally, we employed a standard evaluation metric to measure the effectiveness of our model's predictions. The datasets and evaluation metric are described as follows:

TOP: The TOP dataset serves as the primary dataset for our experiments. It consists of training, development, and test sets, with statistics presented in Table 4.1. The dataset covers two domains: **navigation** and **event**. It offers a rich variety of semantic parsing task, with 29 distinct intents and 36 different slots, enabling a comprehensive evaluation of our model's semantic parsing capabilities. To ensure consistency with previous works (Einolghozati et al., 2019; Rongali et al., 2020; Zhu et al., 2020; Mansimov & Zhang, 2022), we followed their setting by removing all utterances containing the `UNSUPPORTED` intent from the dataset. This preprocessing step resulted in the TOP dataset containing 28,14 training examples, 4032 development examples, and 8241 test examples.

TOPv2 In addition to the TOP dataset, we also utilized the TOPv2 dataset to further evaluate the performance of our proposed model, particularly in low-resource scenarios. The TOPv2 dataset serves as an extension of the TOP dataset and introduces two new domains: **weather** and **reminder**. Within each domain, the TOPv2 dataset offers two different settings: 25 SPIS and 500 SPIS. Here, *"SPIS"*

[1] Provided under the CC-BY-SA license
[2] Provided under the CC BY-NC 4.0 license

Table 4.1: Statistics of TOP and TOPv2 datasets.

Dataset	Domain	Setting	#Train	#Dev	#Test	#Intents	#Slots
TOP	_	_	28414	4032	8241	29	36
TOPv2	weather	25 SPIS	176	147	5767	7	11
		500 SPIS	2372	1202	5767		
	reminder	25 SPIS	493	337	5682	19	32
		500 SPIS	4788	1871	5682		

stands for *samples per intent and slot*. In the case where an intent or slot appears fewer times than the value of SPIS, the utterances containing that intent or slot are included in the training dataset. In simpler terms, a smaller SPIS setting corresponds to a smaller training dataset. For the weather domain in the TOPv2 dataset, there are 7 distinct intents and 11 different slots. While the reminder domain consists of 19 intents and 32 slots. The statistic of data samples is shown in Table 4.1.

Evaluation Metric: For evaluating the performance of our model, we employed the exact match (EM) score as the evaluation metric following the previous studies (Zhu et al., 2020; Rongali et al., 2020; Mansimov & Zhang, 2022). The EM score measures the percentage of predictions that perfectly match the ground truth. It provides a strict evaluation criterion, reflecting the accuracy of the model in generating correct semantic representations (parsed trees).

4.2 Data Preprocessing and Experimental Setting

4.2.1 Data Pre-processing

To prepare the data for training our StructSP model, we followed a series of preprocessing steps. These steps ensured the extraction and transformation of the necessary information from the training samples. The data preprocessing steps are as follows:

- **Step 1: Extraction of Grammar.** In the first step, we extracted the grammar by utilizing the full trees from the training samples. We followed the instruction $\mathcal{G} = \{A \rightarrow B \mid A, B \text{ are non-terminal nodes}\}$ to create the grammar rules. For example, a grammar rule could be represented as

IN:GET_EVENT → SL:CATE_EVENT. This extraction process enabled us to build an inductive grammar with the root denoted by ROOT label.

- **Step 2: Conversion of Ground-Truth Trees.** Next, we converted the ground-truth full trees into multiple sub-parsed trees. Each sub-parsed tree was represented as a triple, consisting of the linearized representation of the current parsed tree, the label associated with that tree, and the corresponding label span. Figure 1.3 provides an example illustrating this transformation.

- **Step 3: Extraction of Label Types.** In the final step, we extracted the set of label types from the training samples. By extracting the labels present in the data, we obtained a collection of the different label types that appeared in the dataset. This information was crucial for training our model to recognize and generate appropriate labels during the decoding phase.

By following these data preprocessing steps, we prepared the dataset to train our models in two fine-tuning phases: **Structure-aware Boosting** and **Grammar-based RINE**.

4.2.2 Experimental Setting

In the training process, we utilized various hyperparameters for our proposed models. The hyperparameters used for each phase of our StructSP method when performing the TOP dataset are presented in Table 4.2.

Structure-aware Boosting: In the Structure-aware Boosting phase, we utilized the Roberta model (Y. Liu et al., 2019b) as our backbone encoder. To optimize our model's performance, we carefully selected hyperparameters. We set the batch size to 16 and the learning rate to 1e-5. The sequence length was set to 300, allowing for sufficient context coverage. Additionally, we performed a hyperparameter search for α, which represents the weighting factor for the structure-focused MLM and relative tree agreement loss functions. We evaluated different values of α and found that the best performance was achieved with values of **0.5** or 1.0, with the **0.5** value producing the most balanced contributions between the two loss functions. The training process was conducted for 10 epochs. In the TOPv2 datasets, we made adjustments to the number of training epochs for the 25 SPIS settings. Since the training data size is smaller in these settings, we increased the number of training epochs to 50. This modification allowed our model to benefit from extended training on the limited data available in low-resource scenarios.

Table 4.2: Hyper-parameters of our models in TOP dataset

Phase	Hyper-Parameter	Value
	Batch size	16
Structure	Learning Rate	1e-5
Aware	Sequence length	512
Boosting	MLM Weight (θ)	0.5
	Training Epoch	10
	Batch Size	32
	Optimizer	Adam
	Learning Rate	1e-5
Grammar	Warmup Step	1000
based	Max Training Epoch	50
RINE	Max length	512
	Logical Token Masking Prob. (α)	0.3
	Attention Dropout	0.2
	MLP Dropout	0.5

Grammar-based RINE: For the Grammar-based RINE phase, we fine-tuned our model by adjusting several key hyperparameters. We set the batch size to 32 and utilized the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1e-5. To ensure stable training, we applied a warmup step of 1000 and limited the maximum training epoch to 50. Considering the maximum length of inputs, we set it to 512. To control the logical token masking probability (α), which is crucial for capturing logical structure information, we conducted a hyperparameter search. We evaluated different values of α from the set $\{0.2, \mathbf{0.3}, 0.4, 0.5, 0.6, 0.7\}$, and the value of **0.3** yielded the highest performance, highlighting its effectiveness in balancing the masking of logical tokens. To control the logical token masking probability (α), which is crucial for capturing logical structure information, we conducted a hyperparameter search. We evaluated different values of α from the set $\{0.2, \mathbf{0.3}, 0.4, 0.5, 0.6, 0.7\}$, and the value of **0.3** yielded the highest performance, highlighting its effectiveness in balancing the masking of logical tokens. In the TOPv2 datasets, we made further adjustments to the number of training epochs specifically for the 25 SPIS settings. In these low-resource settings, we extended the training duration to 100 epochs.

4.3 Main Results

Performance on TOP dataset: We conducted a performance comparison of our StructSP method with several previous approaches on the TOP test set. Table 4.3 presents the results in terms of the exact match (EM) score.

Table 4.3: Performance comparison using exact match score for our StructSP method and previous works on TOP test set.

Method	**Pre-trained**	**EM**
Seq2seq (Bai et al., 2022)	SARA-ROBERTA	82.78
RNNG ensem. + SVMRank (Einolghozati et al., 2019)	ELMo	87.25
Non-AR S2S-Ptr (Shrivastava et al., 2021)	RoBERTa$_{base}$	85.07
S2S-Ptr (Rongali et al., 2020)	RoBERTa$_{base}$	86.67
Decoupled S2S-Ptr (Aghajanyan et al., 2020)	RoBERTa$_{large}$	87.10
Insertion Transformer + S2S-Ptr (Zhu et al., 2020)	RoBERTa$_{base}$	86.74
RINE (Mansimov & Zhang, 2022)	RoBERTa$_{large}$	87.57 ± 0.03
Non-grammar StructSP (ours)	RoBERTa$_{large}$	87.89 ± 0.08
StructSP (ours)	RoBERTa$_{large}$	**88.18 ± 0.24**

Our method outperforms all previous methods, achieving an impressive EM score of 88.18 ± 0.24. Specifically, it achieved a higher EM score than the RNNG model ensemble with ranking SVM (Einolghozati et al., 2019) by 0.93 EM, the non-autoregressive seq2seq model with pointer (Shrivastava et al., 2021) by 3.11 EM, the autoregressive seq2seq model with pointer (Rongali et al., 2020) by 1.51 EM, and the decoupled seq2seq model with pointer (Aghajanyan et al., 2020) by 1.08 EM. Among insertion-based parsing methods, our proposed model outperforms the seq2seq-ptr model based on the Insertion Transformer (Zhu et al., 2020) by 1.44 EM, and the current SOTA model, RINE (Mansimov & Zhang, 2022), by 0.61 EM. Furthermore, our results outperformed those of the Seq2seq model based on SARA-ROBERTA (Bai et al., 2022) by a significant margin (5.40 EM), despite their efforts to enhance the structural information of pre-trained models through general Abstract Meaning Representation (AMR) structures.

The results we obtained clearly demonstrate the effectiveness of our approach, which introduces a novel injecting mechanism for integrating hierarchical structure information from natural sentences into pre-trained language models. Moreover, we observed a notable improvement in our model's performance when we incorporated grammar into the training and inference processes. Specifically, our model that utilized grammar achieved a 0.29-point higher score compared to a version that did not utilize grammar. This highlights the importance of incorporating

grammar to guide the decoding process and avoid unpromising decoding directions when dealing with the TOP dataset. These findings underscore the value and impact of incorporating grammar in improving the accuracy and reliability of semantic parsing models.

Performance on TOPv2 dataset: We present the main results of our experiments on the TOPv2 dataset, comparing the performance of our proposed StructSP method with previous approaches in Table 4.4.

Table 4.4: Performance comparison using exact match score for our StructSP method and previous works on TOPv2 test set.

Method	**Pre-trained**	**Exact Match**			
		Weather		**Reminder**	
		25 SPIS	**500 SPIS**	**25 SPIS**	**500 SPIS**
LSTM Seq2Seq-Ptr (X. Chen et al., 2020)	-	46.2	78.6	21.5	65.9
Seq2seq-Ptr (X. Chen et al., 2020)	RoBERTa$_{base}$	-	83.5	-	71.9
Seq2seq-Ptr (X. Chen et al., 2020)	BART$_{large}$	71.6	84.9	55.7	71.9
RINE (Mansimov & Zhang, 2022)	RoBERTa$_{base}$	74.53 ± 0.86	87.80 ± 0.04	68.71 ± 0.46	80.30 ± 0.04
RINE (Mansimov & Zhang, 2022)	RoBERTa$_{large}$	77.03 ± 0.16	87.50 ± 0.28	71.10 ± 0.63	81.31 ± 0.22
Non-grammar StructSP	RoBERTa$_{large}$	**78.24 ± 0.47**	88.00 ± 0.47	72.07 ± 1.24	81.57 ± 0.27
StructSP	RoBERTa$_{large}$	77.96 ± 0.92	**88.08 ± 0.11**	**72.12 ± 1.13**	**82.28 ± 0.24**

The experimental results indicate that our proposed StructSP method outperforms previous approaches across all SPIS settings. Specifically, at 25 SPIS settings, our models achieve higher scores than the state-of-the-art RINE model (Mansimov & Zhang, 2022) by 0.93 EM in the weather domain and 1.02 EM in the reminder domain. At 50 SPIS settings, our models surpass the performance of the RINE model by 0.58 EM and 0.97 EM, respectively. These findings highlight the effectiveness of our proposed method, which demonstrates significant improvements even in low-resource scenarios.

Furthermore, it is worth noting that our model without using grammar performs better than the model that incorporates grammar at 25 SPIS in the weather domain. We attribute this difference to the extremely limited training data with

only 176 samples available at 25 SPIS settings. Consequently, the grammar extracted from the training data may not be comprehensive enough to capture the grammar patterns present in the validation and test sets. Addressing this issue requires further investigation and future work.

Chapter 5

Analysis

5.1 Ablation Study

To evaluate the impact of different factors on the performance of our proposed model, we conducted a systematic ablation study. The goal was to analyze and assess the contributions of individual components in our model, identifying the key elements that significantly influenced its performance. Initially, we trained the RINE baseline model described in the original paper by Mansimov and Zhang (2022) without any additional techniques or modifications. This baseline model served as a reference for comparing subsequent variations. Next, we conducted a series of ablations on our proposed model, systematically removing or altering specific components to observe their effects. The results of this ablation study, shown in Table 5.1, are as follows:

- In the first row, we evaluated the full-setting model, which included both the Structure-aware boosting and Grammar-based RINE phases. This configuration achieved an impressive exact match (EM) score of 88.26, significantly outperforming the baseline model according to the T-test ($p < 0.05$).
- Moving to the second row, we investigated the impact of each individual component by selectively disabling them. When both structure-focused MLM and relative tree agreement were disabled, while grammar was still employed, the model achieved an EM score of 87.69, representing a significant decrease of 0.57 compared to the full-setting model. In addition, the T-test revealed that this configuration was not significantly better than the baseline model ($p > 0.05$).
- Continuing to the third row, when grammar was disabled while both structure-focused MLM and Relative tree agreement were enabled, the EM score

Table 5.1: The ablation study results on the validation set of the TOP dataset are shown. The symbols ✓ and ✗ represent whether the corresponding component was included or excluded, respectively. The symbol Δ indicates the difference in scores between the full-setting model and other models.

ID	Method	Settings			EM	Δ	T-test (Significantly better at 95%?)
		Structure-aware boosting		Grammar			
		Structure focused MLM	Relative tree agreement				
(1)	StructSP	✓	✓	✓	**88.26**	-	**yes**
(2)		✗	✗	✓	87.69	-0.57	no
(3)		✓	✓	✗	88.09	-0.17	**yes**
(4)		✓	✗	✗	87.93	-0.33	**yes**
(5)		✗	✓	✗	87.62	-0.64	no
(6)	Baseline	✗	✗	✗	87.57	-0.69	-

slightly decreased to 88.09, indicating a drop in performance. However, this configuration was significantly better than the baseline model.

- Furthermore, in the fourth and fifth rows, we examined the isolated influence of each component in the structure-aware boosting phase. Enabling only structure-focused MLM resulted in an EM score of 87.93, a decrease of 0.33 compared to the full-setting model. This configuration was significantly better than the baseline model. Similarly, enabling only Relative tree agreement led to an EM score of 87.62, a decrease of 0.64 compared to the full-setting model. However, this setup was not significantly better than the baseline model.

The baseline model, without any of the mentioned components enabled, achieved an EM score of 87.57, serving as the reference point for comparison. Overall, the results of the ablation study underscored the importance of structure-focused MLM and relative tree agreement in the structure-aware boosting phase, as well as the incorporation of grammar in the grammar-based RINE phase. These factors contributed to enhanced performance on the TOP dataset.

5.2 Impact of Masking Probability α

In this section, we investigate the impact of the masking probability α on the performance of our model. The masking probability α represents the likelihood

of randomly masking logical tokens during the training process (section 3.2.2). To evaluate its effect, we conducted a series of experiments where we varied the value of α and measured the model's performance using the exact match (EM) metric.

Figure 5.1 illustrates the results of our experiments. The x-axis represents the different values of α ranging from 0.2 to 0.6, while the y-axis represents the EM scores achieved by our model and the baseline RINE model. As we analyze the graph, we observe an interesting trend. Initially, as α increases from 0.2 to 0.3, the EM score of our model shows a gradual improvement, surpassing the baseline performance. However, beyond $\alpha = 0.3$, further increasing the masking probability leads to a decline in the EM score, indicating a decrease in the model's performance. This suggests the importance of finding the right balance point between preserving the original logical tokens and introducing masked logical tokens during pretraining.

In addition, comparing our model's performance to the baseline RINE model, we consistently achieve higher EM scores across all α values. This demonstrates the effectiveness of our proposed approach in capturing the underlying semantics of logical tokens. In conclusion, our experiments highlight the significance of the masking probability α in training our model. The results indicate that a moderate value of α, around 0.25, yields the best trade-off between preserving logical tokens and enhancing the model's performance, as measured by the EM metric.

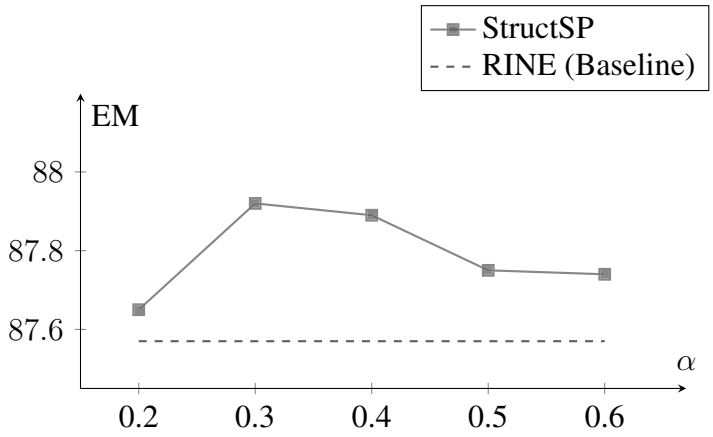


Figure 5.1: Effect of logical-token masking probability (α) on system performance.

5.3 Case Study

To further evaluate the performance of our proposed StructSP model, we conducted a case study comparing its outputs with those of the baseline RINE model. We focused on the validation set of the TOP dataset and examined the outputs for various input queries. Table 5.2 presents the comparison between the outputs of the baseline model and our StructSP model (the outputs in a tree format are shown in Figure 5.2). This table includes three examples, each consisting of the input query, the ground-truth logical form, the output of the baseline model, and the output of our StructSP model.

Table 5.2: Case study results of our StructSP model against the baseline model (RINE) using the validation set from the TOP dataset.

Type	**Ouput**
Input	Where is the nearest Tom Thumb
Ground-Truth	[IN:GET_LOCATION Where is the [SL:LOCATION_MODIFIER nearest] [SL:POINT_ON_MAP Tom Thumb]]
Baseline	[IN:GET_LOCATION Where is the [SL:LOCATION_MODIFIER nearest] [SL:NAME_EVENT Tom Thumb]] ✗
StructSP	[IN:GET_LOCATION Where is the [SL:LOCATION_MODIFIER nearest] [SL:POINT_ON_MAP Tom Thumb]] ✓
Input	What to do after a Pacers game
Ground-Truth	[IN:GET_EVENT What to do [SL:DATE_TIME [IN:GET_EVENT after a [SL:NAME_EVENT Pacers] [SL:CATEGORY_EVENT game]]]]
Baseline	[IN:GET_EVENT What to do after a [SL:NAME_EVENT Pacers] [SL:CATEGORY_EVENT game]]]] ✗
StructSP	[IN:GET_EVENT What to do [SL:DATE_TIME [IN:GET_EVENT after a [SL:NAME_EVENT Pacers] [SL:CATEGORY_EVENT game]]]] ✓
Input	traffic near me right now
Ground-Truth	[IN:GET_INFO_TRAFFIC traffic [SL:LOCATION [IN:GET_LOCATION [SL:LOCATION_MODIFIER [IN:GET_LOCATION [SL:SEARCH_RADIUS near] [SL:LOCATION_USER me]]]]] [SL:DATE_TIME right now]]
Baseline	[IN:GET_INFO_TRAFFIC traffic [SL:LOCATION [IN:GET_LOCATION [SL:SEARCH_RADIUS near][SL:LOCATION_USER me]]] [SL:DATE_TIME right now]] ✗
StructSP	[IN:GET_INFO_TRAFFIC traffic [SL:LOCATION [IN:GET_LOCATION [SL:SEARCH_RADIUS near][SL:LOCATION_USER me]]] [SL:DATE_TIME right now]] ✗

In the first example, the input query is *"Where is the nearest Tom Thumb."* The baseline model fails to generate the correct logical form, as it mistakenly includes the category of the event (`SL:NAME_EVENT`) instead of correctly capturing the point on the map (`SL:POINT_ON_MAP`). In contrast, our StructSP model correctly identifies the point on the map, resulting in an accurate logical form.

This difference can be explained by the fact that the grammar used for extraction doesn't include the constraint (IN:GET_LOCATION => SL:NAME_EVENT). This highlights the effectiveness of incorporating grammar into our model.

In the second example, the input query is *"What to do after a Pacers game."*. In this query, the parsing models need to recognize the span "following a Pacers game" as a date-time slot (SL:DATE_TIME) and accurately interpret the underlying structure within that specific slot. Our model successfully produces the correct output, while the baseline model fails to do so.

Lastly, the third example involves the input query *"traffic near me right now."* This is a particularly challenging example where the models need to predict the ground truth parsed tree with a depth of 5. Unfortunately, both the baseline and our StructSP model produce incorrect outputs. This highlights an area for improvement in the model's output when handling such intricate queries.

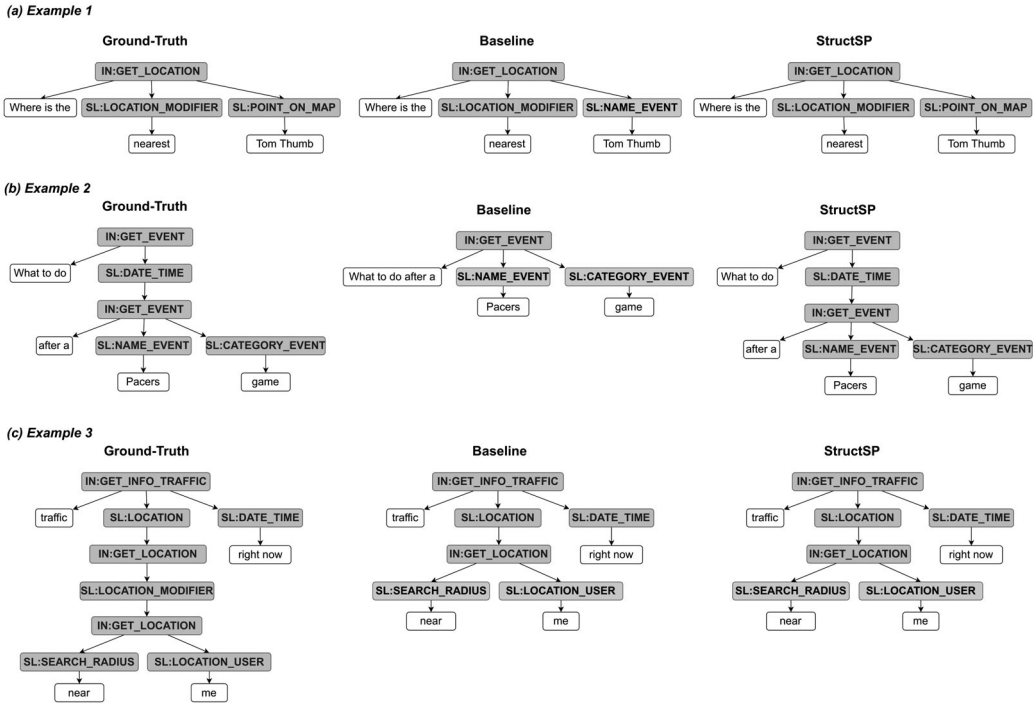


Figure 5.2: Case study outputs with tree representation.

Chapter 6

Conclusions and Future Works

6.1 Conclusion

In this research study, we have introduced a novel method to improve the performance of state-of-the-art models for hierarchical semantic parsing. We achieve this by incorporating knowledge about the structure of utterances into the semantic parsing process. Specifically, our proposed model learns contextual representations from the hierarchical representation of the utterances by using objective functions tailored to the semantic parsing task. In addition, we introduced a novel integration mechanism of grammar rules, which encode structural knowledge, during training and inference to prevent unpromising label predictions. To demonstrate the effectiveness of our method, we conducted experiments on the TOP and TOPv2 datasets, and the results show that our model outperforms previous state-of-the-art approaches.
However, our work has a few limitations that need to be addressed:

- **Non-English datasets**: Currently, our method is only applicable to English text. It means that our findings can be specific to the English language, and the results may vary when applied to other languages. To further explore this, we plan to apply our method to non-English datasets like MTOP (Li et al., 2021).

- **Grammar Constraint**: The effectiveness of using grammar with low resource data can be uncertain, as observed in our experiments on the TOPv2 dataset (section 4.3). This uncertainty arises because the extracted grammar from the training data may not be general enough to capture the constraints of new data instances in validation or test sets. To ensure the effectiveness of our method, it's crucial to provide grammar rules that cover all possible grammar variations, especially in low-resource settings.

- **Prediction Time**: Our current approach uses a recursive insertion-based strategy for prediction. This means that the output of the previous parsing step serves as input for the subsequent parsing step, leading to a recursive process until a terminal signal is reached. As a result, parsing a complex tree with multiple intents/slots can be time-consuming due to the recursive nature of the method. In future research, we will focus on improving the prediction time by predicting all labels at the same level in the parsed tree instead of predicting them one by one.

6.2 Future Works

There are several noteworthy directions for future research and development that we aim to emphasize:

- **Firstly**, we intend to extend the adaptability of our framework to tackle other tasks and other languages that involve nested representation datasets, such as Named Entity Recognition (NER). By exploring the applicability of our approach in diverse domains, we can assess its effectiveness in different contexts and expand its utility beyond the specific task addressed in this thesis.

- **Secondly**, another direction for future research is to develop techniques that enable the model to adapt to dynamic grammar variations. Instead of relying solely on a fixed set of grammar rules during training, the model could be designed to learn and update grammar rules dynamically from the data itself. This adaptive grammar approach would allow the model to handle novel or evolving language patterns more effectively and improve its performance in scenarios where the grammar rules may change over time.

- **Furthermore**, given the characteristics of the recursive insertion-based strategy employed in our framework, parsing complex trees with multiple intents/slots (labels) may result in lengthy processing times. As part of future work, we aim to explore strategies and techniques to optimize the parsing prediction time. This optimization could involve refining the algorithmic efficiency, exploring parallel processing, or employing advanced computational techniques to expedite the parsing process without compromising accuracy.

- **Lastly**, we believe that conducting further analysis and investigation into the model's interpretability and explainability would be an essential direction for future research. Understanding how the model arrives at its predic-

tions and providing human-interpretable explanations can enhance the trust and acceptance of the model in real-world applications.

6.3 Publications and Awards

6.3.1 Publications related to the thesis

- **Dinh-Truong Do**, Minh-Phuong Nguyen, Minh-Le Nguyen. "StructSP: Efficient Fine-tuning of Task-Oriented Dialog System by Using Structure-aware Boosting and Grammar Constraints". *In Findings of the Association for Computational Linguistics: ACL 2023, pages 10206–10220, Toronto, Canada. Association for Computational Linguistics.*

6.3.2 Other publications

- **Dinh-Truong Do**, Chau Nguyen, Vu Tran, Chau Nguyen, Ken Satoh, Yuji Matsumoto, and Minh-Le Nguyen. "CovRelex-SE: Adding semantic information for relation search via sequence embedding". *In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics EACL 2023: System Demonstrations, pages 35–42, Dubrovnik, Croatia, May 2023.*

- **Dinh-Truong Do**, Minh-Phuong Nguyen, Minh-Le Nguyen. "GRAM: Grammar based Refined-Label Representing Mechanism in the Hierarchical Semantic Parsing Task". *In International Conference on Applications of Natural Language to Information Systems, pp. 339-351. Cham: Springer Nature Switzerland, 2023.*

- **Dinh-Truong Do**, Ha Thanh Nguyen, Thang Ngoc Bui, and Hieu Dinh Vo. "Vsec: Transformer-based model for Vietnamese spelling correction". *In Proceedings of PRICAI 2021: 18th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam, November 8–12, 2021. Part II 18, pages 259–272.*

- **Dinh-Truong Do**. "Kodiak@Alqac2021: Deep learning for Vietnamese legal information processing". *In 2021 13th International Conference on Knowledge and Systems Engineering (KSE), pages 1–5.*

- Kien-Tuan Ngo, **Dinh-Truong Do**, Thu-Trang Nguyen, and Hieu Dinh Vo. "Ranking warnings of static analysis tools using representation learning". *In 2021 28th Asia-Pacific Software Engineering Conference (APSEC), pages 327–337.*

- Binh Dang, **Dinh-Truong Do**, and Le-Minh Nguyen. "Tbart: Abstractive summarization based on the joining of topic modeling and Bart". *In 2022 14th International Conference on Knowledge and Systems Engineering (KSE), pages 1–6.*

- Chau Nguyen, Minh-Quan Bui, **Dinh-Truong Do**, Nguyen-Khang Le, Dieu-Hien Nguyen, Thu-Trang Nguyen, Ha-Thanh Nguyen, Vu Tran, Le-Minh Nguyen, Ngoc-Cam Le, Thi-Thuy Le, Minh-Phuong Nguyen,Tran-Binh Dang, Truong-Son Nguyen, Viet-Anh Phan, Thi-Hai-Yen Vuong, Minh-Tien Nguyen, Tung Le, and Tien-Huy Nguyen, "ALQAC 2022: A Summary of the Competition". *In 2022 14th International Conference on Knowledge and Systems Engineering (KSE). 2022, pp. 1-5.*

- Bui, Quan Minh, Chau Nguyen, **Dinh-Truong Do**, Nguyen-Khang Le, Dieu-Hien Nguyen, Thi-Thu-Trang Nguyen, Minh-Phuong Nguyen, and Minh Le Nguyen. "JNLP Team: Deep Learning Approaches for Tackling Long and Ambiguous Legal Documents in COLIEE 2022". *In New Frontiers in Artificial Intelligence: JSAI-isAI 2022 Workshop, JURISIN 2022, and JSAI 2022 International Session, Kyoto, Japan, June 12–17, 2022, Revised Selected Papers, pp. 68-83.*

- Quan Minh Bui, **Dinh-Truong Do**, Nguyen-Khang Le, Dieu-Hien Nguyen, Khac-Vu-Hiep Nguyen, Trang Pham Ngoc Anh, and Minh Le Nguyen. "JNLP @COLIEE-2023: Data Augmentation and Large Language Model for Legal Case Retrieval and Entailment". *In JURISIN 2023 post-proceedings (LNAI) (Accepted)*

6.3.3 Awards

- Ranked first place among all Task 4 (Legal Textual Entailment) competitors of legal competition COLIEE 2023.

- Runner-up prize in Legal Text Retrieval task Zalo AI competition in 2021.

- Organizing committee of the legal Workshop of KSE 2022: Automated Legal Question Answering Competition (ALQAC 2022)

References

Aghajanyan, A., Maillard, J., Shrivastava, A., Diedrick, K., Haeger, M., Li, H., ... Gupta, S. (2020, November). Conversational semantic parsing. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5026–5035). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.408 doi: 10.18653/v1/2020.emnlp-main.408

Babu, A., Shrivastava, A., Aghajanyan, A., Aly, A., Fan, A., & Ghazvininejad, M. (2021, June). Non-autoregressive semantic parsing for compositional task-oriented dialog. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 2969–2978). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.236 doi: 10.18653/v1/2021.naacl-main.236

Bai, X., Song, L., & Zhang, Y. (2022, October). Semantic-based pre-training for dialogue understanding. In *Proceedings of the 29th international conference on computational linguistics* (pp. 592–607). Gyeongju, Republic of Korea: International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2022.coling-1.49

Baranowski, A., & Hochgeschwender, N. (2021, August). Grammar-constrained neural semantic parsing with LR parsers. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 1275–1279). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.108

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 acm conference on fairness, accountability, and transparency* (pp. 610–623).

Chen, Q., Zhuo, Z., & Wang, W. (2019). Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*. Retrieved from https://arxiv.org/abs/1902.10909

Chen, X., Ghoshal, A., Mehdad, Y., Zettlemoyer, L., & Gupta, S. (2020, Novem-

ber). Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 5090–5100). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.413 doi: 10.18653/v1/2020.emnlp-main.413

Desai, S., & Aly, A. (2021, August). Diagnosing transformers in task-oriented semantic parsing. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 57–62). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-acl.5 doi: 10.18653/v1/2021.findings-acl.5

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). 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)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N19-1423 doi: 10.18653/v1/N19-1423

Einolghozati, A., Pasupat, P., Gupta, S., Shah, R., Mohit, M., Lewis, M., & Zettlemoyer, L. (2019). Improving semantic parsing for task oriented dialog. *arXiv preprint arXiv:1902.06000*. doi: 10.48550/ARXIV.1902.06000

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Frosst, N., Papernot, N., & Hinton, G. (2019). Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning* (pp. 2012–2020). Retrieved from http://proceedings.mlr.press/v97/frosst19a/frosst19a.pdf

Gao, T., Yao, X., & Chen, D. (2021, November). SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 6894–6910). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.emnlp-main.552 doi: 10.18653/v1/2021.emnlp-main.552

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., ... Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, *3*(1), 1–23. Retrieved from https://dl.acm.org/doi/10.1145/3458754

Gupta, S., Shah, R., Mohit, M., Kumar, A., & Lewis, M. (2018, October-November). Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2787–2792). Brussels, Belgium:

Association for Computational Linguistics. Retrieved from https://aclanthology.org/D18-1300 doi: 10.18653/v1/D18-1300

He, W., Dai, Y., Yang, M., Sun, J., Huang, F., Si, L., & Li, Y. (2022). Unified dialog model pre-training for task-oriented dialog understanding and generation. In *Proceedings of the 45th international acm sigir conference on research and development in information retrieval* (pp. 187–200).

Herzig, J., & Berant, J. (2021, August). Span-based semantic parsing for compositional generalization. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 908–921). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.acl-long.74 doi: 10.18653/v1/2021.acl-long.74

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: 10.48550/ARXIV.1412.6980

Knuth, D. E. (1965). On the translation of languages from left to right. *Information and Control*, *8*(6), 607-639. Retrieved from https://www.sciencedirect.com/science/article/pii/S0019995865904262

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. doi: https://doi.org/10.1093/bioinformatics/btz682

Li, H., Arora, A., Chen, S., Gupta, A., Gupta, S., & Mehdad, Y. (2021, April). MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume.* Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.eacl-main.257 doi: 10.18653/v1/2021.eacl-main.257

Liu, B., & Lane, I. (2016). Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020). K-bert: Enabling language representation with knowledge graph. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 2901–2908). doi: 10.1609/AAAI.V34I03.5681

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019a). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*. doi: https://doi.org/10.48550/arXiv.1907.11692

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019b). Roberta: A robustly optimized BERT pretraining approach. *CoRR*.

Louvan, S., & Magnini, B. (2020, December). Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th international conference on computational linguistics* (pp. 480–496). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2020.coling-main.42 doi: 10.18653/v1/2020.coling-main.42

Luo, Y., Guo, F., Liu, Z., & Zhang, Y. (2022, October). Mere contrastive learning for cross-domain sentiment analysis. In *Proceedings of the 29th international conference on computational linguistics* (pp. 7099–7111). Gyeongju, Republic of Korea: International Committee on Computational Linguistics. Retrieved from https://aclanthology.org/2022.coling-1.620

Mansimov, E., & Zhang, Y. (2022). Semantic parsing in task-oriented dialog with recursive insertion-based encoder. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 36, pp. 11067–11075). doi: https://doi.org/10.1609/aaai.v36i10.21355

Mesnil, G., He, X., Deng, L., & Bengio, Y. (2013). Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Proc. interspeech 2013* (pp. 3771–3775). doi: 10.21437/Interspeech.2013-596

Nandy, A., Sharma, S., Maddhashiya, S., Sachdeva, K., Goyal, P., & Ganguly, N. (2021, November). Question answering over electronic devices: A new benchmark dataset and a multi-task learning based QA framework. In *Findings of the association for computational linguistics: Emnlp 2021*.

Nguyen, H.-T., & Nguyen, L.-M. (2021). Sublanguage: A serious issue affects pretrained models in legal domain. *arXiv preprint arXiv:2104.07782*.

Pasupat, P., Gupta, S., Mandyam, K., Shah, R., Lewis, M., & Zettlemoyer, L. (2019). Span-based hierarchical semantic parsing for task-oriented dialog. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 1520–1526).

Phuong, N. M., Le, T., & Minh, N. L. (2022). Cae: Mechanism to diminish the class imbalanced in slu slot filling task. In C. Bădică, J. Treur, D. Benslimane, B. Hnatkowska, & M. Krótkiewicz (Eds.), *Advances in computational collective intelligence* (pp. 150–163). Cham: Springer International Publishing.

Rongali, S., Soldaini, L., Monti, E., & Hamza, W. (2020). Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of the web conference 2020* (pp. 2962–2968). doi: https://doi.org/10.1145/3366423.3380064

Rubin, O., & Berant, J. (2021, June). SmBoP: Semi-autoregressive bottom-up semantic parsing. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 311–324). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naacl-main.29 doi: 10.18653/v1/2021.naacl-main.29

Shrivastava, A., Chuang, P., Babu, A., Desai, S., Arora, A., Zotov, A., & Aly, A. (2021, November). Span pointer networks for non-autoregressive task-oriented semantic parsing. In *Findings of the association for computational linguistics: Emnlp 2021* (pp. 1873–1886). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.findings-emnlp.161 doi: 10.18653/v1/2021.findings-emnlp.161

Stern, M., Chan, W., Kiros, J., & Uszkoreit, J. (2019, 09–15 Jun). Insertion transformer: Flexible sequence generation via insertion operations. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 5976–5985). PMLR. Retrieved from https://proceedings.mlr.press/v97/stern19a.html

Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., ... Wu, H. (2019). ERNIE: enhanced representation through knowledge integration. *CoRR*, *abs/1904.09223*. Retrieved from http://arxiv.org/abs/1904.09223

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Xiao, C., Dymetman, M., & Gardent, C. (2016, August). Sequence-based structured prediction for semantic parsing. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1341–1350). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P16-1127

Yan, Z., Duan, N., Chen, P., Zhou, M., Zhou, J., & Li, Z. (2017). Building task-oriented dialogue systems for online shopping. In *Thirty-first aaai conference on artificial intelligence.* doi: https://doi.org/10.1609/aaai.v31i1.11182

Yin, P., & Neubig, G. (2017, July). A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th annual meeting of the associa-*

tion for computational linguistics (volume 1: Long papers) (pp. 440–450). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P17-1041

Yu, T., Wu, C.-S., Lin, X. V., Wang, B., Tan, Y. C., Yang, X., ... Xiong, C. (2020). Grappa: grammar-augmented pre-training for table semantic parsing. *arXiv preprint arXiv:2009.13845*. doi: https://doi.org/10.48550/arXiv.2009.13845

Yu, T., Zhang, R., Polozov, A., Meek, C., & Awadallah, A. H. (2020). Score: Pre-training for context representation in conversational semantic parsing. In *International conference on learning representations*. Retrieved from https://openreview.net/pdf?id=oyZxhRI2RiE

Zettlemoyer, L. S., & Collins, M. (2012). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, 13–18 Jul). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 11328–11339). PMLR. Retrieved from https://proceedings.mlr.press/v119/zhang20ae.html

Zhang, Z., Takanobu, R., Zhu, Q., Huang, M., & Zhu, X. (2020). Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, *63*(10), 2011–2027. doi: https://doi.org/10.1007/s11432-016-0037-0

Zhao, W., Arkoudas, K., Sun, W., & Cardie, C. (2022, July). Compositional task-oriented parsing as abstractive question answering. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4418–4427). Seattle, United States: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2022.naacl-main.328

Zhao, Y., Zheng, Y., Tian, Z., Gao, C., Yu, B., Yu, H., ... Zhang, N. L. (2022). Prompt conditioned vae: Enhancing generative replay for lifelong learning in task-oriented dialogue. *arXiv preprint arXiv:2210.07783*. doi: https://doi.org/10.48550/arXiv.2210.07783

Zhu, Q., Khan, H., Soltan, S., Rawls, S., & Hamza, W. (2020, November). Don't parse, insert: Multilingual semantic parsing with insertion based decoding. In *Proceedings of the 24th conference on computational natural language learning* (pp. 496–506). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.conll-1.40 doi: 10.18653/v1/2020.conll-1.40

Ziai, A. (2019, September). Compositional pre-training for neural semantic parsing. In *Proceedings of the 3rd international conference on natural language and speech processing* (pp. 135–141). Trento, Italy: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W19-7419